INFORMIX®-Universal Server

SNMP Subagent Guide

Version 9.1
March 1997
Part No. 000-4806

Published by INFORMIX® Press

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025

Documentation Team: Diana Chase, Barbara Daniell, Judith Sherwood

Table of Contents

Introduction

Read this introduction for an overview of the information provided in this manual and for an understanding of the documentation conventions used.

About This Manual

The *INFORMIX-Universal Server SNMP Subagent Guide* describes the **onsnmp** subagent that provides information about Informix database servers to network-management tools. This manual provides a short introduction to the Simple Network Management Protocol (SNMP) and documents the use of **onsnmp**. This manual also documents the Management Information Bases (MIBs) that specify the information that **onsnmp** provides to the network-management tools.

Organization of This Manual

This manual includes the following chapters:

- This Introduction provides an overview of the manual and describes the documentation conventions used.
- Chapter 1, "The SNMP Protocol," describes how Informix uses SNMP to provide access to general-purpose network-management tools.
- Chapter 2, "The Informix SNMP Subagent," describes the **onsnmp** utility that gives the network manager access to information about Informix databases.
- Chapter 3, "The Management Information Bases,"describes how the information that is available to network-management tools is specified.

- Chapter 4, "Using SNMP," gives examples of features of network managers and suggests items that are useful to examine.
- Appendix A describes the structure and contents of the application MIB, the RDBMS MIB, and the Informix MIB.

Types of Users

This manual is written for database administrators who use SNMP network-management tools and for programmers who plan to write customized SNMP network-management tools.

If you have limited experience with relational databases, SQL, or your operating system, refer to *Getting Started with INFORMIX-Universal Server* for a list of introductory texts.

Software Dependencies

This manual assumes that you are using INFORMIX-Universal Server, Version 9.1, as your database server.

In this manual, all instances of Universal Server refer to INFORMIX-Universal Server.

In addition to the database server, you must have installed an SNMP-compliant network server and an SNMP-compliant master agent.

Assumptions About Your Locale

Informix products can support many languages, cultures, and code sets. All culture-specific information is brought together in a single environment, called a GLS (Global Language Support) locale.

This manual assumes that you are using the default locale, **en_us.8859-1**. This locale supports U.S. English format conventions for dates, times, and currency. In addition, this locale supports the ISO 8859-1 code set, which includes the ASCII code set plus many 8-bit characters such as é, è, and ñ.

If you plan to use nondefault characters in your data or your SQL identifiers, or if you want to conform to the nondefault collation rules of character data, you need to specify the appropriate nondefault locale(s). For instructions on how to specify a nondefault locale, additional syntax, and other considerations related to GLS locales, see the *Guide to GLS Functionality*.

Major Features

The Introduction to each Version 9.1 product manual contains a list of major features for that product. The Introduction to each manual in the Version 9.1 *Informix Guide to SQL* series contains a list of new SQL features.

Major features for Version 9.1 Informix products also appear in release notes.

The Informix SNMP subagent allows you to use SNMP-compliant tools to monitor Informix database servers of Version 7.22 or later.

Documentation Conventions

This section describes the conventions that this manual uses. These conventions make it easier to gather information from this and other Informix manuals.

The following conventions are covered:

- Typographical conventions
- Icon conventions

Typographical Conventions

This manual uses the following standard set of conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth.

Convention	Meaning
KEYWORD	All keywords appear in uppercase letters in a serif font.
italics	Within text, new terms and emphasized words appear in italics. Within syntax diagrams, values that you are to specify appear in italics.
boldface	Identifiers (names of classes, objects, constants, events, functions, program variables, forms, labels, and reports), environment variables, database names, filenames, table names, column names, icons, menu items, command names, and other similar terms appear in boldface.
`monospace`	Information that the product displays and information that you enter appear in a monospace typeface.
KEYSTROKE	Keys that you are to press appear in uppercase letters in a sans serif font.
♦	This symbol indicates the end of feature-, product-, platform-, or compliance-specific information.

Tip: *When you are instructed to "enter" characters or to "execute" a command, immediately press* RETURN *after the entry. When you are instructed to "type" the text or to "press" other keys, no* RETURN *is required.*

Icon Conventions

Throughout the documentation, you will find text that is identified by several different types of icons. This section describes these icons.

Comment Icons

Comment icons identify warnings, important notes, or tips. This information is always displayed in italics.

Icon	Description
	The *warning* icon identifies vital instructions, cautions, or critical information.
	The *important* icon identifies significant information about the feature or operation that is being described.
	The *tip* icon identifies additional details or shortcuts for the functionality that is being described.

Additional Documentation

For additional information, you might want to refer to the following types of documentation:

- On-line manuals
- Printed manuals
- Documentation notes, release notes, and machine notes
- Related reading

On-Line Manuals

A CD that contains Informix manuals in electronic format is provided with your Informix products. You can install the documentation or access it directly from the CD. For information about how to install, read, and print on-line manuals, see either the installation guide for your product or the installation insert that accompanies the documentation CD.

The documentation set that is provided on the CD describes Universal Server, its implementation of SQL, and its associated application-programming interfaces. For an overview of the manuals in the Universal Server documentation set, see *Getting Started with INFORMIX-Universal Server*.

Printed Manuals

The Universal Server documentation set describes Universal Server, its implementation of SQL, and its associated application-programming interfaces. For an overview of the manuals in the Universal Server documentation set, see *Getting Started with INFORMIX-Universal Server*.

