SQL Server SNMP Subagent™
User's Guide
for UNIX

SQL Server SNMP Subagent Release 11.0
Document ID: 32125-01-1100-01
Last Revised: July 6, 1995

Principal author: Systems Management Products Technical Publications

Contributing authors: Systems Management Products Group

Document ID: 32125-01-1100

This publication pertains to SQL Server SNMP Subagent Release 11.0 of the SYBASE database management software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Document Orders

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor.

Upgrades are provided only at regularly scheduled software release dates.

Sybase Trademarks

SYBASE, the SYBASE logo, APT-FORMS, Data Workbench, DBA Companion, Deft, GainExposure, GainInsight, Gain *Momentum*, PowerBuilder, Powersoft, Replication Server, SQL Advantage, SQL Debug, SQL SMART, SQL Solutions, SQR, Transact-SQL, and VQL are registered trademarks of Sybase, Inc. Adaptable Windowing Environment, ADA Workbench, AnswerBase, Application Manager, Applications from Models, APT-Build, APT-Edit, APT-Execute, APT-Library, APT-Translator, APT Workbench, Backup Server, Bit-Wise, Build *Momentum*, Camelot, Client-Library, Client / Server Architecture for the Online Enterprise, Client / Server for the Real World, Client Services, Configurator, Connection Manager, Database Analyzer, DBA Companion Application Manager, DBA Companion Resource Manager, DB-Library, Deft Analyst, Deft Designer, Deft Educational, Deft Professional, Deft Trial, Developers Workbench, Easy SQR, Embedded SQL, Enterprise Builder, Enterprise Client / Server, Enterprise CONNECT, Enterprise Manager, Enterprise Meta Server, Enterprise Modeler, Enterprise *Momentum*, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, EWA, Gain, Gain Interplay, Gateway Manager, InfoMaker, Interactive Quality Accelerator, Intermedia Server, IQ Accelerator, Maintenance Express, MAP, MDI, MDI Access Server, MDI Database Gateway, MethodSet, Movedb, Navigation Server,

Restricted Rights

Sybase, Inc., 6475 Christie Avenue, Emeryville, CA 94608.

Table of Contents

3. Using the Subagent

4. Objects in the Network Services MIB and RDBMS MIB

5. Objects in the SYBASE MIB

A. Messages

B. SNMP Version 2 Configuration

C. Troubleshooting

List of Figures

List of Tables

Preface

This document describes how to use SQL Server SNMP Subagent™ release 11.0.

Audience

This document is intended for anyone who uses SQL Server SNMP Subagent to monitor servers on a network.

Familiarity with the directory and file management commands for your operating system and the Simple Network Management Protocol (SNMP) is assumed.

How to Use This Book

Consult the following chapters as necessary:

- Chapter 1, "Introduction," describes how SQL Server SNMP Subagent communicates with the objects it manages and how it communicates with network managers.
- Chapter 2, "Using EMANATE," describes how to set up and use the master agent EMANATE.
- Chapter 3, "Using the Subagent," describes how to set up the Subagent initialization file and how to use the SQL Server SNMP Subagent.
- Chapter 4, "Objects in the Network Services MIB and RDBMS MIB," describes the objects defined in the *applTable* of the Network Services MIB and the objects in the Relational Database Management System MIB.
- Chapter 5, "Objects in the SYBASE MIB," describes the objects defined in the SYBASE MIB.
- Appendix A, "Messages," explains error, warning, and informational messages and suggests corrective actions.
- Appendix B, "SNMP Version 2 Configuration," describes the master agent configuration files for SNMP version 2.
- Appendix C, "Troubleshooting," describes how to troubleshoot Subagent and master agent problems.

Related Documents

- *SQL Server SNMP Subagent Installation Guide*, which contains detailed, platform-specific information on installing EMANATE and Subagent.
- *SQL Server SNMP Subagent Release Bulletin*, which contains platform-specific information including special installation instructions, special considerations, known issues and solutions, and technical support.

Other Sources of Information

Sybase offers a system and database administration class called "SQL Server Administration." For details, contact:

Mail: Education Registrars
Sybase Professional Services
77 South Bedford Street
Burlington, MA 01803

Phone: (800) 8-SYBASE or (617) 564-6970

Fax: (800) 792-2733 or (617) 564-6960

E-mail: registrars@sybase.com

Notational Conventions

This document uses the following style conventions:.

Table 1: Notational conventions

Example	Description
Return	Key names are capitalized.
Ctrl+p	A sequence of two key names separated by a plus sign indicates that you hold down the first key while pressing the second.
password	Variable names for which you substitute a value, book names, and directory and file names are in *italics*.
/usr/local	
pal	Command names, command options, utility names, utility flags, and other keywords are in **bold Helvetica**.
tar xvf	Commands you enter are in **bold Courier**.

Table 1: Notational conventions (continued)

Example	Description
`create procedure`	Computer output is in `normal Courier`.

If You Need Help

Each SYBASE® installation has a person designated to contact Sybase Technical Support. If you cannot resolve a problem using the manuals, ask the designated person at your site to contact the Sybase Technical Support Center. See Appendix C, “Troubleshooting,” for when it is appropriate to contact Sybase Technical Support.

1 Introduction

SQL Server SNMP Subagent (the Subagent) is an application that allows network management tools to monitor characteristics of SYBASE SQL Server™ and Open Server™ applications. In addition, the Subagent also acts as an event monitoring tool that automatically notifies the network manager when an event occurs.

To communicate with the Subagent, network managers use the Simple Network Management Protocol (SNMP). The Subagent supports version 1 and version 2 of SNMP.

This chapter describes the SQL Server and Open Server characteristics you can monitor with the Subagent and describes how network managers communicate with the Subagent. See Appendix B, "SNMP Version 2 Configuration" for more information on how the Subagent supports the security features of SNMP version 2.

Features of the SQL Server SNMP Subagent

The Subagent allows you to monitor characteristics of SQL Servers and Open Servers and automatically notifies network managers when specific events occur. The following sections describe these features.

Monitoring SQL Servers

The Subagent monitors the following SQL Server characteristics:

- Status of SQL Server.
- Server contact person. From SQL Server release 4.9.x, the Subagent provides the login of the contact; it cannot provide the full name of the contact.
- Number of active connections.
- Total number of attempted inbound connections.
- Date and time when a change in the number of inbound connections occurred.
- Date and time that the most recent connection occurred.
- Number of transactions completed. This characteristic is not available from SQL Server release 4.9.x.

- Total number of database reads.
- Total number of database writes.
- Number of network packets received.
- Number of network packets sent.
- Maximum number of concurrent users.
- Maximum number of configured connections.
- Current configuration parameters.
- Device names.
- Device space allocated.
- Device space available.
- Maximum amount of device space used.
- Number of times the maximum amount of device space has been reached.
- State of database access.
- Status of database options.
- Date and time the database was made active.
- Database contact person. From SQL Server release 4.9.x, the Subagent provides the login of the contact; it cannot provide the full name of the contact.
- Server that created or last restructured the database.
- Database size.
- Date of last dump transaction log.
- Database segments.
- Maximum size of database segments. This characteristic is not available from SQL Server release 4.9.x.
- Current usage of each segment.
- Maximum usage of each segment.
- Number of times the system attempted to exceed allocated space.

Monitoring Open Servers

The Subagent monitors the following Open Server characteristics:

- Server status

- Number of inbound connections
- Date and time when the number of inbound connections changed

Notifying Network Managers of SQL Server Events

Another feature of the Subagent is automatic event notification. This feature enables the Subagent to notify a network manager when:

- The Subagent starts
- A database status is not "active" or "available"
- A segment becomes full

Accessing Server Information

To access SQL Server or Open Server information using SNMP, you need the following components:

- **Network manager**—requests information from objects in the network. Network managers can also modify the values of some objects. The network manager sends requests to User Datagram Protocol (UDP) port 161 and listens to UDP port 162 for responses to requests and for event traps generated by the Subagent. To request information and to modify values, network managers use protocol data units (PDU).
- **Master agent**—listens to port 161 and routes the requests to the appropriate sub-agent. The master agent returns responses from sub-agents to the network manager using port 162. As part of the Subagent distribution, Sybase provides the master agent EMANATE. See Chapter 2, "Using EMANATE" for more information on EMANATE.
- **SQL Server SNMP Subagent**—monitors characteristics of the SQL Servers and Open Servers and stores this information in an internal cache. When the Subagent receives a request from EMANATE, it accesses the internal cache and returns the requested information. The Subagent also detects events and reports these events as SNMP traps.

Figure 1-1 illustrates how a network manager accesses SQL Server information:

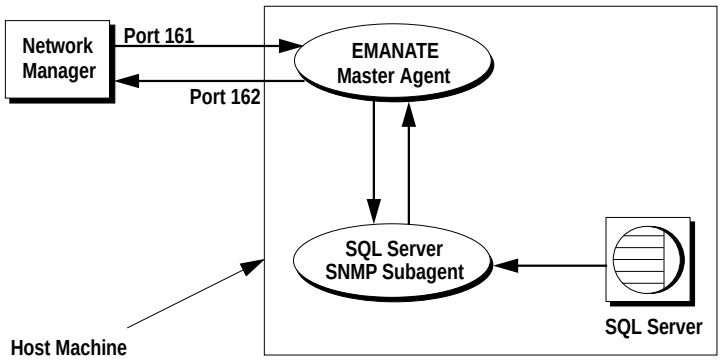

Figure 1-1: Network manager accessing SQL Server information

Management Information Bases

A management information base (MIB) is a virtual information store that allows access to a set of managed objects. Before a network manager can access objects in the network, it must import a set of MIBs to know what objects are available. There are three MIBs that the Subagent supports that allow network managers to access server information using SNMP:

- **Relational Database Management System (RDBMS) MIB**—an Internet standard for supplying information about installed servers, databases, and the relation of servers and databases. A network manager reads a compiled version of the RDBMS MIB so that it knows what types of information it can access about a server or database. The RDBMS MIB is also known as Request for Comments (RFC) 1697. See Chapter 4, "Objects in the Network Services MIB and RDBMS MIB" for more information on the objects in the RDBMS MIB.
- **Network Services MIB**—an Internet standard for supplying information on network service applications. The Subagent supports the application table (*applTable*) of this MIB. The application table contains an entry for each SQL Server or Open Server that the Subagent is monitoring. A network manager reads a compiled version of the Network Services MIB so that it knows which servers are available. The Network Services MIB is also known as RFC 1565. See Chapter 4, "Objects in the Network

Services MIB and RDBMS MIB" for more information on the objects in the Network Services MIB.

- **SYBASE MIB**—a SYBASE-specific MIB that identifies current values of polling intervals and login information used by the Subagent to monitor servers. A network manager reads a compiled version of this MIB so that it can read and modify polling intervals, login information, and servers to monitor. See Chapter 5, "Objects in the SYBASE MIB," for more information on the objects in the SYBASE MIB.

The Subagent distribution includes the MIBs for you to import into your network manager. However, some network managers cannot compile MIB syntax. To assist you, Sybase also provides versions of the MIBs that have been modified for specific network managers. All of the MIBs that Sybase provides are in the *$SYBSNMP/mibs* directory:

- Files ending in *.txt* contain full text of each RFC.
- Files ending in *.my* have been stripped of text and can be compiled.
- Files ending in *.smi* are include files that your compiler may need.
- Files ending in *.snm_v22.my* have been edited so that SunNet Manager v2.2 can compile them.
- Files ending in *.hp.my* have been edited so that HP OpenView can compile them.

See your network manager documentation for information on importing MIBs. If you encounter errors importing MIBs, contact your network manager vendor.

2 Using EMANATE

You must use the master agent EMANATE to communicate with the Subagent; you cannot use a different master agent to communicate with the Subagent. As part of the Subagent distribution, Sybase provides EMANATE.

This chapter describes:

- How EMANATE works
- How to configure EMANATE
- How to define nonstandard communication ports for multiple master agents
- How to start EMANATE
- How to use EMANATE's log file

This chapter does not describe the SNMP version 2 configuration files for EMANATE. See Appendix B, "SNMP Version 2 Configuration" for more information.

How EMANATE Works

EMANATE performs the following functions:

- Listens for PDU requests from the network manager
- Verifies that the network manager has privileges to set or get object information
- Verifies the syntax of PDU requests
- Sends verified PDU requests to the Subagent
- Returns responses to the network manager
- Sends traps to the network manager

When you start EMANATE, it determines which ports it should listen on for communication with the network manager. Typically, master agents listen on port 161 and respond on port 162. However, you can specify which ports you want EMANATE to use. See "Defining Nonstandard Ports" on page 2-7 for more information.

After EMANATE starts, it reads the configuration file *snmpd.cnf*. This file contains configuration information on system variables, community specifications, and trap specifications. See "Configuring EMANATE" on page 2-2 for more information on *snmpd.cnf*.

When you start the Subagent, the Subagent contacts EMANATE to inform it which objects it is managing. Figure 2-1 depicts the interaction between EMANATE, the network manager, the initialization files, and the Subagent:

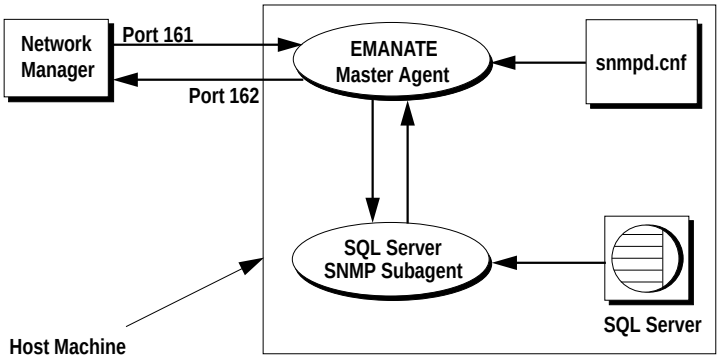

Figure 2-1: EMANATE configuration

➤ ***Note***

EMANATE also reads a set of SNMP version 2 configuration files. See Appendix B, “SNMP Version 2 Configuration” for more information.

Setting Up EMANATE

You must perform the following tasks to set up EMANATE:

1. Set up the configuration file.
2. Define port settings.
3. Define environment variables.

The following sections describe these tasks.

Configuring EMANATE

The configuration file for EMANATE is *snmpd.cnf*. In *snmpd.cnf*, you define host variables, community access, trap specifications, and trap notifications. The installation procedure creates *snmpd.cnf* in the directory identified by the environment variable SR_AGT_CONF_DIR. Typically, this directory is */etc/srconf/agt*, but

you can use another directory if you want. If you did not install EMANATE, contact the person who did so you can locate *snmpd.cnf*.

As part of the product distribution, Sybase provides a generic version of *snmpd.cnf*. During the installation process, the *SQL Server SNMP Subagent Installation Guide* directs you to modify some of the entries in *snmpd.cnf*, but it does not describe all of the entries in *snmpd.cnf*.

The following sections describe the entries in *snmpd.cnf* and the parameters you can modify.

◆ ***WARNING!***

Do not modify *snmpd.cnf* while EMANATE is running. If you do, errors may occur. If you need to modify *snmpd.cnf*, either shut down EMANATE first or make a copy of *snmpd.cnf* to modify. See "Shutting Down EMANATE" on page 2-9 for information.

Modifying System Information

During the installation process, you are required to define three parameters that describe the host system on which EMANATE is installed. These parameters are:

- **sysDescr**—typically describes the host name, make, and model of the machine on which you installed EMANATE. However, you can include any other information about the host that you want.
- **sysLocation**—describes the physical location of the system.
- **sysContact**—identifies the name of the person responsible for the host and information on how to contact him or her.

The descriptions you provide for **sysDescr**, **SysLocation**, and **sysContact** must be less than 254 characters and are limited to one line in length. In addition, your definitions cannot end with one or more spaces. If they do, EMANATE encounters errors.

To modify host parameters, edit *snmpd.cnf* with any text editor and modify the parameters. The format of system variables is:

```
variable_name definition
```

where *variable_name* is the name of the variable and *definition* is the variable definition. The following example depicts valid definitions for **sysDescr**, **sysLocation**, and **sysContact**:

```
sysDescr        SQUONK, SunOS_5.4 LX workstation
sysLocation     Capitol Rm 456
sysContact      George Washington (617)555 1212
```

➤ ***Note***

sysObjectID is another host variable defined in *snmpd.cnf*. Unlike the other host variables, you should not modify the definition of this variable. **sysObjectID** is set to a value that is reserved for Sybase.

Setting Community Access

A **community string** is basically a password that a network manager uses to access information. Each PDU contains a community string that EMANATE uses to authenticate the privileges of the network manager. By default, *snmpd.cnf* defines privileges for the "public" and "private" community strings. If you want, you can modify the privileges assigned to these community strings and insert additional community specifications.

The format of a community specification is:

```
community com_string ip_address privs comm_id
```

where **community** is a keyword, *com_string* is the community string name, *ip_address* is the address of the machine from which requests are authorized, *privs* are the privileges defined for the community, and *comm_id* is the identification number of the community. For example, the definition:

```
community   public  157.135.1.2 read    4
```

allows the network manager on node 157.135.1.2 with "read" access when it uses the "public" community string.

When you define community specifications, be aware of the following:

- There are two possible values for privileges: read or write.
- If you want to provide access from all hosts, use the address "0.0.0.0".
- The community identifier is a feature that is reserved for future use. You should always use the number four (4) for this value.

To define community specifications:

1. Edit *snmpd.cnf*.
2. Locate the "Community specification" section.
3. Insert definitions for each community string on each host.

➤ ***Note***

Each time you start EMANATE, it reads *snmpd.cnf*. If there are new community strings in *snmpd.cnf*, EMANATE adds the community string information to the configuration file *agt.pty*. See "Configuration Files" on page B-2 for more information.

Defining Trap Notifications

A **trap** is a notification that a specific event has occurred. For example, when a database becomes unavailable, the Subagent sends a trap informing EMANATE of the event. Before EMANATE can pass the trap to a network manager, you must insert a trap definition into *snmpd.cnf*.

The format of a trap definition is:

```
trap com_string ip_address
```

where *com_string* identifies the community included in the trap message, and *ip_address* is the address of the host machine to which EMANATE sends the trap. For example, the following definition:

```
trap public 135.136.1.2
```

directs EMANATE to send traps to the address "135.136.1.2" and include "public" in the community string.

To define trap notifications:

1. Edit *snmpd.cnf*.
2. Locate the "Trap community specification" section.
3. Insert trap definitions for each community string on each host.

➤ ***Note***

Each community string you specify for traps must be unique; the same community string cannot appear more than once in the trap definition section. If it does, EMANATE encounters problems.

Enabling and Disabling Authentication Traps

Authentication traps are a special type of security trap. When EMANATE receives a PDU, it verifies that the community string has privileges to set or get OID information. If the community string is invalid, EMANATE sends a trap to each of the hosts defined in the trap definition section of *snmpd.cnf*.

By default, authentication traps are enabled. However, you can disable authentication traps by redefining the parameter **snmpEnableAuthenTraps**.

To disable authentication traps, set **snmpEnableAuthenTraps** to 2.To enable authentication traps, set **snmpEnableAuthenTraps** to 1. For example, the entry:

```
snmpEnableAuthenTraps 2
```

disables authentication traps.

For information on other security features, see Appendix B, "SNMP Version 2 Configuration."

Defining Ports

For EMANATE to communicate with a network manager, you must define ports for EMANATE to use.Typically, a master agent listens to port 161 for requests and returns responses and traps to port 162. However, you can configure EMANATE to use different ports.

When you start EMANATE, it determines which port to use by looking for two environment variables: SR_SNMP_TEST_PORT and SR_SNMP_TRAP_PORT. If you do not define these environment variables, EMANATE looks for port information in the */etc/services* file. You use the environment variables for nonstandard port definitions and */etc/services* for standard port definitions.

The following sections describe how to use standard and nonstandard ports for communication.

➤ ***Note***

A side effect of using nonstandard ports is that when your network manager communicates with two master agents on the same host machine, it appears as if there are two separate machines. In actuality, there is only one machine.

Defining Standard Ports

You must define ports for EMANATE to use before you start it. To define standard ports, place two entries in the file */etc/services*.

To verify that ports are defined in */etc/services*, enter the command:

```
grep snmp /etc/services
```

grep should display the following:

```
snmp        161/udp
snmp-trap   162/udp
```

If **grep** does not display the appropriate entries, edit */etc/services* and put these entries in it.

➤ ***Note***

Because ports 161 and 162 are industry standards, you should use these ports unless they are already being used.

Defining Nonstandard Ports

In a typical SNMP environment, there is a single master agent that listens to port 161 for requests and returns responses and traps to port 162. However, in some environments, you may need to have multiple master agents. If you want, you can configure EMANATE to use nonstandard ports to communicate with network managers.

To configure EMANATE to use nonstandard ports, you must define two environment variables:

- SR_SNMP_TEST_PORT—identifies the port number for EMANATE to use for all communications other than traps.
- SR_TRAP_TEST_PORT—identifies the port number for EMANATE to use for all traps.

For example, the following c-shell commands direct EMANATE to use ports 2170 and 2171 for communication:

```
setenv SR_SNMP_TEST_PORT 2170
setenv SR_SNMP_TRAP_PORT 2171
```

➤ ***Note***

In addition to setting up EMANATE to use nonstandard ports, you must set up your network manager to communicate with EMANATE through these ports. See your network manager documentation or vendor for more information.

Setting Environment Variables

Before you start EMANATE, you must define two environment variables:

- SR_AGT_CONF_DIR—identifies the EMANATE configuration directory. This is the directory that contains *snmpd.cnf*.
- SR_MGR_CONF_DIR—identifies the EMANATE utility directory.