To order printed manuals, call 1-800-331-1763 or send email to **moreinfo@informix.com**.

Please provide the following information:

- The documentation that you need
- The quantity that you need
- Your name, address, and telephone number

Documentation Notes, Release Notes, Machine Notes

In addition to printed documentation, the following on-line files, located in the **$INFORMIXDIR/release/en_us/0333** directory, supplement the information in this manual.

On-Line File	Purpose
ONSNMPDOC_9.1	The documentation-notes file describes features that are not covered in this manual or that have been modified since publication.
SERVERS_9.1	The release-notes file describes feature differences from earlier versions of Informix products and how these differences might affect current products. This file also contains information about any known problems and their workarounds.
IUNIVERSAL_9.1	The machine-notes file describes any special actions that are required to configure and use Informix products on your computer. Machine notes are named for the product described.

Please examine these files because they contain vital information about application and performance issues.

In addition to the information in the machine notes, the actual MIB files are included in the **$INFORMIXDIR/snmp** directory. You can read these files for very detailed information about the MIBs. For a description of the MIB files, refer to Chapter 3, "The Management Information Bases."

Related Reading

For more information about network managers and master agents, refer to the documentation provided by the vendors of those products.

SNMP Information

For additional general information about SNMP, consult the following books:

- *The Simple Book: An Introduction to Internet Management, 2nd Edition*, by Marshall T. Rose (Prentice Hall, 1994)
- *SNMP, SNMPv2 and RMON: Practical Network Management, 2nd Edition*, by William Stallings (Addison-Wesley, 1996)

ASN1 Information

For information about the Abstract Syntax Notation One, refer to the following publications:

- Information Processing - Open Systems Interconnection, *Specification of Abstract Syntax Notation One (ASN.1)*, International Organization for Standardization and International Electrotechnical Committee, 1987, International Standard 8824.
- Information Processing - Open Systems Interconnection, *Abstract Syntax Notation One (ASN.1) - Addendum 1: Extensions to ASN.1*, International Organization for Standardization and International Electrotechnical Committee, 1987, International Standard 8824 / AD 1.

SNMP Protocols

The Request for Comments (RFC) documents are the means for distributing information and proposals about the Internet suite of protocols.

The following RFCs describe SNMPv1:

- Rose M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets," STD 16, RFC 1155, May 1990.
- Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions," STD 16, RFC 1212, March 1991.
- Case, J., M. Fedor, M. Schoffstall, and J. Davin, "The Simple Network Management Protocol," STD 15, RFC 1157, May 1990.
- McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based Internets - MIB-I," STD 17, RFC 1213, March 1991.

The following RFCs describe SNMPv2:

- SNMP Working Group, J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Introduction to Community-based SNMPv2," RFC 1901, January 1996.
- SNMP Working Group, J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 1902, January 1996.
- SNMP Working Group, J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 1903, January 1996.

The following publications describe the RDBMS public MIB:

- SNMP Working Group, J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 1907, January 1996.
- Kille, S., Working Group Chair, and N. Freed, Editor, "The Network Services Monitoring MIB," RFC 1565, January 1994.
- Brower, D., Editor, R. Purvy, Working Group Chair, A. Daniel, M. Sinykin, and J. Smith, "Relational Database Management System (RDBMS) Management Information Base (MIB) using SMIv2," RFC 1697, August 1994.

Compliance with Industry Standards

The Informix **onsnmp** subagent complies with the SNMP and SNMPv2 standards presented by the Internet Engineering Task Force (IETF).

Informix Welcomes Your Comments

Please tell us what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about corrections or clarifications that you would find useful. Include the following information:

- The name and version of the manual that you are using
- Any comments that you have about the manual
- Your name, address, and phone number

Write to us at the following address:

Informix Software, Inc.
SCT Technical Publications Department
4100 Bohannon Drive
Menlo Park, CA 94025

If you prefer to send email, our address is:

doc@informix.com

Or send a facsimile to the Informix Technical Publications Department at:

415-926-6571

We appreciate your feedback.

Chapter

1

The SNMP Protocol

The Simple Network Management Protocol (SNMP) refers to a set of standards for network management. The protocol allows vendors of hardware (such as routers) and software (such as database managers) to present information and control data to general-purpose network management tools. This chapter gives a brief description of SNMP and SNMP applications. The next chapter provides specific information about how SNMP applications can monitor Informix database servers.

Tip: *Because Informix products are database managers, this chapter discusses SNMP from the point of view of a database administrator. However, the discussion could apply to any SNMP-compliant application or hardware device.*

SNMP Network Management Systems

A SNMP *network-management system* (or *network manager*) is a collection of tools for network monitoring and control. An application provider, such as Informix, can provide information to an SNMP network manager. Using the network manager, a database (or network) administrator can remotely monitor the status of all database servers on a network. SNMP managers typically provide the following features:

- Remote monitoring of database servers, operating systems, routers, printers, and other networked devices
- A single operator interface with user-friendly commands
- Low-impact sampling of database server performance
- Correlation of database server metrics with related system and network metrics
- Graphical presentation of information, such as submaps, graphs, and iconic representation of state

Most network managers provide graphical user interfaces such as the one illustrated in Figure 1-1. The user selects a node to monitor and then chooses specific information from a menu.

Figure 1-1
Sample Monitoring Display

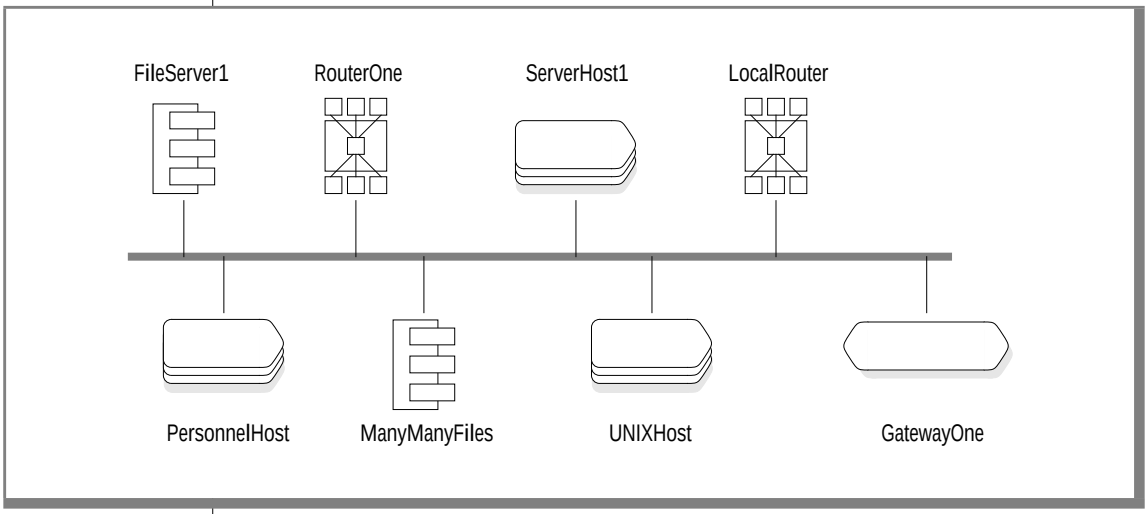


Figure 1-2 shows the results of a request for information about the chunks on the **smoke** database server. (This database server has only one chunk.)

Figure 1-2
Sample Monitoring Information

```
Tue Jul 16 1996 [ smoke ] : Online-MIB.onChunkTable
KEY = 8930072.1.1
IndexValue = 8930072.1.1
onChunkFileName=/ix/windstar/root_chunk
```

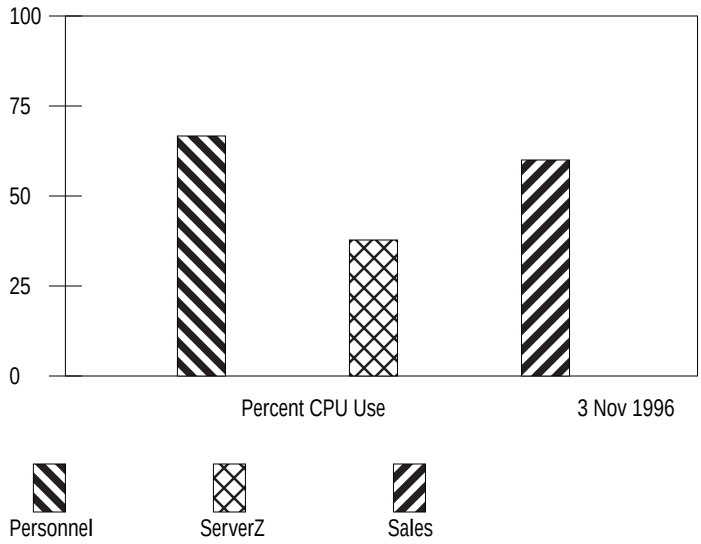
Figure 1-3 shows the same information displayed by a different network manager.

Figure 1-3
Sample Monitoring Information

```
onChunkFileName.8930072.1.1 =
/ix/windstar/root_chunk
onChunkFileOffset.8930072.1.1 = 0
onChunkPagesAllocated.8930072.1.1 = 10000
onChunkPagesUsed.8930072.1.1 = 3447
onChunkType.8930072.1.1 = regularChunk(1)
onChunkStatus.8930072.1.1 = online(2)
onChunkMirroring.8930072.1.1 = notMirrored(1)
onChunkReads.8930072.1.1 = 300
onChunkPageReads.8930072.1.1 = 343
onChunkWrites.8930072.1.1 = 6
onChunkPageWrites.8930072.1.1 = 7
```

In addition to textual information, a network manager can also generate graphs or charts, as Figure 1-4 illustrates.

Figure 1-4
Sample Monitoring Information: CPU Use by Different Servers



For more information about the information that you can monitor, refer to Chapter 3, "The Management Information Bases," and Appendix A, "MIB Tables."

A Brief Background

SNMP is a published, open standard for the management of heterogeneous applications over a TCP/IP network.

The SNMP protocol was originally developed for the remote administration of an Internet system. However, the design of the protocol has allowed it to grow to cover application management as well as systems management.

SNMP network managers use a *connectionless protocol*. That is, each exchange between the network manager and the agent is a separate transaction. Using a connectionless protocol allows the network manager to gather information without putting an excessive load on the network and to function in an environment where heavy traffic might cause network problems.

For more information about SNMP, refer to the manuals that accompany your SNMP application and to the books listed in "Related Reading" on page 9 of the Introduction.