When you performed the Subagent and EMANATE installation, the installation procedure created directories based on your definitions of SR_AGT_CONF_DIR and SR_MGR_CONF_DIR. Sybase recommends that you set SR_AGT_CONF_DIR to */etc/srconf/agt* and that you set SR_MGR_CONF_DIR to */etc/srconf/mgr*. If you did not install EMANATE and the Subagent, contact the person who did to identify these directories.

To set SR_AGT_CONF_DIR, enter the following c-shell command:

```
setenv SR_AGT_CONF_DIR master_agent_conf_dir
```

where *master_agent_conf_dir* identifies the configuration directory. For example:

```
setenv SR_AGT_CONF_DIR /etc/srconf/agt
```

sets SR_AGT_CONF_DIR to */etc/srconf/agt*.

To set SR_MGR_CONF_DIR, enter the following c-shell command:

```
setenv SR_MGR_CONF_DIR master_agent_util_dir
```

where *master_agent_util_dir* identifies the utility directory. For example:

```
setenv SR_MGR_CONF_DIR /etc/srconf/agt
```

sets SR_MGR_CONF_DIR to */etc/srconf/agt*.

Starting EMANATE

The following sections describe what you should do before starting EMANATE, how to start EMANATE, and how to use command line options to capture messages that EMANATE generates.

Before You Start EMANATE

Before you start EMANATE, you must:

- Be "root"
- Stop any existing EMANATE master agent process
- Define communication ports
- Set the SR_AGT_CONF_DIR and SR_MGR_CONF_DIR environment variables

The following sections describe how to set user to "root" and how to shut down a master agent. See "Defining Ports" on page 2-6 for more information on defining communication ports and "Setting Environment Variables" on page 2-8 for information on defining EMANATE's environment variable.

Setting User to root

To start EMANATE you must be "root". You can either log in as "root" or set user to "root" with the command:

```
su root
```

Shutting Down EMANATE

To shut down EMANATE:

1. Identify the process identification number (PID) of EMANATE. If you are on a machine using AIX, HP-UX, or SunOS SVR4, enter the command:

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

   If you are on a machine using SunOS BSD, enter the command:

   ```
   ps -guax | grep "snmp"
   ```

   The **ps** command returns the PID for any active EMANATE process. If **ps** does not return a PID, EMANATE is not running.

2. Terminate the process with the command:

   kill -9 *process_id*

   where *process_id* is the PID.

Starting EMANATE

Before you start EMANATE, make sure that it is not already running. If EMANATE is already running and you try to start another EMANATE process, your request is ignored.

To start EMANATE, enter the command:

```
snmpdm [-aperror -apwarn -aptrace] | [-apall]
       [-d]
```

where the command line arguments are:

- **–aperror**—enables the logging of error messages. Error messages are serious and can be potentially fatal.
- **–apwarn**—enables the logging of warning messages. Warning messages are nonfatal messages.
- **–aptrace**—enables the logging of trace messages. If you use this argument, EMANATE writes all messages to standard output. This release of the Subagent does not use the trace message capability of EMANATE. Therefore, you should never use this argument.
- **–apall**—enables the logging of error, warning, and trace messages. If you use this argument, EMANATE writes all messages to standard output.
- **–d**—redirects all messages to standard output.

By default, EMANATE starts as a daemon process. However, if you use the **-aptrace**, **-apall**, or **-d** arguments, EMANATE does not start as a daemon process because it uses standard output for displaying error messages. If you do not use the **-aptrace**, **-apall**, or **-d** arguments, EMANATE writes all messages to the log file */tmp/snmpd.log*.

➤ ***Note***

If you have problems starting EMANATE, make sure you have performed all the tasks defined in “Before You Start EMANATE” on page 2-9. If you are still encountering problems, see Appendix C, “Troubleshooting” for more information.

Using the Message Log

EMANATE uses the log file */tmp/snmpd.log* to save messages. By default, EMANATE creates this file in the */tmp* directory. However, you can direct EMANATE to create *snmpd.log* in another directory by using the environment variable SR_LOG_DIR.

To direct EMANATE to create *snmpd.log* in a different directory, define SR_DIR_LOG to the directory you want to use. For example, the command:

```
setenv SR_LOG_DIR /tmp/snmp
```

directs EMANATE to create *snmpd.log* in the */tmp/snmp* directory.

Starting EMANATE at Boot Time

Sybase recommends that you automatically start EMANATE when you start your system. You can do this by putting the EMANATE start command in your system startup file. The system startup file is different for each type of machine, but on machines running SunOS BSD the file is */etc/rc.local*. Check with your System Administrator for the file that is appropriate for your environment.

3 Using the Subagent

The SQL Server SNMP Subagent allows you to monitor characteristics of SQL Servers and Open Servers using network management products.

This chapter describes:

- How the Subagent works
- How to set up the Subagent initialization file
- How to start the Subagent
- How to shut down the Subagent

➤ ***Note***

You cannot use network managers to modify SQL Server characteristics.

How the Subagent Works

The Subagent monitors high level characteristics of SQL Servers and Open Servers by polling the servers at regular intervals. To identify which servers to monitor and how often to collect information, the Subagent reads its initialization file. By default, the initialization file is *snmp.ini*, but you can name it anything you want. However, this document refers to the initialization file as *snmp.ini*.

When you start the Subagent, it:

- Connects EMANATE to let it know which objects it is managing.
- Reads its initialization file to identify polling intervals to use and to determine which servers to contact.
- Contacts each server and extracts information. Each piece of information the Subagent extracts is represented by an object in the RDBMS MIB, the Network Services MIB, or the SYBASE MIB. The Subagent maintains this information internally so that network managers can access it or modify it. The Subagent maintains this connection for as long as the server and the Subagent are running.
- Writes database information to *snmp.ini*. The Subagent extracts database information from each SQL Server and puts some of this information in *snmp.ini*.

- Periodically updates MID information and *snmp.ini* with new database information. Also, the Subagent modifies polling intervals and the server monitoring list based on requests from network managers.

Figure 3-1 depicts the communication between the Subagent, the network manager, the master agent, the servers, the interfaces file and *snmp.ini*:

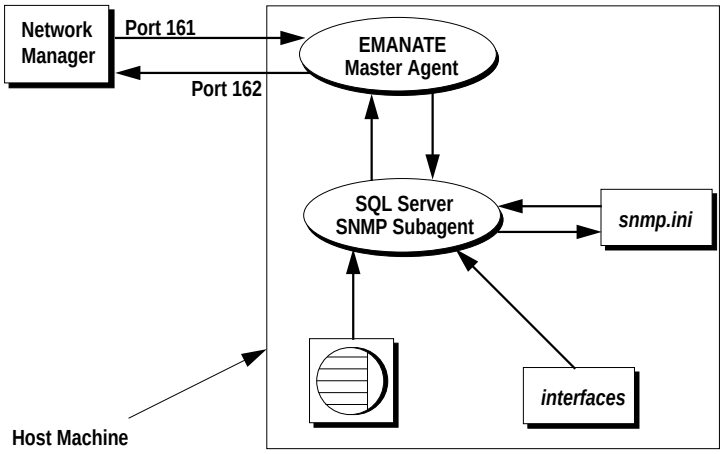

Figure 3-1: SQL Server SNMP Subagent configuration

➤ ***Note***

Sometimes, a server may be down when the Subagent attempts to contact it. In this situation, the Subagent attempts to contact the server during each LongSleep polling cycle.

Setting Up the Subagent

Subagent initialization file is *snmp.ini*. This file:

- Identifies which SQL Servers and Open Servers the Subagent is going to manage
- Controls how often the Subagent collects information from the servers

Sybase recommends that you do not use a single Subagent to monitor more than five servers. If you do, you may encounter performance issues due to network traffic and polling intervals.

The following sections describe how to create the initial version of *snmp.ini*, the format of entries in *snmp.ini*, and the different ways to modify *snmp.ini*.

➤ ***Note***

As part of the installation, you create an initial version of *snmp.ini*. If you did not install the Subagent, contact the individual who did to locate this initialization file.

Monitoring Servers on Remote Hosts

Sybase recommends that you install EMANATE and the Subagent on the same host machine that has the servers you want to monitor. Sybase does not recommend using the Subagent to monitor servers on a different host because:

- *applUpTime*, *applLastChange*, and *applLastInboundActivity*, will be inaccurate because the Subagent generates this value based on the local host's system time.
- *applDirectoryName* will be incorrect because the Subagent generates this value based on the Subagent's definition of the SYBASE environment variable.
- There may be performance issues due to network traffic and polling intervals.

See Chapter 4, "Objects in the Network Services MIB and RDBMS MIB" for information on these objects.

Creating *snmp.ini*

To create *snmp.ini* you use **snmpconfig.** The following sections describe what you need to know before you can use **snmpconfig** and how to use **snmpconfig** to create *snmp.ini*.

◆ ***WARNING!***

If you already have a version of *snmp.ini*, you probably do not want to use *snmpconfig* again. When you use *snmpconfig*, it overwrites the existing file; you cannot use *snmpconfig* to modify an existing file. To update an existing file, see "Modifying snmp.ini" on page 3-6.

Before You Use *snmpconfig*

To create entries in *snmp.ini*, **snmpconfig** reads your interfaces file to determine which servers it can access. To create valid entries in *snmp.ini*, you need the following information:

- Path and name of the initialization file you want to create. By default, **snmpconfig** creates *snmp.ini* in the $SYBASE directory.
- Path and name of interfaces file you want to use. By default, **snmpconfig** uses the interfaces file in the $SYBASE directory.
- Login for each server from which the Subagent collects data.
- Password of each login.
- Host name or IP address of host on which SQL Servers or Open Servers reside if they are not on the same host as where you start **snmpconfig**.

➤ ***Note***

Sybase does not recommend that you use the Subagent to monitor servers on remote hosts. See "Monitoring Servers on Remote Hosts" on page 3-3 for more information.

After you start **snmpconfig**, the utility:

1. Reads the interfaces file.
2. Determines if any servers in the interfaces file reside on the same host from which you started **snmpconfig.** If there are no servers on the same host, asks you for a different host name to use.

3. For each server on the host, **snmpconfig** asks you if you want to monitor the server. When you answer yes, **snmpconfig** asks you for a login and password on the server. The Subagent uses this login to collect information from the server.

The following example depicts a sample session with **snmpconfig**. In this example, bold-faced text depicts typical user input.

```
snmpconfig

Welcome to the Sybase SNMP Agent configuration
Procedure

$SYBASE is /usr/local/system1001

Checking interfaces file for Sybase Servers on
host squonk......

Server GABRIEL on squonk IP Address 157.111.124.6
was found.

Do you want this agent to monitor GABRIEL?(y/n):y

Enter the login name to be used by the agent to
GABRIEL.

Recommend  'sa' for SQL Servers 4.x  and 'SNMP'
for all others.

Enter login name: SNMP

Enter Password for SNMP [NULL for no password]:

Enter SNMP Password again:
```

Using snmpconfig

To use **snmpconfig** to create *snmp.ini*:

1. Enter the following command:

   snmpconfig [-c *ini_file*] [-I *interfaces_file*]

   where *ini_file* identifies the path and name of the file you want to create and *interfaces_file* identifies the path of the interfaces file you want snmpconfig to use. If you do not use the **-c** flag, **snmpconfig** creates *snmp.ini* in the $SYBASE directory. If you do not use the **-I** flag, **snmpconfig** uses the interfaces file in the *$SYBASE* directory.

2. Respond to the prompts from **snmpconfig**.

➤ ***Note***

When you respond to a yes or no prompt, use “y” for yes and “n” for no. When you enter a password, **snmpconfig** does not display the characters you enter on the screen.

Modifying *snmp.ini*

There are two ways you can modify the entries in *snmp.ini*: by editing *snmp.ini* with a text processor or by setting their values using your network manager. Using your network manager, you can modify polling intervals, modify passwords, add server entries, and remove server entries from *snmp.ini*.When you set an object representing an entry in *snmp.ini*, the Subagent writes the new value to *snmp.ini*.

Before you edit *snmp.ini*, you must shut down any Subagent that is using the file. If you do not, changes you make may be lost if the Subagent updates the file. See “How the Subagent Works” on page 3-1 for more information.

Format of snmp.ini

There are two sections in *snmp.ini*: a monitoring interval section and a server list section. The format of all entries must conform to the following rules:

- Each section must begin with a server name enclosed within brackets.
- Each section contains a set of *keyword=value* pairs.
- Keywords cannot contain spaces, commas, brackets, semicolons, or equal signs.
- One or more *keyword=value* pairs can appear under each section.
- If you repeat a *keyword=value* pairs for the same server, the Subagent ignores the first entry and uses the second entry.
- *keyword=value* pairs cannot extend to more than one line.
- To insert a comment line, begin the line with a semicolon. A comment can be on a line by itself or it can follow an entry.

Setting Polling Intervals

You set all polling interval information in the [SNMP$GENERAL] section of *snmp.ini*. There are two variables that determine how often the Subagent collects information:

- **ShortSleep**=*seconds*—the time in seconds between data polls. For example, the following entry:

  ```
  ShortSleep=600
  ```

  directs the Subagent to collect data from the servers every 600 seconds. The default value for ShortSleep is 300 seconds. If you modify ShortSleep, you must use a value between 10 and 1800 seconds, inclusive. Sybase recommends a value of 40 or higher if the Subagent is monitoring multiple servers.

- **LongSleep**=*number*—identifies the number of ShortSleep cycles that will occur before a LongSleep data collection occurs. For example, the following entries:

  ```
  ShortSleep=600
  LongSleep=5
  ```

  direct the Subagent to collect LongSleep information every 3000 seconds (600 * 5). The default value for **LongSleep** is 4. If you modify **LongSleep**, you must use a value between 1 and 20, inclusive.

The following is an example of the polling interval section in *snmp.ini*:

```
[SNMP$GENERAL]
ShortSleep=600
LongSleep=5
```

See “Objects and Polling Intervals” on page 4-107 for more information on the types of information the Subagent collects during ShortSleep and LongSleep polling intervals.

Creating Server Entries

Each server that you want the Subagent to monitor must have its own section in *snmp.ini*. You identify a section for a server with a [*server_name*] keyword, where *server_name* is the name of the server. For example, the following entry:

```
[PRODUCTION_SQL_SERVER]
```

identifies the beginning of the Production SQL Server section. All entries that appear below [PRODUCTION_SQL_SERVER] belong to

this server definition. *server_name* is case-sensitive and must match the entry in the *interfaces* file.

Sybase recommends that you do not use a single Subagent to monitor more than five servers. If you do, you may encounter performance issues due to network traffic and polling intervals.

There are up to six *keyword=value* pairs that can appear for a server entry:

- **LoginName**=*name*—the login the Subagent uses to log on to the server. This entry is optional. If you do not use this entry, the Subagent uses "SNMP" as the login.
- **PassWord**=*password*—the password for the login in a human readable format. To indicate that the login does not have a password, set password equal to "NULL". For example:

  ```
  PassWord=NULL
  ```

  If there is a **PassWord** entry for a server, you should not have a **passX** entry for the same server. For more information, see "Modifying Server Passwords" on page 3-10.
- **passX**=*scrambled*—the password for the login in a secure format. If there is a passX entry for a server, you should not have a PassWord entry for the server. For more information, see "Modifying Server Passwords" on page 3-10.
- **Version**=*server_type*—the type of server where *server_type* can be "SQL Server" or another string. If the entry does not contain "SQL Server," the Subagent treats it as an Open Server. For Open Servers, *server_type* must be less than 119 characters. The value you specify for *server_type* appears in the *applTable* as *applVersion*. This entry is optional.
- **applIndex**=*number*—an integer between 1 and 99999 that identifies the application index number for the server. Each managed object monitored by the Subagent must have a unique number on a host. If you are using other sub-agents on the same host, you should coordinate the numbers in this list with the index numbers in other sub-agent initialization files.
- **ServerContact**=*name*—the name of the contact person responsible for the server. This entry is optional. The name appears in the *rdbmsSrvTable* as the *rdbmsSrvContact*.

The following is an example of SQL Server and Open Server entries in *snmp.ini*:

```
;The following entry defines a SQL Server
[PRODUCTION_SQL_SERVER]
apllIndex=15
loginname=SNMP
PassWord=agent
version=SQL Server

;The following entry is for an open server
[PRODUCTION_SERVER]
apllIndex=16
loginname=SNMP
PassWord=agent
version=Open Server
```

After you start the Subagent, it connects to each server in *snmp.ini* using the login and password. For each SQL Server, the Subagent extracts database information and inserts the following three entries into the appropriate server section:

- **dbname**=*name*—the name of a database managed by SQL Server.
- **dbid**=*number*—the database identification number as stored in the *master...sysdatabases* table.
- **dbcontact**=*name*—the name of the database owner.

The following is an example of *snmp.ini* after the Subagent inserts database entries for a SQL Server:

```
;The following entry defines a SQL Server
[PRODUCTION_SQL_SERVER]
apllIndex=15
loginname=SNMP
PassX=ABCDEFGHIJKLMNOPQRSTUVWXYZ
version=SQL Server

;The following entries are for each database
;that PRODUCTION_SQL_SERVER manages
dbname=master
dbid=1
dbcontact=sa

dbname=pubs2
dbid=2
dbcontact=Mackenzie Anya
```

```
dbname=test
dbid=3
dbcontact=Willy Brodeur

;The following entry is for an open server
[PRODUCTION_SERVER]
apllIndex=16
loginname=SNMP
PassX=ABCDEFGHIJKLMNOPQRSTUVWXYZ
version=Open Server
```

Modifying Server Passwords

Each server monitored by the Subagent must have a login for the agent to use. When you initially created logins in *snmp.ini*, **snmpconfig** took the password you entered and converted it into a secure format. Secure passwords are preceded by the keyword **passX=**.

To modify a password, you must replace the **passX=** entry with a new **PassWord=** entry. The next time you start the Subagent, the Subagent reads *snmp.ini*, converts the human readable password to a secure non-readable password, and then replaces the **PassWord=** entry with a **passX=** entry. Only the Subagent can decipher these passwords.

Creating Logins

For the Subagent to monitor SQL Servers or Open Servers, each server must have a login that the Subagent can use. The Subagent uses this login to collect information from the servers.

For SQL Server release 4.9.2, the Subagent must be able to use the "sa" login. For SQL Server 10.x and higher, Sybase recommends that you create the login "SNMP". When you create a login on SQL Servers 10.x and higher, you must grant the login **sa_role**. On Open Servers, Sybase recommends that you create an "SNMP" login also. The Subagent does not need a login to access Backup Servers.

To create logins on SQL Servers 10.x and higher, Sybase provides the **addSNMP.sql** isql script. **addSNMP.sql** creates an "SNMP" login with a password of "airplane". If you want a different password, modify

addSNMP.sql before you use it. This script is in the *$SYBSNMP/scripts* directory.

To use **addSNMP.sql**, enter the following command:

```
isql -Ulogin -Ppword -Sserver < $SYBSNMP/scripts/addSNMP.sql
```

where *login* is a login that has privileges to create logins on the server, *pword* is the login password, and *server* is the server on which you want to create the login.

Starting the Subagent

The following sections describe what you must do before starting the Subagent, how to start the Subagent, and how to shut down the Subagent.

Before You Start the Subagent

Before you start the Subagent, be aware of the following:

- You must start the master agent before you start the Subagent.
- You must be “root”.
- It is generally a good idea to start the Subagent before you start your SQL Servers or Open Servers. This is because the Subagent monitors the length of time the server has been up and stores this information in the *sysUpTime* OID. If you start the Subagent after you start your servers, the Subagent cannot provide a value for *sysUpTime*.
- You must define the environment variables SYBASE, SYBSNMP, SR_MGR_CONF_DIR and SR_AGT_CONF_DIR. See “Defining Subagent Environment Variables” on page 3-11 for more information on defining these environment variables.
- You must have *$SYBASE/bin* in your path.

Defining Subagent Environment Variables

You must define the following environment variables before you start the Subagent:

- SYBASE—identifies the SYBASE installation directory
- SYBSNMP—identifies the Subagent installation directory
- SR_AGT_CONF_DIR—identifies the master agent configuration directory

To set these environment variables, enter the following commands:

```
setenv SYBASE sybase_path
setenv SYBSNMP subagent_path
setenv SR_AGT_CONF_DIR /etc/srconf/agt
```

where *sybase_path* identifies the SYBASE installation directory, *subagent_path* identifies the Subagent installation directory, and *conf_dir* identifies the master agent configuration directory. Typically, the master agent configuration directory is */etc/srconf/agt*.

Starting SQL Server SNMP Subagent

Before you start the Subagent, make sure that it is not already running. If the Subagent is already running and you try to start another, both Subagents will run simultaneously. This may cause errors if both Subagents are using the same initialization file.

The format of the Subagent command is:

```
sybagt  [-c ini_file] [-D]
        [-e [log_file | stdout | stderr]]
        [-h] [-I interfaces_file] [-v]
```

where the command flags are:

- **–c** *ini_file*—identifies the path and name of the Subagent initialization file. If you do not use this flag, the Subagent looks for *snmp.ini* in the *$SYBASE* directory. If you use this flag but do not specify a path, the Subagent looks in the current directory for the file.
- **–D**—directs the Subagent to run in diagnostic mode. In diagnostic mode, the Subagent reads *snmp.ini*, attempts to contact all of the servers listed in *snmp.ini*, and then stops. If the Subagent encounters errors, it writes messages to the log file (see the **–e** flag). You do not need to be "root" to run the Subagent in diagnostic mode.
- **–e** *log_file*—directs the Subagent to write messages to the log file specified in *log_file*. If you do not specify a log file name, the Subagent uses *sybagt.log* in the */tmp* directory. If you specify **stdout** instead of a log file name, the Subagent writes messages to standard output.
- **–h**—displays command usage help.
- **–I** *interfaces_ file*—identifies the path and name of the interfaces file you want to use.The interfaces file identifies the port numbers of the servers the Subagent is managing. If you do not use this

flag, the Subagent uses the interfaces file in the *$SYBASE* directory. If you use this flag but do not specify a path, the Subagent looks in the current directory for the file.

- **–v**—displays the Subagent version string and then exits.

When you start the Subagent, you should run it in background mode. For example, the command:

```
sybagt &
```

starts the Subagent as a daemon process running in background mode.

Starting the Subagent at Boot Time

Sybase recommends that you automatically start the Subagent when you start your system. You can do this by putting the Subagent start command in your system startup file. The system startup file is different for each type of machine, but on machines running SunOS BSD the file is */etc/rc.local*. Check with your System Administrator for the file that is appropriate for your environment.

Shutting Down the Subagent

To shut down the Subagent, you:

1. Identify the PID of the Subagent process. If you are on a machine using AIX, HP-UX, or SunOS SVR4, enter the command:

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

   If you are on a machine using SunOS BSD, enter the command:

   ```
   ps -guax | grep "snmp"
   ```

   The **ps** command returns the PID for any active Subagent. If **ps** does not return a PID, the Subagent is not running.

2. Terminate the process with the command:

   ```
   kill -9 process_id
   ```

   where *process_id* is the PID.

4 Objects in the Network Services MIB and RDBMS MIB

The Subagent supports the *applTable* object of the Network Services MIB and the RDBMS MIB. This chapter describes:

- An overview of each table
- The objects in *applTable*
- The objects in the RDBMS MIB tables
- The type of information the Subagent collects from each of the objects

➤ ***Note***

Before your network manager can communicate with the objects the Subagent monitors, you must import *applTable* and the RDBMS MIB into your network manager. As part of the distribution, Sybase provides these MIBs. See "Management Information Bases" on page 1-4 for more information.

If you want your network manager to be able to set polling intervals or login passwords, you must import the SYBASE MIB into it also. See Chapter 5, "Objects in the SYBASE MIB," for information on the SYBASE MIB.

Overview of MIB Tables

The Subagent supports the following tables:

- *applTable*—provides information about the servers, such as the server's application index number and name. This table has one row for each SQL Server or Open Server. See "applTable Objects" on page 4-8 for more information.
- *rdbmsSrvTable*—provides information on the database servers running on a host, such as the vendor who produced the server. This table has one row for each SQL Server. See "rdbmsSrvTable Objects" on page 4-59 for more information.
- *rdbmsSrvInfoTable*—provides information on servers that are running, such as the date and time the server started. This table has one row for each SQL Server that is running. If no servers are running, this table is empty. See "rdbmsSrvInfoTable Objects" on page 4-65 for more information.

- *rdbmsSrvParamTable*—provides information on the server configuration parameters. This table has one row for each configuration parameter per SQL Server. See "rdbmsSrvParamTable Objects" on page 4-82 for more information.
- *rdbmsSrvLimitedResourceTable*—provides information on the devices, such as the amount of disk space used by the server. This table has row for each Sybase device for each server. See "rdbmsSrvLimitedResourceTable Objects" on page 4-89 for more information.
- *rdbmsDbTable*—provides information on all databases on a host, such as the database name and the database contact. This table has one row for each database. See "rdbmsDbTable Objects" on page 4-26 for more information.
- *rdbmsDbInfoTable*—provides information on each individual database, such as the size of the database. This table has one row for each database that is running. If no servers are running, this table is empty. See "rdbmsDbInfoTable Objects" on page 4-33 for more information.
- *rdbmsDbParamTable*—provides information on the database configuration parameters. This table has one row for each database parameter per database. See "rdbmsDbParamTable Objects" on page 4-42 for more information.
- *rdbmsDbLimitedResourceTable*—provides information on segments, such as segment names or the current size of a segment. This table has one row for each segment for each database. See "rdbmsDbLimitedResourceTable Objects" on page 4-50 for more information.
- *rdbmsRelTable*—describes the relationship of the databases to the servers. This table has one row for each database associated with SQL Server. See "rdbmsRelTable Objects" on page 4-98 for more information.

Figure 4-1 depicts an entity-relationship diagram of these tables.

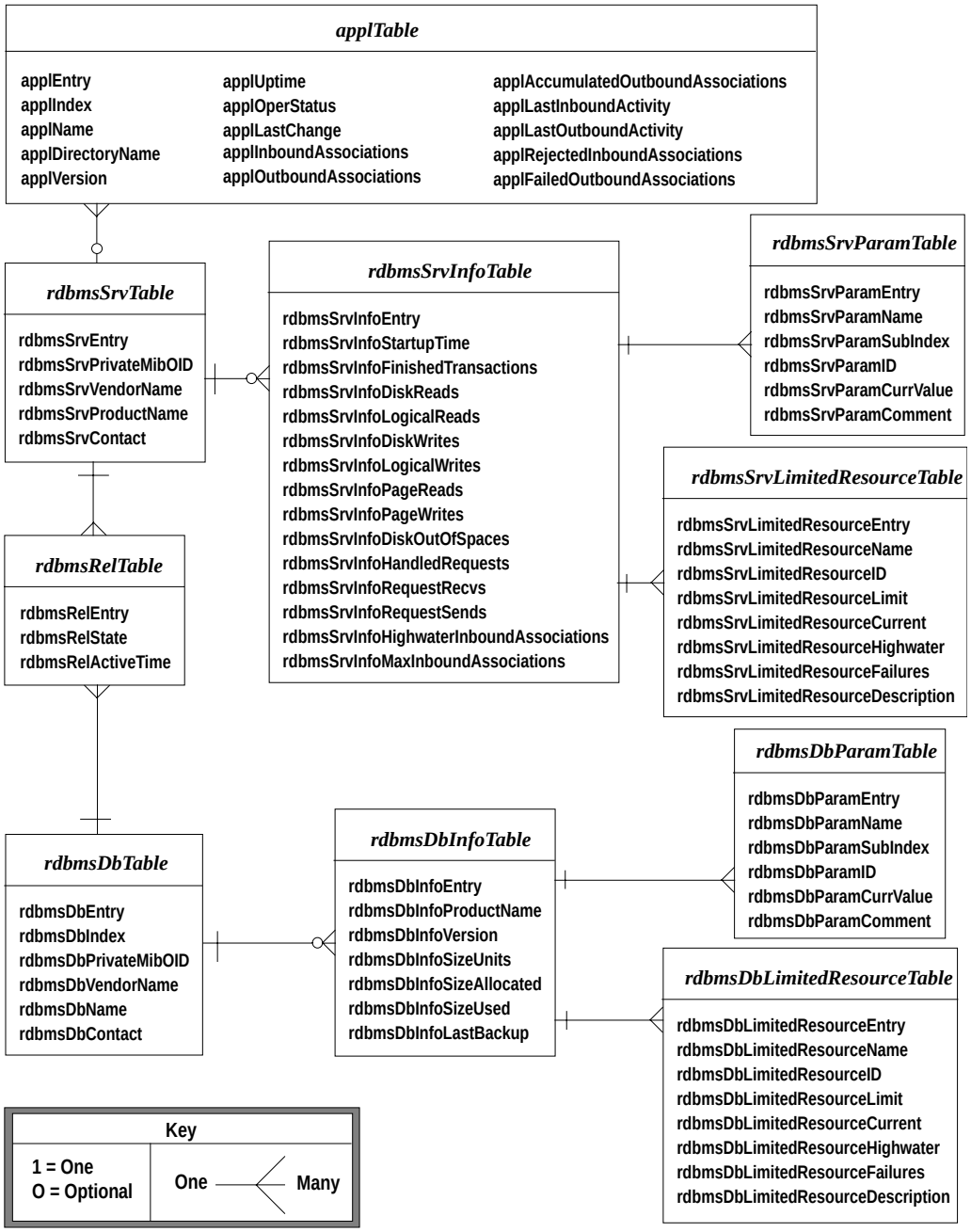

Figure 4-1: Entity-Relationship diagram

Table 4-1 provides the name of each table, each object in the table, the object data type, and your access to the object.

Table 4-1: *applTable* and RDBMS MIB objects

Object	Object Identifier	Data Type	Access
applTable	1.3.6.1.2.1.27.1	Not applicable	Not accessible
applEntry	1.3.6.1.2.1.27.1.1	Not applicable	Not accessible
applIndex	1.3.6.1.2.1.27.1.1.1	*Integer*	Read-Only
applName	1.3.6.1.2.1.27.1.1.2	*DisplayString*	Read-Only
applDirectoryName	1.3.6.1.2.1.27.1.1.3	*DistinguishedName*	Read-Only
applVersion	1.3.6.1.2.1.27.1.1.4	*DisplayString*	Read-Only
applUptime	1.3.6.1.2.1.27.1.1.5	*TimeStamp*	Read-Only
applOperStatus	1.3.6.1.2.1.27.1.1.6	*Integer*	Read-Only
applLastChange	1.3.6.1.2.1.27.1.1.7	*TimeStamp*	Read-Only
applInboundAssociations	1.3.6.1.2.1.27.1.1.8	*Gauge32*	Read-Only
applOutboundAssociations	1.3.6.1.2.1.27.1.1.9	*Gauge32*	Read-Only
applAccumulatedInboundAssociations	1.3.6.1.2.1.27.1.1.10	*Counter32*	Read-Only
applAccumulatedOutboundAssociations	1.3.6.1.2.1.27.1.1.11	*Counter32*	Read-Only
applLastInboundActivity	1.3.6.1.2.1.27.1.1.12	*TimeStamp*	Read-Only
applLastOutboundActivity	1.3.6.1.2.1.27.1.1.13	*TimeStamp*	Read-Only
applRejectedInboundAssociations	1.3.6.1.2.1.27.1.1.14	*Counter32*	Read-Only
applFailedOutboundAssociations	1.3.6.1.2.1.27.1.1.15	*Counter32*	Read-Only
rdbmsDBTable	1.3.6.1.2.1.39.1.1	Not applicable	Not accessible
rdbmsDbEntry	1.3.6.1.2.1.39.1.1.1	Not applicable	Not accessible
rdbmsDbIndex	1.3.6.1.2.1.39.1.1.1.1	*Integer*	Not accessible
rdbmsDbPrivateMibOID	1.3.6.1.2.1.39.1.1.1.2	*ObjectID*	Read-Only
rdbmsDbVendorName	1.3.6.1.2.1.39.1.1.1.3	*DisplayString*	Read-Only
rdbmsDbName	1.3.6.1.2.1.39.1.1.1.4	*DisplayString*	Read-Only
rdbmsDbContact	1.3.6.1.2.1.39.1.1.1.5	*DisplayString*	Read-Write

Table 4-1: *applTable* and RDBMS MIB objects (continued)

Object	Object Identifier	Data Type	Access
rdbmsDbInfoTable	1.3.6.1.2.1.39.1.2	Not applicable	Not accessible
rdbmsDbInfoEntry	1.3.6.1.2.1.39.1.2.1	Not applicable	Not accessible
rdbmsDbInfoProductName	1.3.6.1.2.1.39.1.2.1.1	*DisplayString*	Read-Only
rdbmsDbInfoVersion	1.3.6.1.2.1.39.1.2.1.2	*DisplayString*	Read-Only
rdbmsDbInfoSizeUnits	1.3.6.1.2.1.39.1.2.1.3	*Integer*	Read-Only
rdbmsDbInfoSizeAllocated	1.3.6.1.2.1.39.1.2.1.4	*Integer*	Read-Only
rdbmsDbInfoSizeUsed	1.3.6.1.2.1.39.1.2.1.5	*Integer*	Read-Only
rdbmsDbInfoLastBackup	1.3.6.1.2.1.39.1.2.1.6	*DisplayString*	Read-Only
rdbmsDbParamTable	1.3.6.1.2.1.39.1.3	Not applicable	Not accessible
rdbmsDbParamEntry	1.3.6.1.2.1.39.1.3.1	Not applicable	Not accessible
rdbmsDbParamName	1.3.6.1.2.1.39.1.3.1.1	*DisplayString*	Not accessible
rdbmsDbParamSubIndex	1.3.6.1.2.1.39.1.3.1.2	*Integer*	Not accessible
rdbmsDbParamID	1.3.6.1.2.1.39.1.3.1.3	*ObjectID*	Read-Only
rdbmsDbParamCurrValue	1.3.6.1.2.1.39.1.3.1.4	*DisplayString*	Read-Only
rdbmsDbParamComment	1.3.6.1.2.1.39.1.3.1.5	*DisplayString*	Read-Only
rdbmsDbLimitedResourceTable	1.3.6.1.2.1.39.1.4	Not applicable	Not accessible
rdbmsDbLimitedResourceEntry	1.3.6.1.2.1.39.1.4.1	Not applicable	Not accessible
rdbmsDbLimitedResourceName	1.3.6.1.2.1.39.1.4.1.1	*DisplayString*	Not accessible
rdbmsDbLimitedResourceID	1.3.6.1.2.1.39.1.4.1.2	*ObjectID*	Read-Only
rdbmsDbLimitedResourceLimit	1.3.6.1.2.1.39.1.4.1.3	*Integer*	Read-Only
rdbmsDbLimitedResourceCurrent	1.3.6.1.2.1.39.1.4.1.4	*Integer*	Read-Only
rdbmsDbLimitedResourceHighwater	1.3.6.1.2.1.39.1.4.1.5	*Integer*	Read-Only
rdbmsDbLimitedResourceFailures	1.3.6.1.2.1.39.1.4.1.6	*Counter32*	Read-Only
rdbmsDbLimitedResourceDescription	1.3.6.1.2.1.39.1.4.1.7	*DisplayString*	Read-Only

Table 4-1: *applTable* and RDBMS MIB objects (continued)

Object	Object Identifier	Data Type	Access
rdbmsSrvTable	1.3.6.1.2.1.39.1.5	Not applicable	Not accessible
rdbmsSrvEntry	1.3.6.1.2.1.39.1.5.1	Not applicable	Not accessible
rdbmsSrvPrivateMibOID	1.3.6.1.2.1.39.1.5.1.1	*ObjectID*	Read-Only
rdbmsSrvVendorName	1.3.6.1.2.1.39.1.5.1.2	*DisplayString*	Read-Only
rdbmsSrvProductName	1.3.6.1.2.1.39.1.5.1.3	*DisplayString*	Read-Only
rdbmsSrvContact	1.3.6.1.2.1.39.1.5.1.4	*DisplayString*	Read-Write
rdbmsSrvInfoTable	1.3.6.1.2.1.39.1.6	Not applicable	Not accessible
rdbmsSrvInfoEntry	1.3.6.1.2.1.39.1.6.1	Not applicable	Not accessible
rdbmsSrvInfoStartupTime	1.3.6.1.2.1.39.1.6.1.1	*DisplayString*	Read-Only
rdbmsSrvInfoFinishedTransactions	1.3.6.1.2.1.39.1.6.1.2	*Gauge32*	Read-Only
rdbmsSrvInfoDiskReads	1.3.6.1.2.1.39.1.6.1.3	*Counter32*	Read-Only
rdbmsSrvInfoLogicalReads	1.3.6.1.2.1.39.1.6.1.4	*Counter32*	Read-Only
rdbmsSrvInfoDiskWrites	1.3.6.1.2.1.39.1.6.1.5	*Counter32*	Read-Only
rdbmsSrvInfoLogicalWrites	1.3.6.1.2.1.39.1.6.1.6	*Counter32*	Read-Only
rdbmsSrvInfoPageReads	1.3.6.1.2.1.39.1.6.1.7	*Counter32*	Read-Only
rdbmsSrvInfoPageWrites	1.3.6.1.2.1.39.1.6.1.8	*Counter32*	Read-Only
rdbmsSrvInfoDiskOutOfSpaces	1.3.6.1.2.1.39.1.6.1.9	*Counter32*	Read-Only
rdbmsSrvInfoHandledRequests	1.3.6.1.2.1.39.1.6.1.10	*Counter32*	Read-Only
rdbmsSrvInfoRequestRecvs	1.3.6.1.2.1.39.1.6.1.11	*Counter32*	Read-Only
rdbmsSrvInfoRequestSends	1.3.6.1.2.1.39.1.6.1.12	*Counter32*	Read-Only
rdbmsSrvInfoHighwaterInboundAssociations	1.3.6.1.2.1.39.1.6.1.13	*Gauge32*	Read-Only
rdbmsSrvInfoMaxInboundAssociations	1.3.6.1.2.1.39.1.6.1.14	*Gauge32*	Read-Only
rdbmsSrvParamTable	1.3.6.1.2.1.39.1.7	Not applicable	Not accessible
rdbmsSrvParamEntry	1.3.6.1.2.1.39.1.7.1	Not applicable	Not accessible
rdbmsSrvParamName	1.3.6.1.2.1.39.1.7.1.1	*DisplayString*	Not accessible

Table 4-1: *applTable* and RDBMS MIB objects (continued)

Object	Object Identifier	Data Type	Access
rdbmsSrvParamSubIndex	1.3.6.1.2.1.39.1.7.1.2	*Integer*	Not accessible
rdbmsSrvParamID	1.3.6.1.2.1.39.1.7.1.3	*ObjectID*	Read-Only
rdbmsSrvParamCurrValue	1.3.6.1.2.1.39.1.7.1.4	*DisplayString*	Read-Only
rdbmsSrvParamComment	1.3.6.1.2.1.39.1.7.1.5	*DisplayString*	Read-Only
rdbmsSrvLimitedResourceTable	1.3.6.1.2.1.39.1.8	Not applicable	Not accessible
rdbmsSrvLimitedResourceEntry	1.3.6.1.2.1.39.1.8.1	Not applicable	Not accessible
rdbmsSrvLimitedResourceName	1.3.6.1.2.1.39.1.8.1.1	*DisplayString*	Not accessible
rdbmsSrvLimitedResourceID	1.3.6.1.2.1.39.1.8.1.2	*ObjectID*	Read-Only
rdbmsSrvLimitedResourceLimit	1.3.6.1.2.1.39.1.8.1.3	*Integer*	Read-Only
rdbmsSrvLimitedResourceCurrent	1.3.6.1.2.1.39.1.8.1.4	*Integer*	Read-Only
rdbmsSrvLimitedResourceHighwater	1.3.6.1.2.1.39.1.8.1.5	*Integer*	Read-Only
rdbmsSrvLimitedResourceFailures	1.3.6.1.2.1.39.1.8.1.6	*Counter32*	Read-Only
rdbmsSrvLimitedResourceDescription	1.3.6.1.2.1.39.1.8.1.7	*DisplayString*	Read-Only
rdbmsRelTable	1.3.6.1.2.1.39.1.9	Not applicable	Not accessible
rdbmsRelEntry	1.3.6.1.2.1.39.1.9.1	Not applicable	Not accessible
rdbmsRelState	1.3.6.1.2.1.39.1.9.1.1	*Integer*	Read-Only
rdbmsRelActiveTime	1.3.6.1.2.1.39.1.9.1.2	*DisplayString*	Read-Only

For more information on each table, see the following sections.

applTable Objects

applTable (1.3.6.1.2.1.27.1) provides access to information about the SQL Servers and Open Servers, such as:

- Server index number
- Server name
- Directory name of the server
- Release number
- The length of time the server has been running
- Status
- Current number of logins (inbound connections)
- Total accumulated inbound connections

This table has one row for each SQL Server or Open Server. The following sections describe the objects in *applTable*.

applEntry

Network Services MIB (RFC1565) Description

An entry associated with a network service application.

SQL Server Description

applEntry is an entry associated with a network service application.

Object Identifier

1.3.6.1.2.1.27.1.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

applIndex

Network Services MIB (RFC1565) Description

An index to uniquely identify the network service application.

SQL Server Description

This is a unique, surrogate key for each application. If there is a conflict, you will need to modify the Subagent initialization file to ensure unique numbers. See "Setting Up the Subagent" on page 3-3 for more information.

Object Identifier

1.3.6.1.2.1.27.1.1.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

An integer that uniquely identifies a monitored server on a host. The Subagent determines this number from the *snmp.ini* file. The snmpconfig program initially generates unique numbers, but if you edit the *snmp.ini* file you must make sure they are unique. This primary key will be used as a foreign key in several other tables, for example: *rdbmsSrvTable*.

applName

Network Services MIB (RFC1565) Description

The name the network service application chooses to be known by.

SQL Server Description

applName is the name of the server as defined in the Sybase *interfaces* file and in *snmp.ini*. See "Setting Up the Subagent" on page 3-3 for more information.

Object Identifier

1.3.6.1.2.1.27.1.1.2

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Server name as it appears in the Sybase *interfaces* file and in the *snmp.ini* file.

applDirectoryName

Network Services MIB (RFC1565) Description

The Distinguished Name of the directory entry where static information about this application is stored. An empty string indicates that no information about the application is available in the directory.

SQL Server Description

applDirectoryName identifies the *$SYBASE* installation directory. This is the same value stored in the SYBASE environment variable and it is updated upon initialization of the Subagent.

Object Identifier

1.3.6.1.2.1.27.1.1.3

Data Type

DistinguishedName

Max-Access

Read-Only

Return Values

The value of the SYBASE environment variable when the Subagent was started.

applVersion

Network Services MIB (RFC1565) Description

The version of network service application software.

SQL Server Description

applVersion identifies release information for the server. The Subagent updates *applVersion* upon initialization or connection to a server.

Object Identifier

1.3.6.1.2.1.27.1.1.4

Data Type

DisplayString

Max-Access

Read-Only

Return Values

SQL Server SNMP Subagent returns the value set by the **Version** keyword in the *snmp.ini* file. After a successful connection to a SQL Server, SNMP Subagent returns the value of @@VERSION from the SQL Server. If the server is an Open Server application, SNMP Subagent returns “OPENSERVER”.

applUptime

Network Services MIB (RFC1565) Description

The value of sysUpTime at the time the network service application was last initialized. If the application was last initialized prior to the last initialization of the network management subsystem, then this object contains a zero value.