SNMP Architecture

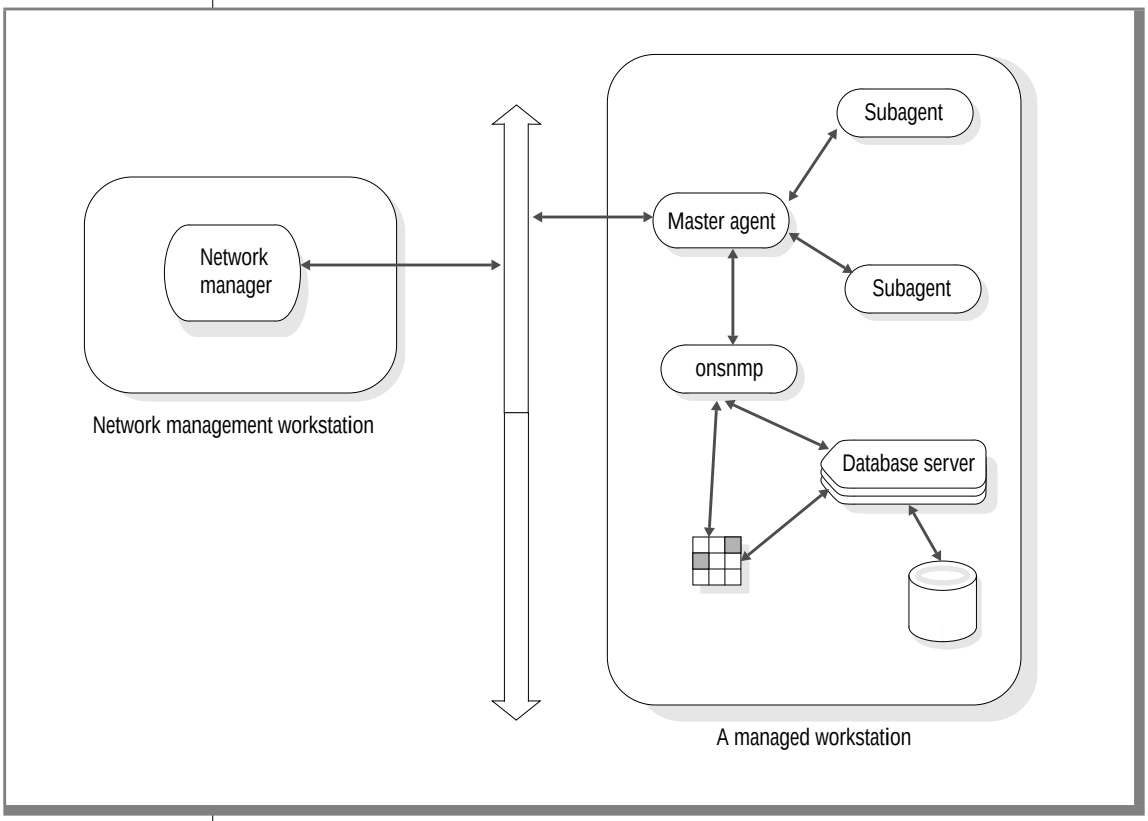
An SNMP network management system is organized in three layers, as follows:

- The network manager
- Master agents
- Subagents

Figure 1-5 on page 1-8 illustrates the organization of a network management system.

The highest layer of the network management system is the *network manager*, the management application that the database (or network) administrator uses. Below the manager are the *master agents* that provide information to the manager. Each computer that has a managed subsystem (for example, a database server) has a master agent. Each subsystem (application) has a *subagent* that provides information to the master agent. The subagent for Informix database servers is **onsnmp**. A managed workstation might have several subagents, but it typically has only one master agent.

Figure 1-5
Organization of an SNMP Management System



Network Managers

The network manager requests information from the master agents and displays that information to the system or database administrator. A network manager typically provides a sophisticated user interface that presents the information in easily read formats, such as graphs, charts, or dials. The database administrator can select the items to monitor and the form in which the information should be displayed.

Many vendors of hardware and network services have created SNMP-compliant network managers. Examples of SNMP-compliant network managers include the following:

- Sun Solstice
- Hewlett-Packard Open View
- IBM Netview / 6000
- Tivoli TME 10 NetView
- CA-Unicenter
- Novell Network Management System

For information about installing the network server, refer to your vendor documentation and to the machine notes described in "Documentation Notes, Release Notes, Machine Notes" on page 8 of the Introduction.

Master Agent

The master agent is responsible for parsing the requests from the network manager and formatting the responses from the subagent. It routes requests from the network manager to the appropriate subagent. The master agent collects statistics from the subagent(s) and returns the information to the network manager.

Each workstation can have only one master agent. Not all workstations are required to have the same master agent. Different workstations on the network can have different (SNMP-compliant) master agents.

Master agents are platform specific. Informix provides a master agent with Universal Server through licensing agreements with a master-agent vendor. For more information about the appropriate master agent for your platform, refer to the machine notes described in "Documentation Notes, Release Notes, Machine Notes" on page 8 of the Introduction.

For information about installing the master agent, refer to your vendor documentation. Installation information is also described in "Starting the Master Agent" on page 2-7 and the machine notes described in "Documentation Notes, Release Notes, Machine Notes" on page 8 of the Introduction.