SQL Server Description

applUptime identifies the time the Subagent identified that a server started. Time is expressed in the *sysUpTime* of the host where the Subagent is running.

The Subagent updates *applUptime* when it successfully connects to the server.

Object Identifier

1.3.6.1.2.1.27.1.1.5

Data Type

TimeStamp

Max-Access

Read-Only

Return Values

Integer representing the number seconds the host operating system was running before the server started. For example, if a server started 60 seconds after the host operating system started, the Subagent sets this OID to 60. For any servers that are already up when the Subagent starts, the Subagent returns zero (0).

applOperStatus

Network Services MIB (RFC1565) Description

Indicates the operational status of the network service application. “down” indicates that the network service is not available. ”running” indicates that the network service is operational and available. ”halted” indicates that the service is operational but not available. ”congested” indicates that the service is operational but no additional inbound associations can be accommodated. ”restarting” indicates that the service is currently unavailable but is in the process of restarting and will be available soon.

SQL Server Description

applOperStatus identifies the status of the server. The Subagent updates *applOperStatus* each time it collects information during a ShortSleep cycle and upon initialization or connection to a server. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.27.1.1.6

Data Type

Integer

Max-Access

Read-Only

Return Values

Possible values are:

- 1—up. The server is operational and available.
- 2—down. The server is not available.
- 3—halted. The server is operational but not available.
- 4—congested. The server is operational but no more connections can be made.
- 5—restarting. The server is up and running, but the Subagent cannot log on. Possible causes are an incorrect login name, an incorrect password, or an incorrect server name. To determine the exact problem, you can run the Subagent in diagnostic mode to

see if the login and password are correct. See "Starting SQL Server SNMP Subagent" on page 3-12 for more information.

applLastChange

Network Services MIB (RFC1565) Description

The value of sysUpTime at the time the network service application entered its current operational state. If the current state was entered prior to the last initialization of the local network management subsystem, then this object contains a zero value.

SQL Server Description

applLastChange identifies the time when the server entered its current state. Time is expressed in the *applUptime* of the host where the Subagent is running. The current state is in *applOperStatus*.

Object Identifier

1.3.6.1.2.1.27.1.1.7

Data Type

TimeStamp

Max-Access

Read-Only

Return Values

SQL Server SNMP Subagent returns a zero (0) until it observes a change in operational status. Then SQL Server SNMP Subagent returns the time at which the operational status changed.

applInboundAssociations

Network Services MIB (RFC1565) Description

The number of current associations to the network service application, where it is the responder. For dynamic single threaded processes, this will be the number of application instances.

SQL Server Description

applInboundAssociations identifies the number of login connections (inbound connections) to a server.

The Subagent updates *applInboundAssociations* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.27.1.1.8

Data Type

Gauge32

Max-Access

Read-Only

Return Values

Integer representing the number of logins. The Subagent login is included in the count.

applOutboundAssociations

Network Services MIB (RFC1565) Description

The number of current associations to the network service application, where it is the initiator. For dynamic single threaded processes, this will be the number of application instances.

SQL Server Description

applOutboundAssociations identifies the number of outbound connections from a server. This information is not available.

Object Identifier

1.3.6.1.2.1.27.1.1.9

Data Type

Gauge32

Max-Access

Read-Only

Return Values

Zero (0).

applAccumulatedInboundAssociations

Network Services MIB (RFC1565) Description

The total number of associations to the application entity since application initialization, where it was the responder. For dynamic single threaded processes, this will be the number of application instances.

SQL Server Description

applAccumulatedInboundAssociations identifies the number of attempted inbound connections to a server.

The Subagent updates *applAccumulatedInboundAssociations* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.27.1.1.10

Data Type

Counter32

Max-Access

Read-Only

Return Values

SQL Server SNMP Subagent returns the number of attempted inbound connections to a SQL Server. SQL Server SNMP Subagent returns zero (0) for Open Server applications.

applAccumulatedOutboundAssociations

Network Services MIB (RFC1565) Description

The total number of associations to the application entity since application initialization, where it was the initiator. For dynamic single threaded processes, this will be the number of application instances.

SQL Server Description

applAccumulatedOutboundAssociations identifies the accumulated number of outbound connections from a server. This information is not available.

Object Identifier

1.3.6.1.2.1.27.1.1.11

Data Type

Counter32

Max-Access

Read-Only

Return Values

Zero (0). This information is not available.

applLastInboundActivity

Network Services MIB (RFC1565) Description

The value of sysUpTime at the time this application last had an inbound association. If the last association occurred prior to the last initialization of the network subsystem, then this object contains a zero value.

SQL Server Description

applLastInboundActivity identifies the time when a change in the number of connections was observed. Time is expressed in the *sysUpTime* of the host where the Subagent is running.

The Subagent updates *applLastInboundActivity* each time it collects information during a ShortSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.27.1.1.12

Data Type

TimeStamp

Max-Access

Read-Only

Return Values

SQL Server SNMP Subagent returns zero (0) until the first change in connection activity occurs. Then SQL Server SNMP Subagent returns the time at which the change was observed.

applLastOutboundActivity

Network Services MIB (RFC1565) Description

The value of sysUpTime at the time this application last had an outbound association. If the last association occurred prior to the last initialization of the network subsystem, then this object contains a zero value.

SQL Server Description

applLastOutboundActivity identifies the time when this application last had an outbound connection. This information is not available.

Object Identifier

1.3.6.1.2.1.27.1.1.13

Data Type

TimeStamp

Max-Access

Read-Only

Return Values

Zero (0). This information is not available.

applRejectedInboundAssociations

Network Services MIB (RFC1565) Description

The total number of inbound associations the application entity has rejected, since application initialization.

SQL Server Description

applRejectedInboundAssociations identifies the number of rejected inbound connections. This information is not available.

Object Identifier

1.3.6.1.2.1.27.1.1.14

Data Type

Counter32

Max-Access

Read-Only

Return Values

Zero (0). This information is not available.

applFailedOutboundAssociations

Network Services MIB (RFC1565) Description

The total number associations where the application entity is initiator and association establishment has failed, since application initialization.

SQL Server Description

applRejectedOutboundAssociations identifies the number of rejected outbound connections. This information is not available.

Object Identifier

1.3.6.1.2.1.27.1.1.15

Data Type

Counter32

Max-Access

Read-Only

Return Values

Zero (0). This information is not available.

rdbmsDbTable Objects

rdbmsTable (1.3.6.1.2.1.39.1.1) provides you with the following information:

- Name of database vendor
- Name of database
- Name of database contact

The following sections describe these objects.

rdbmsDbEntry

RDBMS MIB (RFC1697) Description

An entry for a single database on the host. Whether a particular database is represented by a row in *rdbmsDbTable* may be dependent on the activity level of that database, according to the product's implementation. An instance of *rdbmsRelState* having the value active, other, or restricted implies that an entry, corresponding to that instance, will be present.

SQL Server Description

rdbmsDbEntry identifies an entry for a single database on the host. The Subagent first derives this information from *snmp.ini*, but then updates it when it connects to a server.

rdbmsDbEntry is indexed by *rdbmsDbIndex*.

Object Identifier

1.3.6.1.2.1.39.1.1.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

rdbmsDbIndex

RDBMS MIB (RFC1697) Description

A numeric index, unique among all the databases from all products on this host. This value is a surrogate for the conceptually unique key, which is {*PrivateMibOID*, *databasename*}.

SQL Server Description

rdbmsDbIndex is a numeric index. Each database on a host has a unique value. This is a unique primary surrogate key for the database.

Object Identifier

1.3.6.1.2.1.39.1.1.1.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

➤ ***Note***

rdbmsDbIndex is an index of *rdbmsDbEntry*.

rdbmsDbPrivateMibOID

RDBMS MIB (RFC1697) Description

The authoritative identification for the private MIB for this database, presumably based on the vendor, e.g., {enterprises 111 <optional subidentifiers>} for Oracle databases, {enterprises 757 <optional subidentifiers>} for Ingres databases, {enterprises 897 <optional subidentifiers>} for Sybase databases, etc. If no OBJECT IDENTIFIER exists for the private MIB, attempts to access this object will return *noSuchName* (SNMPv1) or *noSuchInstance* (SNMPv2).

SQL Server Description

rdbmsDbPrivateMibOID identifies the private MIB (SYBASE MIB) for the database. This is the Sybase Internet Assigned Numbers Authority (IANA) number.

The Subagent updates *rdbmsDbPrivateMibOID* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.1.1.2

Data Type

ObjectID

Max-Access

Read-Only

Return Values

The Sybase IANA number: 1.3.6.1.4.1.897

rdbmsDbVendorName

RDBMS MIB (RFC1697) Description

The name of the vendor whose RDBMS manages this database, for informational purposes.

SQL Server Description

rdbmsDbVendorName identifies the name of the vendor whose RDMBS manages this database.

The Subagent updates *rdbmsDbVendorName* during its initialization.

Object Identifier

1.3.6.1.2.1.39.1.1.1.3

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Sybase Inc.

rdbmsDbName

RDBMS MIB (RFC1697) Description

The name of this database, in a product specific format. The product may need to qualify the name in some way to resolve conflicts if it is possible for a database name to be duplicated on a host. It might be necessary to construct a hierarchical name embedding the RDBMS instance / installation on the host, and / or the owner of the database. For instance, " / test-installation / database-owner / database-name".

SQL Server Description

rdbmsDbName identifies the "name" of the database.

The Subagent updates *rdbmsDbName* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.1.1.4

Data Type

DisplayString

Max-Access

Read-Only

Return Values

The name of the database or "unknown." SQL Server SNMP Subagent initially reads the database name from the *snmp.ini* file. If the Subagent successfully connects to a SQL Server, the database name is derived from the name in *master.dbo.sysdatabases*.

rdbmsDbContact

RDBMS MIB (RFC1697) Description

The textual identification of the contact person for this managed database, together with information on how to contact this person. Note: if there is no server associated with this database, an agent may need to keep this in other persistent storage, e.g., a configuration file. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsDbContact identifies the contact person for the database. You can set this object from the network manager.

The Subagent updates *rdbmsDbContact* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.1.1.5

Data Type

DisplayString

Max-Access

Read-Write

Return Values

For SQL Server 10.x, the Subagent returns the full name and login of the database owner (DBO). For SQL Server 4.9.x, the Subagent returns the login of the DBO.

➤ ***Note***

The Subagent sets this object by extracting database information from a SQL Server. However, once you set this object from a network manager, the Subagent does not update it any longer; the Subagent continues to use the value you set.

rdbmsDbInfoTable Objects

rdbmsDbInfoTable (1.3.6.1.2.1.39.1.2) provides access to the following information:

- Name of product
- Version information
- Size of database
- Size allocated for database
- Size currently used
- Date of last backup

The following sections describe the objects in *rdbmsDbInfoTable*.

rdbmsDbInfoEntry

RDBMS MIB (RFC1697) Description

Information that must be present if the database is actively opened. If the database is not actively opened, then attempts to access corresponding instances in this table may result in either *noSuchName* (SNMPv1) or *noSuchInstance* (SNMPv2). "Actively opened" means at least one of the *rdbmsRelState* entries for this database in the *rdbmsRelTable* is active(2).

SYBASE SQL Server Description

rdbmsDbInfoEntry contains information that must be present if the database is actively opened. Actively opened means that the server is up and operational and the database is fully recovered.

If the database is not actively opened, attempts to access corresponding instances in this table may return either "no such name" or "no such instance."

Object Identifier

1.3.6.1.2.1.39.1.2.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

rdbmsDbInfoProductName

RDBMS MIB (RFC1697) Description

The textual product name of the server that created or last restructured this database. The format is product specific.

SQL Server Description

rdbmsDbInfoProductName identifies the product name of the server (SQL Server) that created or last restructured this database.

The Subagent updates *rdbmsDbInfoProductName* upon initialization or connection to a server.

Object Identifier

1.3.6.1.2.1.39.1.2.1.1

Data Type

DisplayString

Max-Access

Read-Only

Return Values

SQL Server

rdbmsDbInfoVersion

RDBMS MIB (RFC1697) Description

The version number of the server that created or last restructured this database. The format is product specific.

SQL Server Description

rdbmsDbInfoVersion identifies the version number of the server that created or last restructured the database. The Subagent collects this information from **@@version**.

The Subagent updates *rdbmsDbInfoVersion* upon initialization or connection to a server.

Object Identifier

1.3.6.1.2.1.39.1.2.1.2

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Version information. The format in which version information is returned is product-dependent.

rdbmsDbInfoSizeUnits

RDBMS MIB (RFC1697) Description

Identification of the units used to measure the size of this database in *rdbmsDbInfoSizeAllocated* and *rdbmsDbInfoSizeUsed*. bytes(1) indicates individual bytes, kbytes(2) indicates units of kilobytes, mbytes(3) indicates units of megabytes, gbytes(4) indicates units of gigabytes, and tbytes(5) indicates units of terabytes. All are binary multiples -- 1K = 1024. If writable, changes here are reflected in the get values of the associated objects.

SQL Server Description

rdbmsDbInfoSizeUnits identifies the unit of measure for identifying the amount of space allocated for the database and used by the database. All units are binary multiples; 1K equals 1024.

The Subagent updates *rdbmsDbInfoSizeUnits* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.2.1.3

Data Type

Integer

Max-Access

Read-Only

Return Values

Possible return values are:

- 2 (kilobytes)
- 3 (megabytes)

See Also

“rdbmsDbInfoSizeAllocated” on page 4-38 for information on space allocation and “rdbmsDbInfoSizeUsed” on page 4-39 for information on space used.

rdbmsDbInfoSizeAllocated

RDBMS MIB (RFC1697) Description

The estimated size of this database (in rdbmsDbInfoSizeUnits), which is the disk space that has been allocated to it and is no longer available to users on this host. *rdbmsDbInfoSize* does not necessarily indicate the amount of space actually in use for database data. Some databases may support extending allocated size, and others may not. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsDbInfoSizeAllocated identifies the maximum size allocated for the database.

The Subagent updates *rdbmsDbInfoSizeAllocated* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.2.1.4

Data Type

Integer

Max-Access

Read-Write

Return Values

The size allocated for the database in megabytes or kilobytes.

See Also

"rdbmsDbInfoSizeUnits" on page 4-37 for information on the unit of measure SQL Server SNMP Subagent is using.

rdbmsDbInfoSizeUsed

RDBMS MIB (RFC1697) Description

The estimated size of this database, in *rdbmsDbInfoSizeUnits*, which is actually in use for database data.

SQL Server Description

rdbmsDbInfoSizeUsed identifies the current reserved space for the data and log.

The Subagent updates *rdbmsDbInfoSizeUsed* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.2.1.5

Data Type

Integer

Max-Access

Read-Only

Return Values

The current size reserved for the database in kilobytes or megabytes.

See Also

"rdbmsDbInfoSizeUnits" on page 4-37 for information on the unit of measure SQL Server SNMP Subagent is using.

rdbmsDbInfoLastBackup

RDBMS MIB (RFC1697) Description

The date and time that the latest complete or partial backup of the database was taken. If a database has never been backed up, then attempts to access this object will result in either *noSuchName* (SNMPv1) or *noSuchInstance* (SNMPv2).

SQL Server Description

rdbmsDbInfoLastBackup identifies the date of the last dump transaction log.

The Subagent updates *rdbmsDbInfoLastBackup* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.2.1.6

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Date and time that the last transaction log was dumped. Date and time is a ten-field format consisting of the following:

Table 4-2: Date and time format

Field	Octets	Contents	Range
1	1-2	Year	0 - 65536
2	3	Month	1 - 12
3	4	Day	1 - 31
4	5	Hour	0 - 23
5	6	Minutes	0 - 59
6	7	Seconds	0 - 60 (Use 60 for leap-second)

Table 4-2: Date and time format (continued)

Field	Octets	Contents	Range
7	8	Deci-seconds	0 - 9
8	9	Direction from UTC (Coordinated Universal Time)	+ or -
9	10	Hours from UTC	0 - 11
10	11	Minutes from UTC	0 - 59

For example, the date and time Tuesday May 26, 1993 at 1:30:15 PM EDT would appear as:

```
1992-5-26,13:30:15.0,-4:0
```

If the offset from UTC is unknown, only the first seven fields are given. If the database has never been backed up, then the Subagent returns:

```
1900-1-1,00:00:0.0,0:0
```

rdbmsDbParamTable Objects

rdbmsDbParamTable (1.3.6.1.2.1.39.1.3) provides access to the configuration parameters for a database. The following sections describe the objects in *rdbmsDbParamTable*.

rdbmsDbParamEntry

RDBMS MIB (RFC1697) Description

An entry for a single configuration parameter for a database. Parameters with single values have a subindex value of one. If the parameter is naturally considered to contain a variable number of members of a class, e.g. members of the DBA user group, or files which are part of the database, then it must be presented as a set of rows. If, on the other hand, the parameter represents a set of choices from a class, e.g. the permissions on a file or the options chosen out of the set of all options allowed, AND is guaranteed to always fit in the 255 character length of a *DisplayString*, then it may be presented as a comma separated list with a subindex value of one. Zero may not be used as a subindex value. If the database is not actively opened, then attempts to access corresponding instances in this table may result in either *noSuchName* (SNMPv1) or *noSuchInstance* (SNMPv2). 'Actively opened' means at least one of the *rdbmsRelState* entries for this database in the *rdbmsRelTable* is active(2).

SQL Server Description

There is a row for every database option. These options can be:

- **abort tran on log full**
- **allow nulls by default**
- **dbname has changed**
- **dbo use only**
- **ddl in tran**
- **don't recover**
- **failed upgrade**
- **no chkpt on recovery**
- **no free space acctg**
- **not recovered**
- **read only**
- **select into/bulkcopy**
- **single user**
- **trunc log on chkpt**

Values returned depend on the version of SQL Server used. Length is limited to 64 characters. See the *SQL Server Commands Reference* for more information on these configuration parameters.

Sybase SQL Server configuration parameters do not have subindexes. However, *rdbmsDbParamEntry* is indexed by:

- *rdbmsDbIndex*
- *rdbmsDbParamName*
- *rdbmsDbParamSubIndex*

Object Identifier

1.3.6.1.2.1.39.1.3.1

Data Type

Not applicable

Max-Access

Not Accessible

Return Values

None

rdbmsDbParamName

RDBMS MIB (RFC1697) Description

The name of a configuration parameter for a database. This name is product-specific. The length is limited to 64 characters to constrain the number of sub-identifiers needed for instance identification (and to minimize network traffic).

SQL Server Description

rdbmsDbParamName identifies the name of a database option. This information corresponds to *status* and *status2* fields of *master.dbo.sysdatabases*.

The Subagent updates *rdbmsDbParamName* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.3.1.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

➤ ***Note***

rdbmsDbParamName is an index of *rdbmsDbParamEntry*.

rdbmsDbParamSubIndex

RDBMS MIB (RFC1697) Description

The subindex value for this parameter. If the parameter is naturally considered to contain a variable number of members of a class, e.g. members of the DBA user group, or files which are part of the database, then it must be presented as a set of rows. If, on the other hand, the parameter represents a set of choices from a class, e.g. the permissions on a file or the options chosen out of the set of all options allowed, AND is guaranteed to always fit in the 255 character length of a DisplayString, then it may be presented as a comma separated list with a subindex value of one. Zero may not be used as a value.

SQL Server Description

rdbmsDbParamSubIndex identifies subindex values for a database option. SYBASE SQL Server database options do not repeat parameters.

The Subagent updates *rdbmsDbParamSubIndex* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.3.1.2

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

➤ ***Note***

rdbmsDbParamSubIndex is an index of *rdbmsDbParamEntry*.

rdbmsDbParamID

RDBMS MIB (RFC1697) Description

The ID of the parameter which may be described in some other MIB (e.g., an enterprise-specific MIB module). If there is no ID for this *rdbmsDbParamName*, attempts to access this object will return *noSuchName* (SNMPv1) or *noSuchInstance* (SNMPv2).

SQL Server Description

This is the Sybase IANA number.

The Subagent updates *rdbmsDbParamID* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.3.1.3

Data Type

ObjectID

Max-Access

Read-Only

Return Values

The Sybase IANA number: 1.3.6.1.4.1.897

rdbmsDbParamCurrValue

RDBMS MIB (RFC1697) Description

The value for a configuration parameter now in effect, the actual setting for the database. While there may multiple values in the temporal domain of interest (for instance, the value to take effect at the next restart), this is the current setting. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsDbParamCurrValue identifies the value of the database configuration parameter now in effect.

The Subagent updates *rdbmsDbParamCurrValue* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.3.1.4

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Possible values are:

- True—configuration parameter is enabled
- False—configuration parameter is disabled

rdbmsDbParamComment

RDBMS MIB (RFC1697) Description

Annotation which describes the purpose of a configuration parameter or the reason for a particular parameter's setting. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsDbParamComment describes the purpose of the database configuration parameter or the reason for the current value of the parameter.

The Subagent updates *rdbmsDbParamComment* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.3.1.5

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Comment text in English.

rdbmsDbLimitedResourceTable Objects

rdbmsDbLimitedResourceTable (1.3.6.1.2.1.39.1.4) provides access to information on the limited resources (segments) for a database. This information includes the following for each database:

- The name of all segments
- The maximum size of each segment
- The current amount of reserved space for each segment
- The maximum amount of reserved space ever used
- The number of times unreserved space has been zero

The following sections describe the objects in *rdbmsDbLimitedResourceTable*.

rdbmsDbLimitedResourceEntry

RDBMS MIB (RFC1697) Description

An entry for a single limited resource kept per-database. A limited resource has maximum use determined by a parameter that might or might not be changeable at run time, or visible in the rdbmsDbParamTable. Examples would be the number of available locks, or disk space on a partition. Arrays of resources are supported through an integer sub index, which should have the value of one for single-instance names. Limited resources that are shared across databases, are best put in the rdbmsSvrLimitedResourceTable instead of this one. If the database is not actively opened, then attempts to access corresponding instances in this table may result in either noSuchName (SNMPv1) or noSuchInstance (SNMPv2). 'Actively opened' means at least one of the rdbmsRelState entries for this database in the rdbmsRelTable is active(2).