Subagent

The subagent receives queries from the master agent, collects the requested information from shared memory or from the database server, and returns the information to the master agent. Each managed element (that is, each database server) has its own subagent. The subagent for Informix database servers, **onsnmp**, is an integral part of the Informix database server.

SNMP does not specify the protocol that should be used to communicate between the master agent and the subagent. The master agent and the subagent can communicate using whatever protocol is convenient. However, the master agent must use SNMP to communicate with the network manager.

Management Information Bases

All three entities (manager, master agent, subagent) refer to information contained in *Management Information Bases* (MIBs). The MIBs specify what information the subagent provides to the master agent. The master agent in turn passes the information to the network manager. For more information about MIBs, refer to Chapter 3, "The Management Information Bases."

Chapter

2

The Informix SNMP Subagent

This chapter describes the Informix SNMP subagent, **onsnmp**, and the SNMP master agent. The **onsnmp** subagent extracts information from the database server and provides that information to a master agent, which in turn passes the information to a network manager.

The onsnmp Subagent

The **onsnmp** subagent is a feature that you can use with Informix database servers, but **onsnmp** is not required for proper functioning of the database server.

If you plan to use an SNMP-compliant network manager, you must start a master agent and the **onsnmp** subagent on each workstation where an Informix database server is running.

Information That the onsnmp Subagent Provides

The information that the **onsnmp** subagent provides to the network manager is specified by three MIBs. The MIBs include the following tables and objects.

Application MIB	RDBMS MIB	Informix MIB
applTable	rdbmsDbTable	onServerTable
	rdbmsDbInfoTable	onDatabaseTable
	rdbmsDbParamTable	onTableTable
	rdbmsDbLimitedResourceTable	onActiveTableTable
	rdbmsSrvTable	onFragmentTable

(1 of 2)

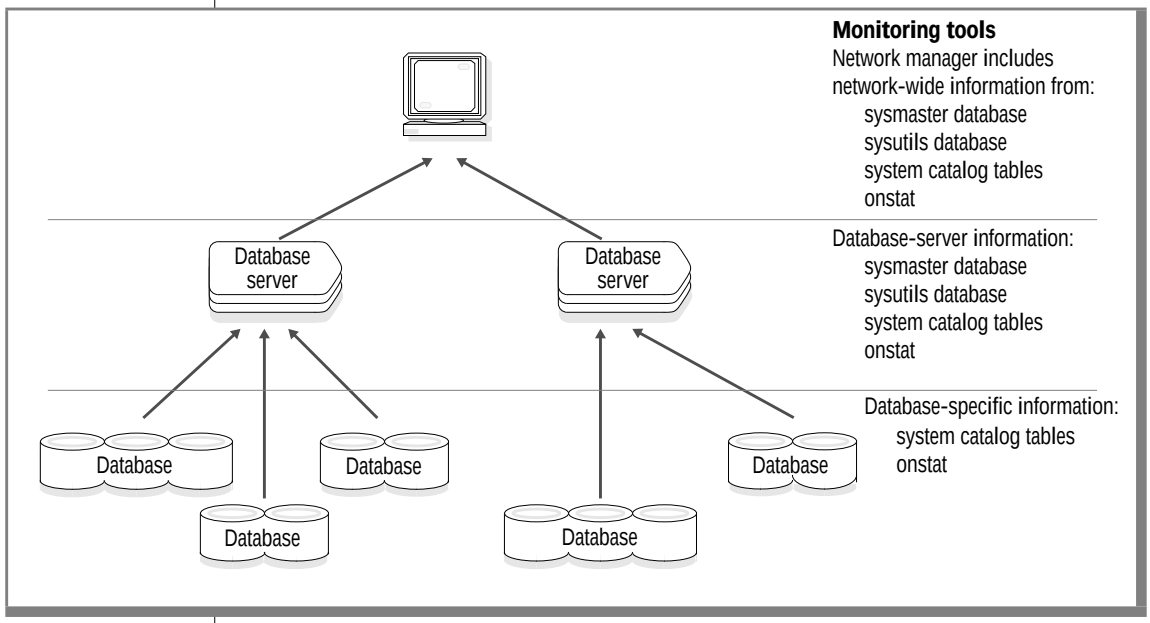
Application MIB	RDBMS MIB	Informix MIB
	rdbmsSrvInfoTable	onDbspaceTable
	rdbmsSrvParamTable	onChunkTable
	rdbmsSrvLimitedResourceTable	onLogicalLogTable
	rdbmsRelTable	onPhysicalLogTable
	rdbmsTrap	onSessionTable
		onLockTable
		onBarTable
		onSqlHostTable

(2 of 2)

For more information about the MIB tables, refer to Chapter 3, "The Management Information Bases."

The information provided by **onsnmp** is, for the most part, available from other sources, such as the system catalog tables, the **sysmaster** and **sysutils** databases, and the **onstat** utility. However, the system catalog tables and the **onstat** utility refer only to a single database, and the **sysmaster** and **sysutils** databases refer only to a single database server. The **onsnmp** subagent provides information that allows a network manager to monitor all of the Informix databases that are on a network. Figure 2-1 on page 2-5 illustrates this point.

Figure 2-1
Monitoring Informix Databases



Managing the onsnmp Subagent

The **onsrvapd** daemon watches for Informix database servers. Each time that an Informix database server starts, **onsrvapd** starts an **onsnmp** subagent for that database server. You never explicitly start **onsnmp**.

Starting the onsrvapd Daemon

When you install an Informix database server on your workstation, add **onsrvapd** to the startup procedure so that the **onsrvapd** daemon starts each time that the workstation is booted up.

When you add **onsrvapd** to your bootup files, you can use the following arguments to modify the behavior of **onsrvapd** and **onsnmp**.

Argument	Purpose
-d	Do not start **onsrvapd** as a daemon.
-l *pathname*	Store the log file in the *pathname* directory. The default path is **/tmp**.
-p *pollsecs*	Set the server discovery polling time to *pollsecs* seconds. The default polling time is 5 seconds. The **onsrvapd** daemon passes this value to the **onsnmp** subagent.
-k *lingermmnts*	Set the time that **onsrvapd** should wait for a server which has gone down to come back up to *lingermnts* minutes. If the linger time expires, the corresponding subagent is killed. If *lingermnts* is 0, wait indefinitely. The **onsrvapd** daemon passes this value to the **onsnmp** subagent.

Stopping the onsnmp Subagent

When a database server is halted, either with an **onmode -k** command or through an abnormal situation, the **onsrvapd** daemon notices that the database server is not active and removes the associated **onsnmp** subagent. You do not need to stop **onsnmp**.

Using More Than One Database Server

The **onsrvapd** daemon notices each time that an Informix database starts. If you start a second database server on a workstation, the **onsrvapd** daemon starts a second **onsnmp** subagent. The network manager can differentiate between the two database servers because the definitions in the MIB tables use the SERVERNUM as part of the identifier for each database server.

Files That onsnmp Uses

The **onsnmp** subagent, the **onsrvapd** daemon, and the master agent all create log files. The default location for the log files is the **/tmp** directory. To modify the location of **onsrvapd.log**, add an argument to the **onsrvapd** command in the bootup files.

The following list describes some of these files. (There might be additional files, depending on the master agent that you use.)

File name	Created by	Purpose
onsnmp.*servername*.log	**onsnmp**	The **onsnmp** error log
onsrvapd.log	**onsrvapd**	The **onsrvapd** error log

The Master Agent

The master agent is the intermediary between the **onsnmp** subagent and the network manager. The **onsnmp** subagent expects that a master agent is always present when the workstation is active.

You can install any SNMP-compliant master agent on your workstation. Many hardware vendors and software developers offer provide SNMP-compliant master agents. The following sections discuss general aspects of managing a master agent and give an example of installing a specific master agent.

For information more about installing a master agent, refer to the vendor documentation and to the information in the machine-notes file that is described in "Documentation Notes, Release Notes, Machine Notes" on page 8 of the Introduction.

Starting the Master Agent

When you install the master agent on your workstation, make sure that the master agent is included in the startup procedure so that the master agent starts each time that the workstation is booted up.

Environment Variables

Check the vendor documentation to see whether the master agent requires any environment variables. You might add any required environment variables to the **.login** files or to the setup files for your Informix database server.

Example of Master Agent Installation

Through a licensing agreement with SNMP Research, Incorporated, Informix includes Release 12.3 of the SNMP Research master agent with this version of Universal Server.

To install the SNMP Research master agent

1. Become **root**. You must have **root** privileges to install and start the master agent.

   ```
   su root
   ```

2. Verify that the snmp ports are defined in your **/etc/services** file.

   ```
   grep snmp /etc/services
   snmp        161/udp
   snmp-trap   162/udp
   ```

3. Set the master agent configuration environment variable as follows:

   ```
   setenv SR_AGT_CONF_DIR $INFORMIXDIR/snmp/snmpr
   ```

4. Kill any existing SNMP master agent or daemon.

5. Start the agent.

   ```
   $INFORMIXDIR/bin/snmpdm &
   ```

Chapter

3

The Management Information Bases

This chapter describes the Management Information Bases (MIBs) that you use with an SNMP-compliant network manager. For a detailed list of the MIB information that **onsnmp** provides, refer to Appendix A.

When you use a network manager, the user interface hides much of the structure that this chapter discusses. However, an understanding of the hierarchy of the MIBs might help you as you use the displays that the network manager generates. This chapter also provides an introduction to MIBs for the person who is developing a custom network manager.

Purpose of the Management Information Bases

A MIB describes information about a managed object (in this case, an Informix database server) that a network manager can access. The network manager uses this information to prepare reports, graphs, charts, and so on for the database or system administrator.

All MIBs use a common interface definition language. The Structure of Management Information (SMI) defines this language and dictates how each object in the MIB should be described using Abstract Syntax Notation One (ASN.1). For detailed information about SMI and ASN.1, refer to the documents listed in "Related Reading" on page 9 of the Introduction.