SQL Server Description

rdbmsDbLimtedResourceEntry identifies a single resource (segment) per database. *rdbmsDbLimtedResourceEntry* is indexed by:

- *rdbmsDbIndex*
- *rdbmsDbLimitedResouceName*

Object Identifier

1.3.6.1.2.1.39.1.4.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

rdbmsDbLimitedResourceName

RDBMS MIB (RFC1697) Description

The name of the resource, for instance “global locks” or “locks for the FOO database”, or “data space on /dev/rdsk/5s0 for FOO”. The length is limited to 64 characters to constrain the number of sub-identifiers needed for instance identification (and to minimize network traffic).

SQL Server Description

rdbmsDbLimitedResourceName identifies the segment name. The segment can be either a data or log segment.

The Subagent updates *rdbmsDbLimitedResourceName* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.4.1.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

➤ ***Note***

rdbmsDbLimtedResourceName is an index of *rdbmsDbLimtedResourceEntry.*

rdbmsDbLimitedResourceID

RDBMS MIB (RFC1697) Description

The ID of the resource which may be described in some other MIB (e.g., an enterprise-specific MIB module). If there is no ID for this *rdbmsDbLimitedResourceName*, attempts to access this object will return *noSuchName* (SNMPv1) or *noSuchInstance* (SNMPv2).

SQL Server Description

rdbmsDbLimitedResourceID is the ID of the resource which may be described in another MIB.

The Subagent updates *rdbmsDbLimitedResourceID* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.4.1.2

Data Type

ObjectID

Max-Access

Read-Only

Return Values

The Sybase IANA number: 1.3.6.1.4.1.897.

If there is no ID for the resource in *rdbmsDbLimitedResourceName*, the Subagent returns "no such name".

rdbmsDbLimitedResourceLimit

RDBMS MIB (RFC1697) Description

The maximum value the resource use may attain. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsDbLimtedReourceLimit identifies the maximum size of a segment.

The Subagent updates *rdbmsDbLimtedReourceLimit* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.4.1.3

Data Type

Integer

Max-Access

Read-Only

Return Values

An integer from 0 to 2147483647. The return value is either megabytes or kilobytes. See "rdbmsDbInfoSizeUnits" on page 4-37 for information on the unit of measure the Subagent is using.

rdbmsDbLimitedResourceCurrent

RDBMS MIB (RFC1697) Description

The current value for the resource.

SQL Server Description

rdbmsDbLimtedReourceCurrent identifies the amount of reserved space within a segment.

The Subagent updates *rdbmsDbLimtedReourceCurrent* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.4.1.4

Data Type

Integer

Max-Access

Read-Only

Return Values

An integer in range from 0 to 2147483647. The return value is either megabytes or kilobytes. See "rdbmsDbInfoSizeUnits" on page 4-37 for information on the unit of measure the Subagent is using.

rdbmsDbLimitedResourceHighwater

RDBMS MIB (RFC1697) Description

The maximum value of the resource seen since *applUpTime* was reset for the earliest server which has the database actively opened. If there are two servers with the database open, and the oldest one dies, the proper way to invalidate the value is by resetting *sysUpTime*.

SQL Server Description

rdbmsDbLimitedResourceHighwater identifies the highest observed usage of the resource since the Subagent started.

The Subagent updates *rdbmsDbLimitedResourceHighwater* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.4.1.5

Data Type

Integer

Max-Access

Read-Only

Return Values

Integer in range from 0 to 2147483647. Usage is returned in megabytes or kilobytes.

See Also

"rdbmsDbInfoSizeUnits" on page 4-37 for information on the unit of measure SQL Server SNMP Subagent is using.

rdbmsDbLimitedResourceFailures

RDBMS MIB (RFC1697) Description

The number of times the system wanted to exceed the limit of the resource since *applUpTime* was reset for the earliest server which has the database actively opened. If there are two servers with the DB open, and the oldest one dies, the proper way to invalidate the value is by resetting *sysUpTime*.

SQL Server Description

rdbmsDbLimitedResourceFailures identifies the number of times that the Subagent observed a current value reach the limit of the resource. The observation is within the time since the Subagent started.

The Subagent updates *rdbmsDbLimitedResourceFailures* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.4.1.6

Data Type

Counter32

Max-Access

Read-Only

Return Values

Integer in range from 0 to 2147483647.

rdbmsDbLimitedResourceDescription

RDBMS MIB (RFC1697) Description

A description of the resource and the meaning of the integer units used for Limit, Current, and Highwater. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsDbLimitedResourceDescription describes the name of the segment.

The Subagent updates *rdbmsDbLimitedResourceDescription* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.4.1.7

Data Type

DisplayString

Max-Access

Read-Only

Return Values

The name of the segment.

rdbmsSrvTable Objects

rdbmsSrvTable (1.3.6.1.2.1.39.1.5) identifies the database servers running or installed on a host. This information includes the:

- Name of the vendor that produced the server
- Contact person for the server
- Product name of the server

The following sections describe the objects in *rdbmsSrvTable*.

rdbmsSrvEntry

RDBMS MIB (RFC1697) Description

An entry for a single database server. A server is an independent entity that provides access to one or more databases. Failure of one does not affect access to databases through any other servers. There might be one or more servers providing access to a database. A server may be a 'process' or collection of 'processes', as interpreted by the product.

SQL Server Description

rdbmsSrvEntry identifies a single SYBASE SQL Server.

Object Identifier

1.3.6.1.2.1.39.1.5.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

rdbmsSrvPrivateMibOID

RDBMS MIB (RFC1697) Description

The authoritative identification for the private MIB for this server, presumably based on the vendor, e.g., {enterprises 111 <optional subidentifiers>} for Oracle servers, {enterprises 757 <optional subidentifiers>} for Ingres servers, {enterprises 897 <optional subidentifiers>} for Sybase servers, etc. If no OBJECT IDENTIFIER exists for the private MIB, attempts to access this object will return *noSuchName* (SNMPv1) or *noSuchInstance* (SNMPv2).

SQL Server Description

rdbmsSrvPrivateMibOID identifies the private MIB for the server, based on vendor. This is the Sybase IANA number.

The Subagent updates *rdbmsSrvPrivateMibOID* upon initialization or connection to a server.

Object Identifier

1.3.6.1.2.1.39.1.5.1.1

Data Type

ObjectID

Max-Access

Read-Only

Return Value

The Sybase IANA number: 1.3.6.1.4.1.897

rdbmsSrvVendorName

RDBMS MIB (RFC1697) Description

The name of the vendor whose RDBMS manages this database, for informational purposes.

SQL Server Description

rdbmsSrvVendorName identifies the name of the vendor whose RDBMS manages this database.

The Subagent updates *rdbmsSrvVendorName* upon initialization or connection to a server.

Object Identifier

1.3.6.1.2.1.39.1.5.1.2

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Sybase Inc.

rdbmsSrvProductName

RDBMS MIB (RFC1697) Description

The product name of this server. This is normally the vendor's formal name for the product, in product specific format.

SQL Server Description

rdbmsSrvProductName identifies the product name of the server.

The Subagent updates *rdbmsSrvProductName* upon initialization or connection to a server.

Object Identifier

1.3.6.1.2.1.39.1.5.1.3

Data Type

DisplayString

Max-Access

Read-Only

Return Value

SQL Server

rdbmsSrvContact

RDBMS MIB (RFC1697) Description

The textual identification of the contact person for this managed server, together with information on how to contact this person. Note: if there is no active server associated with this object, an agent may need to keep this in other persistent storage, e.g., a configuration file. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsSrvContact identifies the full name of the contact person for the managed server with information on how to contact the person.

You can set this OID from the network manager. If you do not enter a value, the Subagent uses the full name assigned to the "sa" login. The Subagent updates *rdbmsSrvContact* upon initialization or connection to a server.

Object Identifier

1.3.6.1.2.1.39.1.5.1.4

Data Type

DisplayString

Max-Access

Read-Write

Return Values

The full name of the contact person or the full name of "sa" login. SQL Server release 4.9.2 returns only the login name.

rdbmsSrvInfoTable Objects

rdbmsSrvInfoTable (1.3.6.1.2.1.39.1.6) identifies additional information about SQL Servers which are up and running. Entries in this table correspond to entries in *applTable*. This information includes:

- Date and time each SQL Server was started
- Total number of completed transactions
- Total number of disk reads and writes
- Total number of network packet read and writes
- Largest number of simultaneous users
- Maximum number of user connections allowed

The following sections describe the objects *rdbmsSrvInfoTable*.

rdbmsSrvInfoEntry

RDBMS MIB (RFC1697) Description

Information that must be present for a single "up" database server, with visibility determined by the value of the corresponding *applOperStatus* object. If an instance of *applOperStatus* is not up(1), then attempts to access corresponding instances in this table may result in either *noSuchName* (SNMPv1) or *noSuchInstance* (SNMPv2) being returned by the agent.

SQL Server Description

rdbmsSrvInfoEntry identifies information for databases that are up and running.

Object Identifier

1.3.6.1.2.1.39.1.6.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

rdbmsSrvInfoStartupTime

RDBMS MIB (RFC1697) Description

The date and time at which this server was last started.

SQL Server Description

rdbmsSrvInfoStartupTime identifies the date and time when the SQL server was started.

The Subagent updates *rdbmsSrvInfoStartupTime* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.1

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Date and time when the server last started. This return value consists of ten fields in the following format:

Table 4-3: Date and time format

Field	Octets	Contents	Range
1	1-2	Year	0 - 65536
2	3	Month	1 - 12
3	4	Day	1 - 31
4	5	Hour	0 - 23
5	6	Minutes	0 - 59
6	7	Seconds	0 - 60 (Use 60 for leap-second)
7	8	Deci-seconds	0 - 9

Table 4-3: Date and time format (continued)

Field	Octets	Contents	Range
8	9	Direction from UTC (Coordinated Universal Time)	+ or -
9	10	Hours from UTC	0 - 11
10	11	Minutes from UTC	0 - 59

For example, the date and time Tuesday May 26, 1993 at 1:30:15 PM EDT would appear as:

```
1992-5-26,13:30:15.0,-4:0
```

If the offset from UTC is unknown, only the first seven fields are given. If the database has never been backed up, then the Subagent returns:

```
1900-1-1,00:00:0.0,0:0
```

rdbmsSrvInfoFinishedTransactions

RDBMS MIB (RFC1697) Description

The number of transactions visible to this server that have been completed by either commit or abort. Some database operations, such as read-only queries, may not result in the creation of a transaction.

SQL Server Description

rdbmsSrvInfoFinishedTransactions identifies the number of transactions that have been completed by either commit or rollback since the server was started.

The Subagent updates *rdbmsSrvInfoFinishedTransactions* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.2

Data Type

Gauge32

Max-Access

Read-Only

Return Values

The total number of transactions completed. For SYBASE SQL Server release 4.9.x, the Subagent returns 0.

rdbmsSrvInfoDiskReads

RDBMS MIB (RFC1697) Description

The total number of reads of database files issued to the operating system by this server since startup. Numbers are not comparable between products. What constitutes a read and how it is accounted is product-specific.

SQL Server Description

rdbmsSrvInfoDiskReads identifies the total number of disk reads by the server that have occurred since the server was started.

The Subagent updates *rdbmsSrvInfoDiskReads* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.3

Data Type

Counter32

Max-Access

Read-Only

Return Values

Number of reads

rdbmsSrvInfoLogicalReads

RDBMS MIB (RFC1697) Description

The total number of logical reads of database files made internally by this server since startup. The values of this object and those of *rdbmsSrvInfoDiskReads* reveal the effect of caching on read operation. Numbers are not comparable between products, and may only be meaningful when aggregated across all servers sharing a common cache.

SQL Server Description

This information is not available.

Object Identifier

1.3.6.1.2.1.39.1.6.1.4

Data Type

Counter32

Max-Access

Read-Only

Return Values

Zero (0).

rdbmsSrvInfoDiskWrites

RDBMS MIB (RFC1697) Description

The total number of writes to database files issued to the operating system by this server since startup. Numbers are not comparable between products.

SQL Server Description

rdbmsSrvInfoDiskWrites identifies the total number of disk writes that have occurred since the server was started.

The Subagent updates *rdbmsSrvInfoDiskWrites* each time it collects information during a ShortSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.5

Data Type

Counter32

Max-Access

Read-Only

Return Values

Number of writes.

rdbmsSrvInfoLogicalWrites

RDBMS MIB (RFC1697) Description

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.

SQL Server Description

rdbmsSrvInfoLogicalWrites identifies the total number of disk writes that have occurred since the server was started.

The Subagent updates *rdbmsSrvInfoLogicalWrites* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.6

Data Type

Counter32

Max-Access

Read-Only

Return Values

Number of writes.

rdbmsSrvInfoPageReads

RDBMS MIB (RFC1697) Description

The total number of pages in database files read by this server since startup. 'Pages' are product specific units of disk I / O operations. This value, along with *rdbmsSrvInfoDiskReads*, reveals the effect of any grouping read-ahead that may be used to enhance performance of some queries, such as scans.

SQL Server Description

This information is not available.

Object Identifier

1.3.6.1.2.1.39.1.6.1.7

Data Type

Counter32

Max-Access

Read-Only

Return Values

Zero (0).

rdbmsSrvInfoPageWrites

RDBMS MIB (RFC1697) Description

The total number of pages in database files written by this server since startup. Pages are product-specific units of disk I/O. This value, with *rdbmsSrvInfoDiskWrites*, shows the effect of write strategies that collapse logical writes of contiguous pages into single calls to the operating system.

SQL Server Description

This information is not available.

Object Identifier

1.3.6.1.2.1.39.1.6.1.8

Data Type

Counter32

Max-Access

Read-Only

Return Values

Zero (0).

rdbmsSrvInfoDiskOutOfSpaces

RDBMS MIB (RFC1697) Description

The total number of times the server has been unable to obtain disk space that it wanted, since server startup. This would be inspected by an agent on receipt of an *rdbmsOutOfSpace* trap.

SQL Server Description

rdbmsSrvInfoDiskOutOfSpaces identifies the number of times the Subagent has observed the unreserved disk space go to zero since server startup.

The Subagent updates *rdbmsSrvInfoDiskOutOfSpaces* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.9

Data Type

Counter32

Max-Access

Read-Only

Return Values

An integer indicating the number of times the Subagent has observed the unreserved disk space go to zero since server startup.

rdbmsSrvInfoHandledRequests

RDBMS MIB (RFC1697) Description

The total number of requests made to the server on inbound associations. The meaning of "requests" is product specific, and is not comparable between products. This is intended to encapsulate high level semantic operations between clients and servers, or between peers. For instance, one request might correspond to a "select" or an "insert" statement. It is not intended to capture disk I/O described in *rdbmsSrvInfoDiskReads* and *rdbmsSrvInfoDiskWrites*.

SQL Server Description

This information is not available.

Object Identifier

1.3.6.1.2.1.39.1.6.1.10

Data Type

Counter32

Max-Access

Read-Only

Return Values

Zero (0).

rdbmsSrvInfoRequestRecvs

RDBMS MIB (RFC1697) Description

The number of receive operations made processing any requests on inbound associations. The meaning of operations is product specific, and is not comparable between products. This is intended to capture lower-level I/O operations than shown by HandledRequests, between clients and servers, or between peers. For instance, it might roughly correspond to the amount of data given with an "insert" statement. It is not intended to capture disk I/O described in *rdbmsSrvInfoDiskReads* and *rdbmsSrvInfoDiskWrites*.

SQL Server Description

rdbmsSrvInfoRequestRecvs identifies the number of input packets read by the server since it was started.

The Subagent updates *rdbmsSrvInfoRequestRecvs* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.11

Data Type

Counter32

Max-Access

Read-Only

Return Values

The number of network packets received by the server since it was started.

rdbmsSrvInfoRequestSends

RDBMS MIB (RFC1697) Description

The number of send operations made processing requests handled on inbound associations. The meaning of operations is product specific, and is not comparable between products.This is intended to capture lower-level I/O operations than shown by HandledRequests, between clients and servers, or between peers. It might roughly correspond to the number of rows returned by a "select" statement. It is not intended to capture disk I/O described in DiskReads.

SQL Server Description

rdbmsSrvInfoRequestSends identifies the number of network packets the server has sent since it was started.

The Subagent updates *rdbmsSrvInfoRequestSends* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.12

Data Type

Counter32

Max-Access

Read-Only

Return Values

The number of network packets the server has sent since it started.

rdbmsSrvInfoHighwaterInboundAssociations

RDBMS MIB (RFC1697) Description

The greatest number of inbound associations that have been simultaneously open to this server since startup.

SQL Server Description

rdbmsSrvInfoHighwaterInboundAssociations identifies the largest number of user connections that have been simultaneously connected to a server since startup.

The Subagent updates *rdbmsSrvInfoHighwaterInboundAssociations* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.13

Data Type

Gauge32

Max-Access

Read-Only

Return Values

Highest number of simultaneous user connections observed since the server started.

rdbmsSrvInfoMaxInboundAssociations

RDBMS MIB (RFC1697) Description

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. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsSrvInfoMaxInboundAssociations identifies maximum number of simultaneous user connections allowed by SQL Server as determined from *master.dbo.syscurconfigs*.

The Subagent updates *rdbmsSrvInfoMaxInboundAssociations* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.6.1.14

Data Type

Gauge32

Max-Access

Read-Only

Return Values

Maximum number of user connections allowed.

rdbmsSrvParamTable Objects

rdbmsSrvParamTable (1.3.6.1.2.1.39.1.7) provides access to information on the configuration parameters for a server. The following sections describe the objects in *rdbmsSrvParamTable*.

rdbmsSrvParamEntry

RDBMS MIB (RFC1697) Description

An entry for a single configuration parameter for a server. Parameters with single values have a subindex value of one. If the parameter is naturally considered to contain a variable number of members of a class, e.g. members of the DBA user group, or tracepoints active in the server, then it must be presented as a set of rows. If, on the other hand, the parameter represents a set of choices from a class, e.g. the permissions on a file or the options chosen out of the set of all options allowed, AND is guaranteed to always fit in the 255 character length of a DisplayString, then it may be presented as a comma separated list with a subindex value of one. Zero may not be used as a subindex value. Entries for a server must be present if the value of the corresponding applOperStatus object is up(1). If an instance of applOperStatus is not up(1), then attempts to access corresponding instances in this table may result in either noSuchName (SNMPv1) or noSuchInstance (SNMPv2) being returned by the agent.

SQL Server Description

rdbmsSrvParamEntry identifies an entry for a single configuration parameter for a SQL Server. Sybase SQL Server configuration parameters do not have subindexes.

rdbmsSrvParamEntry is indexed by:

- *applIndex*
- *rdbmsSrvParamName*
- *rdbmsSrvParamSubIndex*

Object Identifier

1.3.6.1.2.1.39.1.7.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

rdbmsSrvParamName

RDBMS MIB (RFC1697) Description

The name of a configuration parameter for a server. This name is product-specific. The length is limited to 64 characters to constrain the number of sub-identifiers needed for instance identification (and to minimize network traffic).

SQL Server Description

rdbmsSrvParamName identifies the name of the SQL Server configuration parameter. The name can be from 1 to 64 characters long. This information comes from the comment column of *master.dbo.syscurconfigs*.

The Subagent updates *rdbmsSrvParamName* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.7.1.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

➤ ***Note***

rdbmsSrvParamName is an index of *rdbmsSrvParamEntry*.

rdbmsSrvParamSubIndex

RDBMS MIB (RFC1697) Description

The subindex value for this parameter. If the parameter is naturally considered to contain a variable number of members of a class, e.g. members of the DBA user group, or files which are part of the database, then it must be presented as a set of rows. If, on the other hand, the parameter represents a set of choices from a class, e.g. the permissions on a file or the options chosen out of the set of all options allowed, AND is guaranteed to always fit in the 255 character length of a *DisplayString*, then it may be presented as a comma separated list with a subindex value of one. Zero may not be used as a value.

SQL Server Description

rdbmsSrvParamSubIndex identifies a subindex for a configuration parameter. SQL Server configuration parameters do not have a subindex.

The Subagent updates *rdbmsSrvParamSubIndex* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.7.1.2

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

➤ ***Note***

rdbmsSrvParamSubINdex is an index of *rdbmsSrvParamEntry*.

rdbmsSrvParamID

RDBMS MIB (RFC1697) Description

The ID of the parameter which may be described in some other MIB. If there is no ID for this *rdbmsSrvParamName*, attempts to access this object will return noSuchName (SNMPv1) or noSuchInstance (SNMPv2).

SQL Server Description

This is the Sybase IANA number.

The Subagent updates *rdbmsSrvParamID* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.7.1.3

Data Type

ObjectID

Max-Access

Read-Only

Return Value

The Sybase IANA number: 1.3.6.1.4.1.897

rdbmsSrvParamCurrValue

RDBMS MIB (RFC1697) Description

The value for a configuration parameter now in effect, the actual setting for the server. While there may multiple values in the temporal domain of interest (for instance, the value to take effect at the next restart), this is the current setting. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsSrvParamCurrValue identifies the value for the configuration parameter currently in effect.