Tip: *A MIB is analogous to (but not the same as) an Informix database schema. Like a database schema, the MIB tells the network manager the characteristics of the data that it can access.*

The Informix SNMP subagent, **onsnmp**, consults three MIBs to provide information to the network manager, as follows:

- The relational database public MIB (RDBMS MIB)
- The application public MIB (APPL MIB)

  The **onsnmp** subagent consults only that portion of the application MIB that is required by the RDBMS MIB.
- The Informix private MIB (Informix MIB)

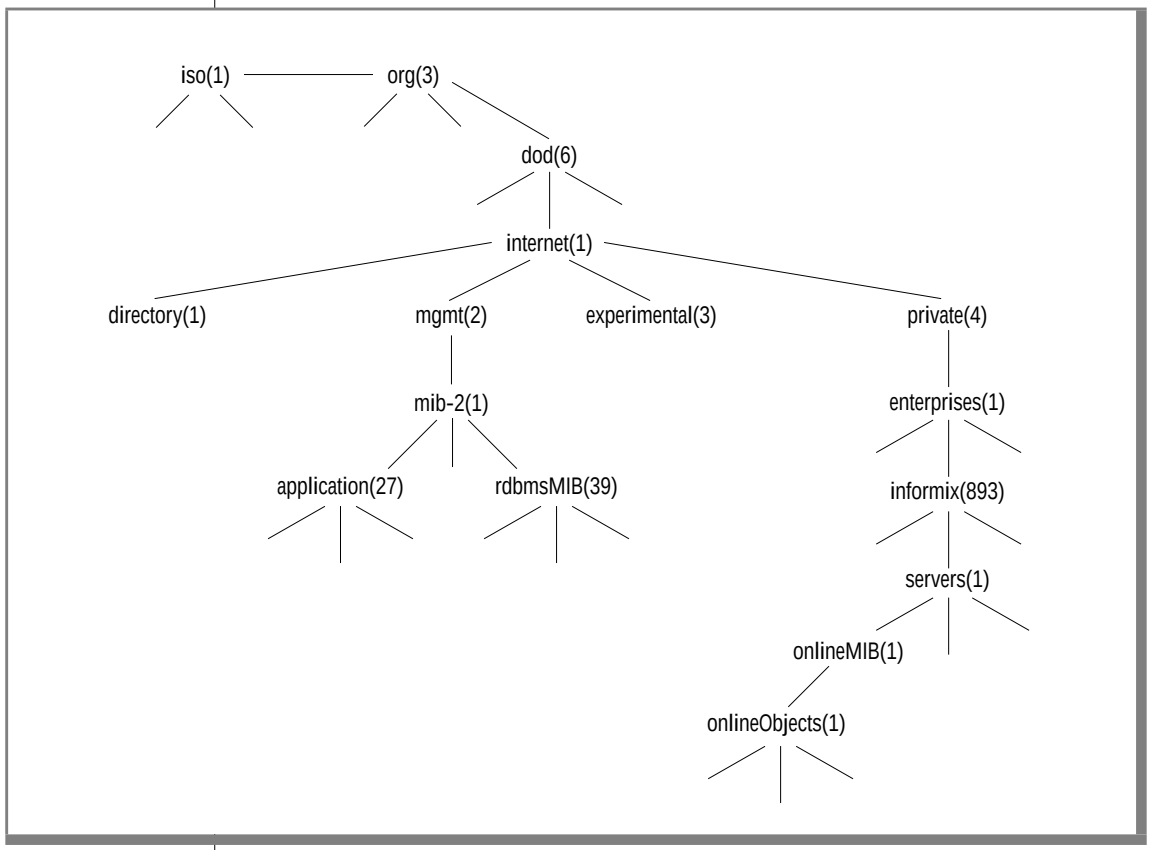
SNMP specifies the contents of the RDBMS MIB and the application MIB. These two MIBs are *public* MIBs. Public MIBs are defined by the Internet Engineering Task Force (IETF) and are the same for all database servers that use SNMP network managers. However, some of the definitions in the RDBMS MIB are purposely somewhat vague, so that the entries can be tailored to the specific database server. For example, the **rdbmsSrvLimitedResourceTable** contains information about the resources (for example, memory, buffers, and disk space) that a database server uses. Each relational-database vendor can decide what resources should be included in that table.

The Informix *private* MIB is a specialized MIB that describes information that is relevant to the specific architecture and features of Informix databases and database servers. A private MIB is unique to the enterprise (in this case, Informix) that defines it.

MIB Hierarchy

All MIBs are part of a hierarchy of information that is defined by the Internet Assigned Numbers Authority (IANA). The hierarchy defines both how objects in a MIB should be named and how the numerical *object identifiers* (OIDs) are derived. For information about IANA, refer to the material listed in "Related Reading" on page 9 of the Introduction. Figure 3-1 on page 3-5 shows the MIB hierarchy.

Figure 3-1
MIB Numerical Hierarchy



The RDBMS MIB is part of the **mgmt** (management) subtree of the Internet hierarchy. Figure 3-1 illustrates the path to the RDBMS MIB, as follows:

```
iso.org.dod.internet.mgmt.mib-2.rdbmsMIB
```

The Informix MIB is part of the **enterprises** subtree of the Internet hierarchy. The path to the Informix branch is as follows:

```
iso.org.dod.internet.private.enterprises.informix
```

The IANA defined the path to the **informix** node. That path never changes. Informix has designed the path below the **informix** node to provide flexibility for future expansion.

The hierarchy below the **informix** node has three more nodes: the **server** node, the **MIB** node, and the **objects** node.

A database administrator rarely, if ever, sees the full path to a node. The path to a node is important because the network manager uses the numerical equivalent of the path to locate objects (that is, pieces of data) uniquely.