The Subagent updates *rdbmsSrvParamCurrValue* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.7.1.4

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Values are derived from *master.dbo.syscurconfig*.

rdbmsSrvParamComment

RDBMS MIB (RFC1697) Description

Annotation which describes the purpose of a configuration parameter or the reason for a particular parameter's setting. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsSrvParamComment describes the configuration parameter.

The Subagent updates *rdbmsSrvParamComment* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.7.1.5

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Comment text up to 64 bytes.

rdbmsSrvLimitedResourceTable Objects

rdbmsSrvLimitedResourceTable (1.3.6.1.2.1.39.1.8) identifies the limited resources (devices) for SQL Server. This information includes:

- Name of each SYBASE device
- Maximum size of each device
- Current space allocated on each device
- The maximum amount of space ever used on a device
- The number of times unallocated space was zero

The following sections describe the objects in *rdbmsSrvLimitedResourceTable*.

rdbmsSrvLimitedResourceEntry

RDBMS MIB (RFC1697) Description

An entry for a single limited resource kept by the server. A limited resource has maximum use determined by a parameter that might or might not changeable at run time, or visible in the *rbmsSrvParamTable*. Examples would be the number of available locks, or number of concurrent executions allowed in a server. Arrays of resources are supported through an integer subindex, which should have the value of one for single-instance names. Limited resources that are shared across servers or databases are best duplicated in this table across all servers accessing the resource.

SQL Server Description

rdbmsSrvLimitedResourceEntry identifies SYBASE SQL Server devices including size and usage. For SQL Server, the limited resource is the device space usage. *rdbmsSrvLimtedResourceEntry* is indexed by:

- *applIndex*
- *rdbmssrvLimitedResouceName*

Object Identifier

1.3.6.1.2.1.39.1.8.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

rdbmsSrvLimitedResourceName

RDBMS MIB (RFC1697) Description

The name of the resource, for instance “threads” or “semaphores”, or “buffer pages”.

SQL Server Description

rdbmsSrvLimitedResourceName identifies a SQL Server device name.

The Subagent updates *rdbmsSrvLimitedResourceName* each time it collects information during a LongSleep cycle. See “Setting Polling Intervals” on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.8.1.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

➤ ***Note***

rdbmsSrvLimtedResourceName is an index of *rdbmsSrvLimtedResourceEntry.*

rdbmsSrvLimitedResourceID

RDBMS MIB (RFC1697) Description

The ID of the resource which may be described in some other MIB. If there is no ID for this *rdbmsSrvLimitedResourceName*, attempts to access this object will return noSuchName (SNMPv1) or noSuchInstance (SNMPv2).

SQL Server Description

rdbmsSrvLimitedResourceID is the ID of the resource which may be described in another MIB. This is the Sybase IANA number.

The Subagent updates *rdbmsSrvLimitedResourceID* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.8.1.2

Data Type

ObjectID

Max-Access

Read-Only

Return Values

The Sybase IANA number: 1.3.6.1.4.1.897

rdbmsSrvLimitedResourceLimit

RDBMS MIB (RFC1697) Description

The maximum value the resource use may attain. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsSrvLimitedResourceLimit identifies the maximum limit of the resource.

The Subagent updates *rdbmsSrvLimitedResourceLimit* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.8.1.3

Data Type

Integer

Max-Access

Read-Only

Return Values

The size of the device in megabytes.

See Also

"rdbmsDbInfoSizeUnits" on page 4-37 for information on the unit of measure SQL Server SNMP Subagent is using.

rdbmsSrvLimitedResourceCurrent

RDBMS MIB (RFC1697) Description

The current value for the resource.

SQL Server Description

rdbmsSrvLimitedResourceCurrent identifies the current usage of the limited resource.

The Subagent updates *rdbmsSrvLimitedResourceCurrent* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.8.1.4

Data Type

Integer

Max-Access

Read-Only

Return Values

Space allocated to databases on this device.

See Also

"rdbmsDbInfoSizeUnits" on page 4-37 for information on the unit of measure SQL Server SNMP Subagent is using.

rdbmsSrvLimitedResourceHighwater

RDBMS MIB (RFC1697) Description

The maximum value of the resource seen since *applUpTime* was reset.

SQL Server Description

rdbmsSrvLimitedResourceHighwater identifies the largest amount of disk space used that the Subagent has observed.

The Subagent updates *rdbmsSrvLimitedResourceHighwater* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.8.1.5

Data Type

Integer

Max-Access

Read-Only

Return Values

Number of megabytes allocated to databases on this device.

See Also

"rdbmsDbInfoSizeUnits" on page 4-37 for information on the unit of measure SQL Server SNMP Subagent is using.

rdbmsSrvLimitedResourceFailures

RDBMS MIB (RFC1697) Description

The number of times the system wanted to exceed the limit of the resource since *applUpTime* was reset.

SQL Server Description

rdbmsSrvLimitedResourceFailures identifies the number of times the amount of free space on a device was zero. The limit is set in *rdbmsSrvLimitedResourceLimit*.

The Subagent updates *rdbmsSrvLimitedResourceFailures* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.8.1.6

Data Type

Counter32

Max-Access

Read-Only

Return Values

Number of times all of the space on a device has been allocated to databases.

rdbmsSrvLimitedResourceDescription

RDBMS MIB (RFC1697) Description

A description of the resource and the meaning of the integer units used for Limit, Current, and Highwater. Note that a compliant agent does not need to allow write access to this object.

SQL Server Description

rdbmsSrvLimitedResourceDescription describes the device name.

The Subagent updates *rdbmsSrvLimitedResourceDescription* each time it collects information during a LongSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.8.1.7

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Device name

rdbmsRelTable Objects

rdbmsRelTable (1.3.6.1.2.1.39.1.9) relates the databases and servers on a host. This information includes:

- One row for every database
- Status of the database
- The date and time the database became active

The following sections describe the objects in *rdbmsRelTable*.

rdbmsRelEntry

RDBMS MIB (RFC1697) Description

An entry relating a single database server to a single database to which it may provide access. The table is indexed first by the index of *rdbmsDbTable*, and then *rdbmsSrvTable*, so that all servers capable of providing access to a given database may be found by SNMP traversal operations (get-next and get-bulk). The makeup of this table depends on the product's architecture, e.g. if it is one server - many databases, then each server will appear n times, where n is the number of databases it may access, and each database will appear once. If the architecture is one database - many servers, then each server will appear once and each database will appear n times, where n is the number of servers that may be accessing it.

SQL Server Description

rdbmsRelEntry identifies a single database.

Object Identifier

1.3.6.1.2.1.39.1.9.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

rdbmsRelState

RDBMS MIB (RFC1697) Description

The state of this server's access to this database. Active(2) means the server is actively using the database. Available(3) means the server could use the database if necessary. Restricted(4) means the database is in some administratively determined state of less-than-complete availability. Unavailable(5) means the database is not available through this server. Other(1) means the database/server is in some other condition, possibly described in the vendor private MIB.

SQL Server Description

rdbmsRelState identifies the state of a server's access to a database.

The Subagent updates *rdbmsRelState* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.9.1.1

Data Type

Integer

Max-Access

Read-Only

Return Values

One or more of the following values:

- 1—Indicates that the server or database failed an upgrade or crashed on loading or is suspect. Some network managers display this value as "Other(1)".
- 2—active. Indicates that the server or database did not crash while loading, is not suspect, is not read-only, is not in "dbo use only" mode, and is not in "single-user" mode. The server is up and running and the Subagent is logged in. Some network managers display this value as "Active(2)".
- 3—available. The server could use the database if necessary. Some network managers display this value as "Available(3)".

- 4—restricted. Indicates that the server or database is read-only, in "dbo use only" mode, or in "single-user" mode. Some network managers display this value as "Restricted(4)".
- 5—unavailable. Indicates that the database name appears in the *snmp.ini* file but not in the *master.dbo.sysdatabases* file and / or that the Subagent cannot log into SQL Server. Some network managers display this value as "Unavailable(5)".

If more than one condition is true, the Subagent returns the appropriate values in the following order:

- 5 (Unavailable)
- 1 (Other)
- 4 (Restricted)
- 3 (Available)
- 2 (Active)

rdbmsRelActiveTime

RDBMS MIB (RFC1697) Description

The time the database was made active by the server. If an instance of *rdbmsRelState* is not active(2), then attempts to access the corresponding instance of this object may result in either noSuchName (SNMPv1) or noSuchInstance (SNMPv2) being returned by the agent.

SQL Server Description

rdbmsRelActiveTime identifies the time the database was activated by the server.

The Subagent updates *rdbmsRelActiveTime* each time it collects information during a ShortSleep cycle. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.1.9.1.2

Data Type

DisplayString

Max-Access

Read-Only

Return Values

Date and time when the database was made active by the server. This return value consists of ten fields in the following format:

Table 4-4: Date and time format

Field	Octets	Contents	Range
1	1-2	Year	0 - 65536
2	3	Month	1 - 12
3	4	Day	1 - 31
4	5	Hour	0 - 23
5	6	Minutes	0 - 59

Table 4-4: Date and time format (continued)

Field	Octets	Contents	Range
6	7	Seconds	0 - 60 (Use 60 for leap-second)
7	8	Deci-seconds	0 - 9
8	9	Direction from UTC (Coordinated Universal Time)	+ or -
9	10	Hours from UTC	0 - 11
10	11	Minutes from UTC	0 - 59

For example, the date and time Tuesday May 26, 1993 at 1:30:15 PM EDT would appear as:

```
1992-5-26,13:30:15.0,-4:0
```

If the offset from UTC is unknown, only the first seven fields are given. If the database has never been backed up, then the Subagent returns:

```
1900-1-1,00:00:0.0,0:0
```

the database became active.

rdbmsTraps Objects

rdbmsTraps (1.3.6.1.2.1.39.2) identifies two objects that describe traps that the Subagent generates:

- *rdbmsStateChange*
- *rdbmsOutOfSpace*

The following sections describe these objects.

rdbmsStateChange

RDBMS MIB (RFC1697) Description

An *rdbmsStateChange* trap signifies that one of the database server / databases managed by this agent has changed its rdbmsRelState in a way that makes it less accessible for use. For these purposes, both active(2) and available(3) are considered fully accessible. The state sent with the trap is the new, less accessible state.

SQL Server Description

rdbmsStateChange is a trap which occurs when a server or database has changed its state so that it is neither "active" nor "available".

The Subagent updates *rdbmsStateChange* during the ShortSleep polling interval. See "Setting Polling Intervals" on page 3-7 for more information.

Object Identifier

1.3.6.1.2.1.39.2.1

Data Type

Not applicable

Max-Access

Not applicable

Return Values

The Subagent automatically returns a trap containing the following information:

- Community string—the community string as specified in *snmpd.cnf*. See "Defining Trap Notifications" on page 2-5 for more information on the community string specification for traps.
- Enterprise authority—the value of *rdbmsMIB*, which is 1.3.6.1.2.1.39.
- Bindings—*rdbmsRelState.rdbmsDbIndex.applIndex=new_rel_state* where *new_rel_state* is the new state of the server or database.

rdbmsOutOfSpace

RDBMS MIB (RFC1697) Description

An *rdbmsOutOfSpace* trap signifies that one of the database servers managed by this agent has been unable to allocate space for one of the databases managed by this agent. Care should be taken to avoid flooding the network with these traps.

SQL Server Description

rdbmsOutOfSpace indicates that a server cannot allocate space for a database. When the Subagent observes that a segment is full, it generates a trap.

This data point is updated during seldom polls (between *LongSleep* intervals).

Object Identifier

1.3.6.1.2.1.39.2.2

Data Type

Not applicable

Max-Access

Not applicable

Return Values

The Subagent automatically returns a trap containing the following information:

- Community string—the community string as specified in *snmpd.cnf*. See "Defining Trap Notifications" on page 2-5 for more information on the community string specification for traps.
- Enterprise authority—the value of *rdbmsMIB*, which is 1.3.6.1.2.1.39.
- Bindings—*rdbmsSrvInfoDisOutOfSpace.applIndex=1 rdbmsSrvLimitedResourceFailures.rdbmsDbIndex.segment_name=1* where *segment_name* is the name of the affected segment.

Objects and Polling Intervals

During the ShortSleep and LongSleep polling intervals, the Subagent collects different information. The following two sections identify which objects the Subagent updates after a ShortSleep polling interval and which objects the Subagent updates after a LongSleep polling interval.

ShortSleep Polling Interval

During a ShortSleep polling interval, the Subagent updates the following OIDs:

- *applUptime*
- *applOperStatus*
- *applLastChange*
- *applInboundAssociations*
- *applAccumulatedInboundAssociations*
- *applLastInboundActivity*
- *rdbmsSrvInfoFinishedTransactions*
- *rdbmsSrvInfoDiskReads*
- *rdbmsSrvInfoDiskWrites*
- *rdbmsSrvInfoLogicalWrites*
- *rdbmsSrvInfoRequestRecvs*
- *rdbmsSrvInfoRequestSends*
- *rdbmsSrvInfoHighwaterInboundAssociations*
- *rdbmsRelState*
- *rdbmsRelActiveTime*
- *rdbmsStateChange*

LongSleep Polling Interval

During a LongSleep polling interval, the Subagent updates the following OIDs:

- *rdbmsDbPrivateMibOID*
- *rdbmsDbVendorName*

- *rdbmsDbName*
- *rdbmsDbContact*
- *rdbmsDbInfoSizeUnits*
- *rdbmsDbInfoSizeAllocated*
- *rdbmsDbInfoSizeUsed*
- *rdbmsDbInfoLastBackup*
- *rdbmsDbParamName*
- *rdbmsDbParamSubIndex*
- *rdbmsDbParamID*
- *rdbmsDbParamCurrValue*
- *rdbmsDbParamComment*
- *rdbmsDbLimitedResourceName*
- *rdbmsDbLimitedResourceID*
- *rdbmsDbLimitedResourceLimit*
- *rdbmsDbLimitedResourceCurrent*
- *rdbmsDbLimitedResourceHighwater*
- *rdbmsDbLimitedResourceFailures*
- *rdbmsDbLimitedResourceDescription*
- *rdbmsSrvInfoStartupTime*
- *rdbmsSrvInfoDiskOutofSpaces*
- *rdbmsSrvInfoMaxInboundAssociations*
- *rdbmsSrvParamName*
- *rdbmsSrvParamSubIndex*
- *rdbmsSrvParamID*
- *rdbmsSrvParamCurrValue*
- *rdbmsSrvParamComment*
- *rdbmsSrvLimitedResourceName*
- *rdbmsSrvLimitedResourceID*
- *rdbmsSrvLimitedResourceLimit*
- *rdbmsSrvLimitedResourceCurrent*
- *rdbmsSrvLimitedResourceHighwater*

- *rdbmsSrvLimitedResourceFailures*
- *rdbmsSrvLimitedResourceDescription*
- *rdbmsOutOfSpace*

5 Objects in the SYBASE MIB

The SYBASE MIB contains objects that allow you to:

- Modify the value of ShortSleep
- Modify the value of LongSleep
- Add servers to the monitoring list
- Drop servers from the monitoring list
- Change the passwords of logins

The SYBASE MIB is part of the Subagent distribution. Before your network manager can interact with these objects, you must import it into your network manager.

This chapter describes the objects in the SYBASE MIB. See your network manager documentation for information on importing MIBs. See "Modifying Object Values" on page 5-11 for more information on modifying object values from a network manager.

SYBASE MIB Objects

Table 5-1 contains the objects in the SYBASE MIB:

Table 5-1: SYBASE MIB objects

Object	Object Identifier	Data Type	Max-Access
sybShortSleep	1.3.6.1.4.1.897.3.1.1	*Integer*	Read-Write
sybLongSleep	1.3.6.1.4.1.897.3.1.2	*Integer*	Read-Write
sybApplModTable	1.3.6.1.4.1.897.3.1.3	Not applicable	Not accessible
sybApplModEntry	1.3.6.1.4.1.897.3.1.3.1	Not applicable	Not accessible
sybApplModIndex	1.3.6.1.4.1.897.3.1.3.1.1	*Integer*	Read-Write
sybApplIndex	1.3.6.1.4.1.897.3.1.3.1.2	*Integer*	Read-Write
sybApplModCMD	1.3.6.1.4.1.897.3.1.3.1.3	*Integer*	Read-Write
sybApplModNewPassword	1.3.6.1.4.1.897.3.1.3.1.4	*DisplayString*	Read-Write
sybApplModApplName	1.3.6.1.4.1.897.3.1.3.1.5	*DisplayString*	Read-Write

sybShortSleep

sybShortSleep allows you to modify the ShortSleep value that the Subagent uses to monitor the servers. You set this OID from the network manager. Its value can be from 10 to 1800 seconds. Sybase recommends that you use a value of at least 40 because if you set this value too low, you may experience poor response time from the Suabgent. This occurs because the Subagent is always busy contacting servers and collecting information. If you set *sybShortSleep* too high, the latency between a value changing and the time you see the change may be too slow.

See “Setting Polling Intervals” on page 3-7 for information on ShortSleep. See “Objects and Polling Intervals” on page 4-107 for information on the objects the Subagent monitors during a ShortSleep cycle.

➤ ***Note***

Periodically, the Subagent updates the initialization file *snmp.ini*. If you set *sybShortSleep*, the Subagent writes your new monitoring interval to *snmp.ini*.

Object Identifier

1.2.6.1.4.1.897.3.1.1

Data Type

Integer

Max-Access

Read-Write

Return Values

A value in seconds from 10 to 1800 that represents ShortSleep

sybLongSleep

sybLongSleep allows you to modify the LongSleep value that the Subagent uses to monitor the servers. You set this OID from the network manager. Its value can be from 1 to 20. Sybase recommends you use a value of at least 2. If the object information that the Subagent collects during a LongSleep poll is important to you, keep this value low. If the object information is not extremely important, use a higher value.

See "Setting Polling Intervals" on page 3-7 for more information on LongSleep. See "Objects and Polling Intervals" on page 4-107 for information on the objects the Subagent monitors during a LongSleep cycle.

➤ ***Note***

Periodically, the Subagent updates the initialization file *snmp.ini*. If you set *sybLongSleep*, the Subagent writes your new monitoring interval to *snmp.ini*.

Object Identifier

1.3.6.1.4.1.897.3.1.2

Data Type

Integer

Max-Access

Read-Write

Return Values

Integer from 1 to 20

sybApplModTable Objects

sybApplModTable is a one-row table that you use to modify the servers list that the Subagent is monitoring or to modify the passwords of the logins the Subagent uses. You modify the objects in *sybApplModTable* from the network manager. If you try to get information from the objects in *sybApplModTable*, the Subagent returns a row of asterisks (***).

The following sections describe the objects in *sybApplModTable*. See "Modifying snmp.ini" on page 3-6 for more information on the monitored server list. See "Modifying Object Values" on page 5-11 for more information on modifying objects from a network manager.

➤ ***Note***

Periodically, the Subagent updates the initialization file *snmp.ini*. If you set any of the objects *sybApplModTable*, the Subagent writes your new monitoring intervals to *snmp.ini*.

sybApplModEntry

sybApplModEntry provides access to the rows in the table.

Object Identifier

1.3.6.1.4.1.897.3.1.3.1

Data Type

Not applicable

Max-Access

Not accessible

Return Values

None

sybApplModIndex

Uniquely identifies the single row in the *sybApplModTable*.

Object Identifier

1.3.6.1.4.1.897.3.1.3.1.1

Data Type

Integer

Max-Access

Read-Write

Return Values

Always returns 1.

sybApplIndex

sybApplIndex is an integer that corresponds to the *applIndex* in the *applTable*. This value identifies a specific server. You set this OID from the network manager.

When you specify an operation using *sybApplModCMD*, the Subagent reads the value of *sybApplIndex* to determine which server entry to use. See “Modifying Object Values” on page 5-11 for more information.

Object Identifier

1.3.6.1.4.1.897.3.1.3.1.2

Data Type

Integer

Max-Access

Read-Write

Return Values

An integer representing the *applIndex* value assigned to a specific server.

sybApplModCMD

sybApplModCMD is an integer that specifies an operation you want to perform. When you set this object to a specific value, the Subagent performs the associated action:

- 1—adds a new row to the *applTable* for this new server. After you add a new row to *applTable*, the information is eventually propagated to all other tables as appropriate. In some network managers, this value appears as “addServer”.
- 2—drops a row from the *applTable* corresponding to the server. After you drop a row from *applTable*, the information is eventually dropped from all other appropriate tables. In some network managers, this value appears as “deleteServer”.
- 3—changes the password for that the Subagent uses to contact the server. In some network managers, this value appears as “changePassword”.

You set this OID from a network manager.

Object Identifier

1.3.6.1.4.1.897.3.1.3.1.3

Data Type

Integer

Max-Access

Read-Write

Return Values

Integer from 1 to 3.

sybApplModNewPassword

sybApplModNewPassword allows you to specify the password the Subagent is using to connect to a server. The password can be from 2 to 30 characters long. You set this OID from the network manager.

After you set this object, you use *sybApplModCMD* to change the password. When you use *sybApplModCMD*, the Subagent uses the value you have specified in *sybApplModNewPassword*. See "Modifying Object Values" on page 5-11 for more information.

Object Identifier

1.3.6.1.4.1.897.3.1.3.1.4

Data Type

DisplayString

Max-Access

Read-Write

Return Values

A row of asterisks (***).

sybApplModApplName

sybApplModApplName is the same as the value of *applName* in the *applTable*. This name represents a server and can be up to 30 characters in length. You set this OID from the network manager.

When you use *sybApplModCMD* to add or drop a server or to change a password, the Subagent uses the value of *sybApplModApplName*. See "Modifying Object Values" on page 5-11 for more information.

Object Identifier

1.3.6.1.4.1.897.3.1.3.1.5

Data Type

DisplayString

Max-Access

Read-Write

Return Values

Always returns the string "Server Name".

Modifying Object Values

sybApplModTable allows you to modify the list of servers the Subagent is monitoring and the password the Subagent uses to connect to a server. You perform all of these actions from the network manager.

To add a server to the monitored server list:

1. Set *sybApplIndex* to the index number you want to assign to the server.
2. Set *sybApplModApplName* to the name of the server you are adding.
3. Set *sybApplModCMD* to 1 (addServer).

To drop a server from the monitored server list:

1. Set *sybApplIndex* to the index number of the server you want to drop.
2. Set *sybApplModApplName* to the name of the server you are dropping.
3. Set *sybApplModCMD* to 2 (deleteServer).

To change a password:

1. Set *sybApplIndex* to the index number of the server whose password you want to change.
2. Set *sybApplModApplName* to the name of the server you are modifying.
3. Set *sybApplModNewPassword* to the new password you want to use.
4. Set *sybApplModCMD* to 3 (changePassword).

A Messages

This appendix contains error messages generated by the Subagent. There are four categories of messages:

- **Configuration**—identifies configuration problems. Messages in this category begin with the letter "C".
- **Internal**—identifies internal errors or system allocation errors. Messages in this category begin with the letter "E". If you encounter internal errors, you may need to contact Sybase Technical Support.
- **Initialization**—identifies problems with the initialization file. Messages in this category begin with the letter "S".
- **Informational**—identifies situations of which you may have interest, such as a server shutting down. Messages in this category begin with the letter "I".
- **Temporary**—identifies temporary errors. Unless these errors repeat, you can safely ignore them. Messages in this category begin with the letter "T".

The following sections describe the message format and the Subagent messages.

Message Format

Every message has format:

```
sybagt: date_and_time
.... message_number: message_text
```

where *date_and_time* identifies when the Subagent created the message, *message_number* uniquely identifies the error, and *message_text* is the text of the message. For example:

```
sybagt: June 6, 1995
.... C1001: $SYBASE environment variable is not
defined
```

Is a configuration error describing the SYBASE environmental variable.

Messages

C1001

Message

```
C1001: $SYBASE environment variable is not defined
```

Description

When you started the Subagent, the SYBASE environmental variable was not set. This error causes the Subagent to terminate.

Work-Around

Set SYBASE and restart the Subagent.

C1002

Message

```
C1002: $SYBSNMP  environment variable is not
defined
```

Description

When you started the Subagent, the SYBSNMP environmental variable was not set. This error causes the Subagent to terminate.

Work-Around

Set SYBSNMP and restart the Subagent.

C1003

Message

```
C1003: cs_ctx_alloc() failed - Could be mixed
versions of CT-Library
```

Description

The Subagent could not allocate required resources. The most common reason for this error is that you are using mixed versions of

Open Client™ Client-Library™. This error causes the Subagent to terminate.

Work-Around

When you install the Subagent, you install Open Client Client-Library under the *$SYBSNMP* directory. If you encounter this error, you should reinstall the Subagent. When the installation procedure prompts you to overwrite files, respond no. If you respond yes, the installation procedure may overwrite existing configuration files.

C1004

Message

```
C1004: ct_init() failed
```

Description

The Subagent could not initialize required resources. The most common reason for this error is that you are using mixed versions of Open Client Client-Library. This error causes the Subagent to terminate.

Work-Around

When you install the Subagent, you install Open Client Client-Library under the *$SYBSNMP* directory. If you encounter this error, you should reinstall the Subagent. When the installation procedure prompts you to overwrite files, respond no. If you respond yes, the installation procedure may overwrite existing configuration files.

C1005

Message

C1005: Cannot open interfaces file: *interfaces_file*

Description

This is a fatal error because the Subagent could not open the interfaces file.

Work-Around

If you used the **-I** flag, make sure that the path is correct and that the directory contains a valid interfaces file. If you did not use the **-I** flag,

make sure that the *$SYBASE* directory contains a valid interfaces file. Also, check the file permissions. You typically start the Subagent as "root", therefore "root" must have read permission. Although "root" generally has read permission on files, some NFS mounted file systems do not recognize "root".

C1006

Message

```
C1006: Operating system error:  <message>
```

Description

This message appears after message I3003. This problem often occurs when there are networking errors or a failure to locate some needed resource.

Work-Around

None.

C1007

Message

```
C1007:  rdbmsSrvEntry duplicate attempted
```

Description

Attempt to add two servers with the same *applIndex*. Because of this, the Subagent cannot make an entry in the *rdbmsSrvTable*.

Work-Around

Check *snmp.ini* and make sure that every *applIndex* is unique.

C1008

Message

```
C1008: DbIndex # is a duplicate
```

Description

DbIndexes must be unique for a given host machine. The Subagent attempts to keep all *DbIndexes* unique and this error should not occur.

Work-Around

None.

C1009

Message

```
C1009: DbIndex # is a duplicate
```

Description

DbIndexes must be unique for a given host machine. The Subagent attempts to keep all *DbIndexes* unique and this error should not occur.

Work-Around

None.

C1010

Message

C1010: *ini_file*: *num* error encountered

Description

num identifies the total number of errors encountered while parsing the initialization file identified by *ini_file* (typically *snmp.ini*).

Work-Around

Identify and correct each initialization file error.

C1011

Message

```
C1011: ini_file, num applIndex value must be
between 1 & 99999
```

Description

ini_file identifies the initialization file and *num* is the offending number. *ApplIndex* values must be in the range of 1 to 99999.

Work-Around

Shut down the Subagent and change the *applIndex* to a value within this range. See "Format of snmp.ini" on page 3-6 for more information on server entries in the initialization file.

C1012

Message

```
C1012: Cannot open file tmp_ini_file
```

Description

The Subagent cannot open the temporary initialization file specified by *tmp_ini_file*. The Subagent uses this file for the temporary storage of information. This is not a fatal error, but the Subagent cannot update the initialization file until you correct the error.

Work-Around

$SYBASE/snmp.ini.tmp is the default temporary initialization file. Check this file or use the **-c** flag to specify a different temporary file.

C1013

Message

```
C1013: InitSubagent:SNMP initialization error. Is
Master agent installed?
```

Description

The Subagent can not register itself with the master agent.

Work-Around

You must start the master agent before starting the Subagent. Make sure that the master agent is running. If it is not, stop the Subagent, start the master agent, and restart the Subagent.

C1014

Message

C1014: *server_name* is not in interfaces file.

Description

The server *server_name* is in the initialization file but it is not in the interfaces file. The Subagent does not monitor servers if it cannot find an entry for it in the interfaces file.

Work-Around

Check the name of the server in the initialization file and in the interfaces file. Make sure that they match.

C1015

Message

```
C1015: SNMP Subagent cannot login to: server_name -
login name: login_name or password is bad
```

Description

The Subagent is not able to login to the server using the *login_name* login and password.

Work-Around

Check that the *login_name* login and password are correct and that the server has a valid account with that name and password.

C1016

Message

```
C1016: ct_cmd_alloc failed
```

Description

This is an Open Client Client-Library error.

Work-Around

See your Open Client Client-Library documentation for more information.

C1017

Message

```
C1017: server_name applIndex = num is a
duplicate.  Row in applTable not created
```

Description

The Subagent cannot monitor the server *server_name* in the initialization file because its *applIndex* number is not unique.

Work-Around

Shut down the Subagent and modify the server entry in the initialization file so that it has a unique *applIndex*.

➤ ***Note***

A special case exists if you are running a sub-agent from another vendor which also supports the application table. Make sure that the *applIndex* numbers are unique for all entries. If there are duplicates among different sub-agents, the error will go undetected and the Subagent cannot monitor some servers.

C1018

Message

```
C1018: ini_file, num DbIndex value must be between
1 & 255
```

Description

There is an invalid *DbIndex* number in the initialization file *ini_file*.

Work-Around

Normally, the Subagent makes sure that each *DbIndex* value is unique. If this problem persists, shut down the Subagent, edit the initialization file, and remove all database entries. Then, restart the Subagent.

C1019

Message

```
C1019: LongSleep: num is out of limits
```

Description

The LongSleep time specified by *num* is outside allowable limits. The Subagent uses the default value.

Work-Around

None.

C1020

Message

```
C1020: ShortSleep: num is out of limits
```

Description

The ShortSleep time specified by *num* is outside allowable limits. The Subagent uses the default value.

E2001

Message

```
E2001: applIndex_value, rdbmsSrvParamName_value
was not created in rdbmsSrvParamTable
```

Description

The Subagent cannot create a row in the *rdbmsSrvParamTable* because of insufficient memory or because a duplicate entry was attempted.

Work-Around

None.

E2002

Message

```
E2002: applIndex_value,
rdbmsSrvLimitedResouceName_value was not created
in rdbmsSrvLimitedResouceTable
```

Description

The Subagent did not create a row in the *dbmsSrvLimitedResouceTable* because of insufficient memory or because it is a duplicate.

Work-Around

None.

E2003

Message

```
E2003: rdbmsDbIndex_value,rdbmsParamName_value was
not created in rdbmsDbParamTable
```

Description

A row in the *dbmsDbParamTable* was not created either because of insufficient memory or because it is a duplicate.

Work-Around

None.

E2004

Message

```
E2004: rdbmsDbIndex_value, segment_name_value was
not created in rdbmsDbLimitedResourceTable
```

Description

A row in the *dbmsDbLimitedResouceTable* was not created either because of insufficient memory or because it is a duplicate.

Work-Around

None.

E2005

Message

E2005: *applIndex_value* was not created in *RdbmsSrvTable*

Description

A row in the *dbmsSrvTable* was not created either because of insufficient memory or because it is a duplicate.

Work-Around

None.

E2006

Message

E2006: *applIndex_value* row was not created in *RdbmsSrvInfoTable*

Description

A row in the *dbmsSrvInfoTable* was not created either because of insufficient memory or because it is a duplicate.

Work-Around

None.

E2007

Message

E2007: *DbIndex_value*,*applIndex_value* was not created in *rdbmsDbTable*; in *dbPinger*

Description

A row in the *rdbmsDbTable* was not created either because of insufficient memory or because it is a duplicate.

Work-Around

None.

E2008

Message

```
E2008: CT- Library failed to install client
callback handler
```

Description

Open Client Client-Library failed to install callback handler. This is a fatal error.

Work-Around

See your Open Client Client-Library documentation for more information.

E2009

Message

```
E2009: CT- Library failed to install server
callback handler
```

Description

Open Client Client-Library failed to install callback handler. This is a fatal error.

Work-Around

See your Open Client Client-Library documentation.

E2010

Message

```
E2010: ct_config error failed when setting timeouts
```

Description

This is an Open Client Client-Library error. This error causes the Subagent to terminate.

Work-Around

See your Open Client Client-Library documentation for more information.

E2011

Message

```
E2011: Recursion detected in message handler
```

Description

If you can reproduce this message, it is a bug. This is a fatal error.

Work-Around

None. Save the log file messages and call Sybase Technical Support.

E2012

Message

E2012: *ct_res_info* failed.

Description

Open Client Client-Library error while retrieving results. Could be a failed connection.

Work-Around

See your Open Client Client-Library documentation for more information.

E2013

Message

```
E2013: bind_data: too few columns returned
```

Description

The Subagent was expecting more columns in the managed server. Perhaps this is not a supported SQL Server.

Work-Around

None.

E2014

Message

```
E2014: ct_bind failed
```

Description

Encountered Open Client Client-Library error while retrieving results. Could be a failed connection.

Work-Around

See your Open Client Client-Library documentation for more information.

E2015

Message

E2015: *ct_cancel* failed.

Description

The Subagent received an unexpected result from the server and attempted to cancel the query. The cancel failed.

Work-Around

None.

E2016

Message

E2016: *ct_results* failed

Description

This is an Open Client Client-Library error. A command to the server or a result set from the server failed. This could be due to a server going down or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

E2017

Message

`E2017: rdbmsSrvEntry allocation error.`

Description

An allocation operation failed. Probably an insufficient memory problem.

Work-Around

None.

E2018

Message

`E2018: Unable to create new ApplEntry`

Description

An allocation operation failed. Probably an insufficient memory problem.

Work-Around

None.

E2019

Message

`E2019: ct_con_alloc failed`

Description

This is an Open Client Client-Library error.

Work-Around

See your Open Client Client-Library documentation.

E2020

Message

`E2020: ct_con_props CS_USERNAME failed`

Description

This is an Open Client Client-Library error.

Work-Around

See your Open Client Client-Library documentation.

E2021

Message

`E2021: ct_con_props CS_PASSWORD failed`

Description

This is an Open Client Client-Library error.

Work-Around

See your Open Client Client-Library documentation for more information.

E2022

Message

`E2022: ct_con_props CS_APPNAME failed`

Description

This is an Open Client Client-Library error.

Work-Around

See your Open Client Client-Library documentation for more information.

E2023

Message

`E2023: ct_con_props CS_HOSTNAME failed`

Description

This is an Open Client Client-Library error.

Work-Around

See your Open Client Client-Library documentation for more information.

E2024

Message

`E2024: ct_con_props CS_PACKETSIZE failed`

Description

This is an Open Client Client-Library error.

Work-Around

See your Open Client Client-Library documentation for more information.

E2027

Message

`E2027: Invalid OperStatus`

Description

Internal logic error.

Work-Around

None.

E2028

Message

`E2028: A connection has terminated unexpectedly rdbmsSrvInfoEntry`

Description

A connection has unexpectedly terminated.

Work-Around

None.

E2029

Message

`E2029: ct_command failed in rdbmsSrvInfoEntry`

Description

This is an Open Client Client-Library error that could be due to an unstable connection or temporary inconsistency.

Work-Around

See your Open Client Client-Library documentation for more information.

E2030

Message

```
E2030: A connection has unexpectedly terminated in
rdbmsSrvInfoEntry.seldom
```

Description

A connection has unexpectedly terminated.

Work-Around

None.

E2031

Message

E2031: *ct_command* failed in *rdbmsDbEntry_seldom*

Description

This is an Open Client Client-Library error that could be due to an unstable connection or temporary inconsistency.

Work-Around

See your Open Client Client-Library documentation for more information.

E2032

Message

E2032: *cs_loc_alloc* failed during initialization

Description

This is an Open Client Client-Library error.

Work-Around

See your Open Client Client-Library documentation for more information.

E2033

Message

```
E2033: cs_locale failed during initialization.
```

Description

This is an Open Client Client-Library error.

Work-Around

See your Open Client Client-Library documentation for more information.

I3001

Message

```
I3001: Shutting down.
```

Description

Subagent has terminated.

Work-Around

None.

I3002

Message

```
I3002: server_name server timed out after number
seconds
```

Description

The named server *server_name* did not respond within *number* seconds.

Work-Around

The Subagent will retry the command.

I3003

Message

```
I3003: Client msg: LAYER = (num) ORIGIN = (num)
SEVERITY = (num) NUMBER = (num)
error_message
```

Description

This is a general trace routine which reports all Open Client Client-Library client-side error messages, where *error_message* is the actual string.

Work-Around

See your Open Client Client-Library documentation for more information.

I3004

Message

```
I3004: Terminating a connection
```

Description

Terminating a connection to a server, because the server is giving improper responses. For example, the server has shut down or the connection has been broken.

Work-Around

The agent will attempt a new connection at a later time.

I3005

Message

```
I3005: Server message number=num ,Severity=num,
Line=num"
```

Description

A message / error from the server.

Work-Around

See your server documentation for more information.

I3006

Message

```
I3006: From server server_name: server_message
```

Description

This is a follow on message from I3005 and contains the *server_message*.

Work-Around

Consult the SQL Server (or open server application) for the meaning of these messages.

I3008

Message

```
I3008: DbIndex DbIndex_value Database was dropped
```

Description

A database was dropped from the server.

Work-Around

None.

S4001

Message

```
S4001:snmp.ini_path: num Syntax error -- Missing']'
section_name
```

Description

A closing brace was missing at the end of a *section_name* where *section_name* is the name of the section. The error occurred on line *num* in the file.

Work-Around

Stop the Subagent and edit *snmp.ini*. Then, restart the Subagent.

S4002

Message

```
S4002: snmp.ini_path: num Syntax error -- Section
name is empty
section_name
```

Description

The section *section_name* was found, but there was not a valid section name within the brackets. The error occurred on line *num* in *snmp.ini*.

Work-Around

Stop the Subagent and edit *snmp.ini*. Then, restart the Subagent.

S4003

Message

```
S4003: snmp.ini_path: num Syntax error --Section
name contains white space
section_name
```

Description

You cannot have white space between the brackets of a section. The error occurred on line *num* in *snmp.ini* in the *section_name* section.

Work-Around

Stop the Subagent and edit *snmp.ini*. Then, restart the Subagent.

S4004

Message

```
S4004: snmp.ini_path: num Syntax error -- keyword
not in a section|
ini_entry
```

Description

The keyword *ini_entry* was found outside of the appropriate section. The error occurred on line *num* in *snmp.ini*.

Work-Around

Stop the Subagent and edit *snmp.ini*. Then, restart the Subagent.

S4005

Message

```
S4005: snmp.ini_path: num Syntax error -- keyword
not found
invalid_keyword
```

Description

There is an invalid keyword in a section. The error occurred on line *num* in *snmp.ini*.

Work-Around

Stop the Subagent and edit *snmp.ini*. Then, restart the Subagent.

S4006

Message

```
S4006: snmp.ini_path: num Syntax error -- no
keyword found
ini_entry
```

Description

The Subagent found and entry that did not begin with a keyword. The error occurred on line *num* in *snmp.ini*.

Work-Around

Stop the Subagent and edit *snmp.ini*. Then, restart the Subagent.

S4007

Message

```
S4007: snmp.ini_path: num Syntax error -- no '='
found following keyword
ini_entry
```

Description

The Subagent found a keyword but the it was missing the "=". The error occurred on line *num* in *snmp.ini*.

Work-Around

Stop the Subagent, edit *snmp.ini*, and then restart the Subagent.

S4008

Message

```
S4008: snmp.ini_path: num Syntax error -- no
'value' found following '='
ini_entry
```

Description

The entry *ini_entry* did not have a value after the "=". The error occurred on line *num* in *snmp.ini*.

Work-Around

Stop the Subagent and edit *snmp.ini*. Then, restart the Subagent.

S4009

Message

```
S4009: snmp.ini_path: num Syntax error --
Unrecognized keyword:
keyword
```

Description

The Subagent found the *keyword* keyword in the file but it is not a valid keyword. The error occurred on line *num* in *snmp.ini*.

Work-Around

Stop the Subagent and edit *snmp.ini*. Then, restart the Subagent.

S4010

Message

```
S4010: Unable to open configuration file snmp.ini_
file_name
```

Description

The Subagent cannot open *snmp.ini*. The default file is in the *$SYBASE* directory. This error causes the Subagent to terminate.

Work-Around

You can override the file name with the **-c** option when starting the Subagent.

T5001

Message

T5001: *ct_send* error in *update_SrvParam*

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5002

Message

T5002: *ct_send* error in *update_SrvLimitedResource*

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5003

Message

T5003: *ct_send* error from *update_DbParam*

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5004

Message

```
T5004: Database_name was empty in
update_DbLimitedResource
```

Description

An *rdbmsDbEntry* had no database name.

Work-Around

None.

T5005

Message

```
T5005: Database name in rdbmsDbTable was in wrong
format in update_DbLimitedResource
```

Description

A database name was improperly formatted.

Work-Around

None.

T5006

Message

```
T5006: ct_send error in update_DbLimitedResource
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5007

Message

```
T5007: ct_send error in function dbPinger
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5008

Message

```
T5008: dbIndex_value, applIndex_value was not
created in dbmsDbInfoTable; in dbPinger
```

Description

A row in the *rdbmsDbInfoTable* was not created either because of insufficient memory or because it is a duplicate.

Work-Around

None.

T5009

Message

```
T5009: DbIndex_value, applIndex_value was not
created in rdbmsRelTable; in dbPinger
```

Description

A row in the *rdbmsRelTable* was not created either because of insufficient memory or because it is a duplicate.

Work-Around

None.

T5010

Message

`T5010: ct_command error in routine do_sp_who`

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5011

Message

`T5011: ct_send error in routine do_sp_who`

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5012

Message

`T5012: ct_results failed`

Description

This is an Open Client Client-Library error that occurred while retrieving data from server. Could be a server going down, or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5013

Message

```
T5013: ct_send failed in ApplEntry::connect
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5014

Message

```
T5014: ct_command failed in ApplEntry::connect
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5016

Message

```
T5016:  ct_command failed in ApplEntry::used
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5017

Message

```
T5017: ct_send failed in  ApplEntry::usedb
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5018

Message

```
T5018: ApplEntry::usedb: Cannnot change to
database_name database
```

Description

The server could be transitioning after a database was dropped. *database_name* could be restricted, not fully recovered, or suspect.

Work-Around

None.

T5019

Message

```
T5019: ct_command failed in
ApplEntry::often_refresh
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5020

Message

```
T5020: ct_send failed in  ApplEntry::often_refresh
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5021

Message

```
T5021: ct_command failed in rdbmsSrvEntry::seldom
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5022

Message

```
T5022: ct_send failed in ApplEntry::often_refresh
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5023

Message

```
T5023: ct_send failed in ApplEntry::often_refresh
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5024

Message

```
T5024: ct_command failed in rdbmsSrvEntry::seldom
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5025

Message

```
T5025: ct_send failed in ApplEntry::often_refresh
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5026

Message

```
T5026: rdbmsDbEntry.seldom: Can't find matching
applEntry
```

Description

The *rdbmsDbTable* is temporarily out of synchronization.

Work-Around

None.

T5027

Message

```
T5027: ct_send failed in
rdbmsDbEntry::seldom_refresh
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5028

Message

```
T5028: rdbmsDbInfoEntry.seldom: Can't find
matching applEntry
```

Description

An inconsistency exists between *rdbmsDbInfoTable* and the *ApplTable*.

Work-Around

None.

T5029

Message

```
T5029: rdbmDbInfoEntry.seldom: Can't find DbEntry
```

Description

An inconsistency exists between *rdbmsDbInfoTable* and *rdbmsDbTable*

Work-Around

None.

T5030

Message

```
T5030: ct_command failed in rdbmsDbEntry::seldom
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5031

Message

```
T5031: ct_send failed in  rdbmsDbEntry::seldom
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5032

Message

```
T5032: ct_command failed in ApplEntry::connect
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

T5033

Message

```
T5033: ct_send failed in ApplEntry::connect
```

Description

This is an Open Client Client-Library error that could be due to a temporary error or an unstable connection.

Work-Around

See your Open Client Client-Library documentation for more information.

B SNMP Version 2 Configuration

EMANATE is a master agent that supports SNMP version 1 (V1) and version 2 (V2). As part of the master agent installation, Sybase provides a default set of V2 configuration files. These files allow network managers to:

- Contact EMANATE from any host
- Issue all V1 PDU commands and the **get-bulk** V2 PDU command
- Use the "public" community string for "read" access
- Use the "private" community string for "read and write" access

If these defaults are not acceptable, you can modify the existing V2 configuration files for your environment or create new configuration files. However, before you modify the default V2 configuration files, you should be completely familiar with RFC's 1441 through 1452.

To help you create new configuration files, Sybase provides the **snmpv2** utility. You can use **snmpv2** to:

- Modify the default community strings Sybase provides
- Generate a set of basic configuration files based on the hosts and network managers in your environment

This appendix describes:

- The location and purpose of the default V2 configuration files
- How to use **snmpv2** to create new V2 configuration files
- What you should do after using **snmpv2**
- How and when to contact Sybase Technical Support for V2-related issues

This appendix does not describe the contents of the configuration files and it does not describe how to modify the parameters in the configuration files.

➤ ***Note***

If you encounter problems with the default V2 configuration files or if you need assistance producing V2 security files using **snmpv2**, you can contact Sybase Technical Support. However, if you modify the parameters in these files, Sybase Technical Support cannot help you debug your configuration. See "When to Contact Sybase Technical Support" on page C-1 for more information.

Configuration Files

When you set up the EMANATE environment, you define the environment variable SR_AGT_CONF_DIR. This environment variable identifies the configuration directory that contains master agent-specific configuration files. Typically, this directory is */etc/srconf/agt*.

In the SR_AGT_CONF_DIR directory there are the following V2 configuration files.

- *agt.pty*—contains entries for each master agent as a party in a communication with a network manager.
- *acl.pty*—controls the access of network managers to object values. This file contains an entry for each master agent and network manager that communicate.
- *context.pty*—specifies a party's **MIB view** or **proxy relationship**. The MIB view identifies the part of the MIB that a network manager can access. You configure the MIB view in *view.pty.* A proxy relationship describes the interaction of a master agent and monitored server when the master agent and the monitored server are on different hosts.
- *view.pty*—specifies the portion of the MIB that is accessible to a network manager.

If you want, you can modify these files to take advantage of additional V2 security features or to tailor them to your environment. However, if you modify the parameters in these files, Sybase Technical Support cannot help you debug your configuration. See "When to Contact Sybase Technical Support" on page C-1 for more information.

For information on the entries in each configuration file, edit the file and read the comments inside the file. Each entry in the

configuration files has a description of its parameter as it relates to RFCs 1441 through 1452.

➤ ***Note***

Each time you start EMANATE, it reads *snmpd.cnf*. If there are new community strings in *snmpd.cnf*, EMANATE adds the community string information to the configuration file *agt.pty*. However, if you remove community strings from *snmpd.cnf*, EMANATE does not remove them from *agt.pty* or *acl.pty*. Therefore, EMANATE still treats these community strings as valid. To remove these community strings from *acl.pty* and *agt.pty*, you must regenerate the basic V2 configuration files. See "Creating Basic V2 Configuration Files" for more information.

Creating Basic V2 Configuration Files

If the default configuration files that Sybase provides are not acceptable to you, you can modify the existing files or create a new set of configuration files. **snmpv2** enables you to generate configuration files for a specific list of master agents and network managers on multiple hosts. The following sections describe how to create input files and how to run **snmpv2.**

Creating Input Files

To use **snmpv2,** you must create three input files:

- *agents*—contains a list of hosts on which master agents reside
- *mgrs.v1*—contains a list of hosts on which V1 network managers reside, host community strings, and community string privileges
- *mgrs.v2*—contains a list of hosts on which V2 network managers reside

The following sections describe how to create these files.

Creating *agents*

agents contains a set of master agent entries. Each entry must begin on a separate line, and you should create an entry for every master agent you want to configure.

The syntax for each entry is:

ip_address host_name

where *ip_address* is the IP address of the machine on which EMANATE resides, and *host_name* is the alias string for the IP address. For example, the entries:

```
192.147.142.16   SQUONK
192.147.142.17   GABRIEL
```

identify that EMANATE resides on host SQUONK (IP address 192.147.142.16) and on host GABRIEL (IP address 192.147.142.17).

➤ ***Note***

If you install EMANATE on multiple hosts, you should have an entry in *agents* for each EMANATE instance you have. After **snmpv2** produces the configuration files, you must move the configuration files to the hosts on which EMANATE resides. See “Moving Configuration Files” on page B-6 form more information.

◆ ***WARNING!***

***agents* cannot have a blank line at the end of the file. If the last line in the file is blank, snmpv2 encounters errors.**

Creating *mgrs.v1*

mgrs.v1 contains entries for network managers that support V1. Each entry must begin on a separate line, and you should create an entry for every V1 network manager that contacts EMANATE. When you create entries, be aware that all V1 traps are sent to all managers in *mgrs.v1*.

The syntax for each entry is:

ip_address host_name read_community write_community

where *ip_address* is the IP address of the machine from which the network manager contacts EMANATE, *host_name* is the alias string of the IP address, *read_community* is the community string that provides read access to the manager, and *write_community* is the

community string that provides read and write access to the manager. For example, the entry:

```
192.147.142.33 SQUONK public private
```

provides the network manager on host SQUONK with read access when it uses the "public" community string and read and write access when it uses the "private" community string.

◆ ***WARNING!***

***mgrs.v1* cannot have a blank line at the end of the file. If the last line in the file is blank, snmpv2 encounters errors.**

Creating *mgrs.v2*

mgrs.v2 contains entries for network managers that support V2. Each entry must begin on a separate line, and you should create an entry for every V2 network manager that contacts EMANATE. When you create entries, be aware that all V2 traps are sent only to the first manager in *mgrs.v2*.

The syntax for each entry is:

ip_address host_name

where *ip_address* is the IP address of the machine from which the network manager contacts EMANATE, and *host_name* is the alias string of the IP address. For example, the entry:

```
192.147.142.33 SQUONK
```

identifies the network manager on host SQUONK to EMANATE.

◆ ***WARNING!***

***mgrs.v2* cannot have a blank line at the end of the file. If the last line in the file is blank, snmpv2 encounters errors.**

Using snmpv2

snmpv2 takes a set of input files and produces a set of output files. The input files identify the host from which network managers contact EMANATE, the community strings that provide read and write access, and the host locations of EMANATE. See "Creating Input Files" on page B-3 for more information. The output files identify

how each network manager and master agent interact. snmpv2 creates the output files in *$SYBSNMP/install*/configs. **If this directory does not exist, you must create before you run snmpv2.**

To run **snmpv2**, you must be in the *$SYBSNMP/install* directory. After you change directory to *$SYBSNMP/install*, enter the command:

```
snmpv2
```

After **snmpv2** completes, you should have a set of directories and configuration files in the *$SYBSNMP/install/configs* directory. See "Moving Configuration Files" on page B-6 for more information.

Moving Configuration Files

When you run **snmpv2** to create configuration files, **snmpv2**:

1. Reads agents to identify the hosts on which EMANATE resides.
2. Creates directories in *$SYBSNMP/install/configs* for every host in *agents*. For example, if *agents* has a host named SQUONK, **snmpv2** creates the *$SYBSNMP/install/configs/squonk* directory.
3. Creates the directory *agt* directory in each host directory. For example, **snmpv2** creates the *$SYBSNMP/install/configs/squonk/agt* directory.
4. Creates the configuration files in the *agt* directory.

Figure B-1 depicts the directory structure that **snmpv2** creates based on the preceding example:

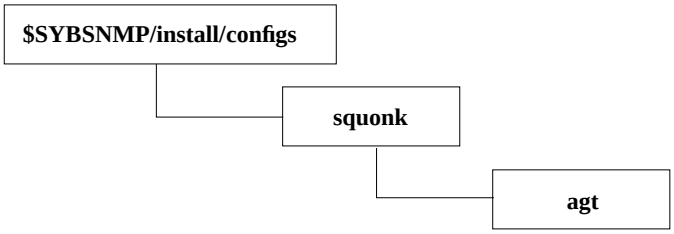

Figure B-1: Directory structure that snmpv2 creates

After you create the configuration files, you must move them to the *$SR_AGT_CONF_DIR* directory on each host in the *agents* file. See "Configuration Files" on page B-2 for more information on the configuration files.

C Troubleshooting

This appendix provides troubleshooting information for the master agent EMANATE and the Subagent. If the information in this appendix does not help you solve your problem, see "When to Contact Sybase Technical Support" before calling Sybase Technical Support.

When to Contact Sybase Technical Support

Before you contact Sybase Technical support, you should try to resolve your issues with the information in this appendix. If you are encountering problems with master agent configuration files or network managers, see the following two sections for additional technical support requirements.

SNMP Version 2 Configuration Files

As part of the master agent installation, Sybase provides a default set of SNMP version 2 (V2) configuration files. If the default settings in these files are not acceptable, you can modify the existing V2 configuration files for your environment or create new configuration files. However, before you modify the default V2 configuration files, you should be completely familiar with RFCs 1441 through 1452.

If you encounter problems with the default V2 configuration files or if you need assistance producing V2 security files using **snmpv2**, you can contact Sybase Technical Support. However, if you modify the default parameters in the configuration files, Sybase Technical Support cannot help you debug your configuration.

To determine if you have the default configuration files, edit *agt.pty* in the *$SR_AGT_CONF_DIR* directory. In *agt.pty*, look for the entry that begins `snmpUDPDomain`. If the two lines following each `snmpUDPDomain` entry consist of a dash (-), you have the default configuration files. For example, the following is a default configuration file:

```
snmpUDPDomain 127.0.0.1 161 300 1458
1 nonVolatile true 0
-
-
```

However, if the second of the two lines following any of the `snmpUDPDomain` entries contain a set of hexadecimal numbers, you do not have the default configuration files. For example, the following is not a default configuration file:

```
snmpUDPDomain 127.0.0.1 161 300 1458
3 nonVolatile true 0
-
2a 47 66 37 5a 2f
```

Before you call Sybase Technical Support, you should reset your environment to the default configuration files using **snmpdv2**. See Appendix B, "SNMP Version 2 Configuration," for more information on creating a new set of default configuration files for your environment.

Developing Network Managers

If you are developing a network manager and are having problems communicating with the master agent EMANATE, you should not contact Sybase Technical Support. Instead, you should contact SNMP Research, Inc. (support@snmp.com) or Sybase Professional Services. Both organizations have consultants that are willing to assist you.

Using Subagent Diagnostic Capabilities

The Subagent has two built-in diagnostic features: you can run the Subagent in diagnostic mode, or you can use the Subagent log file to capture messages. In addition, Sybase provides the **getmany** command that allows you to query the Subagent for current values. The following sections describe the Subagent's diagnostic capabilities and **getmany.**

Running the Subagent in Diagnostic Mode

You can start the subagent in two different modes: normal or diagnostic. When you start the Subagent in diagnostic mode, it:

1. Reads the initialization file *snmp.ini*.
2. Connects to all of the servers listed in *snmp.ini*.
3. Collects information from each server.
4. Rewrites *snmp.ini* with additional database information.

5. Converts all passwords from human-readable form to a secure, unreadable format.

Before you run the Subagent in diagnostic mode, make sure all of the SQL Servers and Open Servers that you listed in *snmp.ini* are running.

To run the Subagent in diagnostic mode, enter the command:

```
sybagt -D
```

After the Subagent connects to each server and updates *snmp.ini*, it shuts down. During the diagnostic process, the Subagent writes all status and error messages to *sybagt.log* in the */tmp* directory. See Appendix A, "Messages," for details on messages and correcting problems.

➤ ***Note***

By default the -D flag directs the Subagent to send all messages to *sybagt.log* in the */tmp* directory. However, you can send messages to a different file by using the **-e** flag. See "Starting the Subagent" on page 3-11 for more information.

Using the Log File

When you start the Subagent, you can direct it to send all informational, warning, and error messages to a log file. If you are encountering problems, the log file is a good tool to use for capturing information.

To direct the Subagent to use the log file, use the -e flag in the command line. For example, the command:

```
sybagt –e $SYBSNMP/subagent.log
```

directs the Subagent to send all log file messages to the file *subagent.log* in the *$SYBSNMP* directory. See "Starting the Subagent" on page 3-11 for more information on the **-e** flag. See Appendix A, "Messages," for details about interpreting error messages and correcting problems.

➤ ***Note***

After you have diagnosed and resolved the problem, it is good practice to restart the Subagent without the **-e** flag.

Using getmany

As part of the Subagent distribution, Sybase provides **getmany. getmany** allows you to extract values from objects that the Subagent is maintaining. You typically use **getmany** to determine if the values the Subagent has are the same values that the network manager is displaying.

Before you use **getmany,** you must set the environment variable SR_MGR_CONF_DIR to the master agent utility directory. Typically, this directory is */etc/srconf/mgr*, but it may be different for your environment. See "Setting Environment Variables" on page 2-8 for more information.

getmany is in the *$SYBSNMP/bin* directory and its syntax is:

```
getmany host com_string object
```

where *host* is the name of the host on which the agents are running, *com_string* is a community string, and *object* is the MIB object whose value you want to check. For example, the command:

```
getmany squonk public sybase
```

contacts the master agent on "squonk" using the "public" community string for information on the "sybase" object. Using this command, **getmany** produces something similar to the following:

```
sybShortSleep.0 = 20
sybLongSleep.0 = 2
sybApplModIndex.1 = 1
sybApplIndex.1 = 0
sybApplModCMD.1 = addServer(1)
sybApplModNewPassword.1 = **********
sybApplModApplName.1 = Server Name
```

➤ ***Note***

If you use "system" to look at system variables, be aware that if there are any spaces after *sysDescr*, *sysLocation*, or *sysContact*, **getmany** displays numbers instead of the appropriate information. See "Modifying System Information" for more information.

Situations and Solutions

The following sections describe possible situations you might encounter and the solutions to these problems.

Delayed or No Response from the Subagent

When you monitor multiple servers, it is important to have an interval between ShortSleep and LongSleep intervals that allows the Subagent to adequately poll all the servers. If the network manager is getting no response or delayed response to queries for status updates, try increasing the ShortSleep and LongSleep polling intervals.

You can modify ShortSleep and LongSleep by editing *snmp.ini* or by setting new intervals from the network manager. See "Modifying snmp.ini" on page 3-6 and "Modifying Object Values" on page 5-11 for more information.

➤ ***Note***

If you are using the Subagent to monitor a server on a different host, response time is slower.

Incorrect *applUptime*

Sybase recommends that you have a Subagent on each host that has a server you want to monitor. However, the Subagent does not restrict you from monitoring servers on a different host. If you use the Subagent to monitor a server on a different host, be aware that values for *applUptime* will not always be correct.

Using Multiple Subagents in the same *$SYBASE* Directory

If you have more than one Subagent sharing the same *$SYBASE* directory, the name of the *snmp.ini* file must be different for each Subagent to avoid duplicate monitoring. When you configure the *snmpi.ini* file during the installation process, rename the *snmp.ini* file and specify an alternate filename on the command line:

```
snmpconfig -c filename
```

Master Agent Does Not Start

When you start EMANATE, it reads its initialization file *snmpd.cnf*. This file is in the directory defined by the environment variable SR_AGT_CONF_DIR.

If you cannot start the master agent, make sure that the *sysObjID* entry in *snmpd.cnf* has no extra spaces at the end of the entry. If there are, the master agent cannot start. In addition, the master agent puts the following message in its log file:

```
ParseSubIdDecimal, bad digit:
                    at line num in file oid_lib.c
```

where *num* is the line number in *snmpd.cnf*.

Glossary

authentication trap

A special type of security trap. When EMANATE receives a PDU, it verifies that the community string has privileges to set or get object information. If the community string is invalid, EMANATE sends a trap to each of the hosts defined in the trap definition section of *snmpd.cnf*.

community string

An SNMP password that allows a network manager to access a Subagent. The community string is part of the PDU the network manager sends. See **PDU**.

context

The context specifies the MIB view or proxy relationship of an agent. You specify the context in the configuration file, *context.pty*.

LongSleep

A time interval the Subagent waits between polling the server to update data. LongSleep specifies the number of times the Subagent cycles through its ShortSleep polls. The default value is 4 ShortSleep cycles. The range is from 1 to 20 ShortSleep cycles.

master agent

A request broker that listens to port 161 and routes the requests to the appropriate subagent. The master agent returns responses from subagents to the network manager using port 162. As part of the Subagent distribution, Sybase provides the master agent EMANATE. See Chapter 2, “Using EMANATE” for more information on EMANATE.

MIB

A Management Information Base describes common metrics of relational database systems. The MIB describes the data elements that are supported by a Subagent. The MIB states the maximum accessibility (such as Read-Only, Read-Write, or Not accessible), data type (such as *Integer* or *DisplayString*), description, and unique object identifier for each data element. There are three MIBs included with SQL Server SNMP Subagent: RDBMS, Network Services, and Sybase Common. Nearly all the data in the MIB is in table format. See also **Network Services MIB**, **Relational Database Management System MIB**, **SYBASE MIB**.

MIB View

The portion of a MIB that is accessible to an agent. You define the view in the snmpV2 security configuration file, *view.pty*.

network manager

Requests information from objects in the network. Network managers can also modify the values of some objects. The network manager sends requests to User Datagram Protocol (UDP) port 161 and listens to UDP port 162 for responses to requests and for event traps generated by the Subagent. To request information and to modify values, network managers use protocol data units (PDU).

Network Services MIB

An Internet standard for supplying information on network service applications. The Subagent supports the application table (*applTable*) of this MIB. The application table contains an entry for each SQL Server or Open Server that the Subagent is monitoring. A network manager reads a compiled version of the Network Services MIB so that it knows which servers are available. See Chapter 4, "Objects in the Network Services MIB and RDBMS MIB" for more information on the Network Services MIB.

objects

An entity described in a MIB that a network manager can use.

object identifier

See **OID**.

OID

The unique object identifier (OID) for each data object supported by SQL Server SNMP Subagent. Data objects are described in the MIBs.

PDU

A protocol data unit enables a network manager to get a value from a managed object, set or modify a value, and be alerted through a trap. Version 1 of SNMP supports five PDUs: **get-request**, **get-next-request**, **set-request**, **get-response**, and **trap**. Version 2 of SNMP supports an additional two PDUs: **get-bulk-request** and **get-bulk-response**. With the exception of traps, all messages originate with the network manager. The Subagent generates traps.

Ports 161 and 162

The industry standard ports for sending and receiving PDUs and traps. The network manager sends PDUs requesting information to port 161 and receives responses and traps from port 162.

protocol data unit

See **PDU**.

Relational Database Management System MIB

An Internet standard for supplying information about installed servers, databases, and the relation of servers and databases. A network manager reads a compiled version of the RDMBS MIB so that it knows what types of information it can access about a server or database. See Chapter 4, "Objects in the Network Services MIB and RDBMS MIB" for more information on the RDBMS MIB.

ShortSleep

A time interval (measured in seconds) that SQL Server SNMP Subagent waits between polling the server to update data. These polls are called frequent polls. The range is from 10 to 1800 seconds.

snmp.ini

The Subagent initialization file created during installation when you enter password and polling interval information in response to prompts. The file contains an entry for each server to be monitored and information SQL Server SNMP Subagent must retain while it is down. The information includes the *ShortSleep* and *LongSleep* polling intervals and the scrambled passwords the Subagent must use to log into monitored servers. You can edit the file to change passwords and polling intervals.

snmpd.cnf

The master agent configuration file created by the installation process. The file contains entries that define the host on which the master agent is running, hosts on which you can do **set** operations, hosts to which you send traps, and so on. You also disable or enable authentication-failure traps in this file. You can edit the file.

SQL Server SNMP Subagent

Monitors characteristics of the SQL Servers and Open Servers and stores this information in an internal cache. When the Subagent receives a request from a master agent, it accesses the internal cache and returns the requested information to the master agent. In addition, the Subagent also detects events and reports these events as SNMP traps on port 162.

subagent

An intermediate process between a network manager and a monitored object. Subagents obtain data from the monitored objects and presents the information to the network manager using the SNMP protocol. See also **master agent**, **SQL Server SNMP Subagent**.

subtree

The lowest level of information in a MIB that is accessible to an agent. You define the subtree view in the snmpV2 security configuration file, *view.pty*.

SYBASE MIB

A Sybase-specific MIB that identifies polling intervals and login information used by the Subagent to monitor servers. A network manager reads a compiled version of this MIB so that it can read and modify polling intervals and login information. See Chapter 5, “Objects in the SYBASE MIB” for more information on the SYBASE MIB.

trap

A notification that a specific event has occurred. See also **Authentication trap**.

Index

D

E

F

G

H

I

K

L

M

P

R

S