The MIB Numerical Hierarchy

Every object in an SNMP database has an *object identifier* (OID). The OID uniquely describes each piece of data that you can retrieve with the network manager. The OIDs are written as a string of numbers separated by dots (.), as in the following example:

```
.1.3.6.1.4.1.893.1.1.1.8.1
```

The first part of the OID is derived from the numbers associated with each node on the hierarchy illustrated in Figure 3-1 on page 3-5. For example, objects in the RDBMS MIB all have OID values that start with the following string:

```
.1.3.6.1.2.39
```

Objects in the application MIB have OID values that start with the following string:

```
.1.3.6.1.2.1.27.1
```

The final part of the OID assigns values sequentially to each table in a MIB, each column within the table, and each item within a column. For example, a piece of data from the application MIB might have an OID such as the following:

```
.1.3.6.1.2.1.27.1.1.8.2
```

As with the path to a node, a database administrator rarely, if ever, sees the full OID.

Informix OIDs

The IANA assigns a unique enterprise identifier to each company that uses the SNMP protocol. The enterprise identifier for Informix is 893. Thus, the OID prefix for all objects in the Informix MIB is as follows:

```
.1.3.6.1.4.1.893
```

For Informix database servers, all of the nodes below the **informix** node have a value of 1. Thus, each piece of information that is accessed through the Informix MIB has an OID that starts with the following prefix:

```
.1.3.6.1.4.1.893.1.1.1
```

MIB Naming Conventions

The name of each MIB table and each object within the table starts with the name of the parent MIB. Thus each table and object in the RDBMS MIB starts with **rdbms**. Each item in the Informix MIB starts with **on**. For example, the MIBs include tables named **rdbmsSrvTable** and **onDbspaceTable**.

The names of objects within a table use the table name (excluding **Table**) as the prefix for the object. For example, **rdbmsSrvVendorName** and **onDbspaceName** are items in the **rdbmsSrvTable** and **onDbspaceTable** tables, respectively.

Chapter

4

Using SNMP

This chapter contains suggestions for ways that you can use the information that **onsnmp** provides to a network manager.

Using a Network Manager

This section discusses some of the monitoring tools that a typical network manager provides. For more detailed information about monitoring tools, refer to the documentation that the vendor of the network manager provides.

Event Notification

When an event occurs that affects the performance or availability of your database server, the network manager can alert you to that condition. Some of the choices that you can make include the following:

- Define the conditions that should be monitored.

  Events that require an alert might include the following:

  - The database server is not available (**onServerMode**).
  - The availability of a database changes (**rdbmsRelState**).
  - A chunk fails (**onChunkStatus**).
  - A table is about to run out of space (**onTablePagesAllocated** and **onTablePagesUsed**).

- Specify how frequently to check for the condition (that is, the frequency of polling).

  When you plan the frequency of polling, you must balance the need for prompt notification of an undesirable condition and the burden that polling puts on the network.

- Specify what happens when the network manager notices an event.

  When an event occurs, you might choose to have an icon blink or change colors.

Data Requests

A data request can be a one-time request or a periodic request. You might issue a one-time request if you want to compare the configuration parameters of two database servers. Data gathered over a period of time can provide statistical information for assessing the performance of your database or indicate when certain resources are overburdened.

Traps

A subagent can be programmed to detect any extraordinary event that might occur in the application and to notify the network about this event. The subagent alerts the network manager by sending a message called an *unsolicited trap*. The network manager can then query the application to determine the cause and extent of the problem.

The Informix subagent, **onsnmp**, sends only one trap, the **rdbmsState-Change** trap. The subagent sends this trap when the status of the database server changes from its current status to any status that is less available. (For example, if a dbspace goes down, the server status changes from full to limited availability.) When the network manager receives a trap, it alerts an administrator.

Using GLS with a Network Manager

The GLS feature of Informix products allows you to work with languages that use code sets other than the standard English code set. However, the SNMP protocol does not provide for the use of different code sets.

The **onsnmp** subagent uses the U.S. English locale when it sends information to the master agent. If **onsnmp** cannot convert the code set of the database to the U.S. English locale, it fails and returns error -23101, as follows:

```
Unable to load locale categories.
```

The **onsnmp** subagent transmits only 7-bit characters; if an eighth bit is present, it is truncated. The **onsnmp** subagent returns *something* when character information is requested, but that *something* might not reflect the name of the database or table. The **onsnmp** subagent transmits numeric information correctly, regardless of the code set that the database uses.

Examples

This section discusses some scenarios that illustrate the functionality of **onsnmp** and the network manager.

A Database Server Becomes Unavailable

A Informix database server becomes unavailable because the database server requires attention from its administrator. (For example, the logical logs might be full.) The **onsnmp** subagent notices that the server is unavailable and sends a trap to the network manager. The network manager is configured so that the icon that represents the department in which the unavailable server resides starts blinking. The blinking attracts the attention of the database administrator who is on duty at the console. The database administrator then sends data requests to determine the cause of the failure.

A User Complains About an Application

A user sends email to the help desk complaining that an update application that uses an Informix database server has stopped responding. The help desk notifies the database administrator, who looks at the table of session-related information to determine the cause of the problem.

Technical Support

A database administrator on a local host calls the technical support representative and reports that the Informix database server throughput for the transactions running in a particular situation is less than expected. From the remote location, the technical support representative queries the **onsnmp** subagent to determine the configuration of the database server, monitors the performance of the server, and identifies the bottleneck.

Appendix

A

MIB Tables

This appendix lists the columns of each table of the MIBs that the Informix subagent uses:

- The application MIB
- The RDBMS MIB
- The Informix private MIB

The brief descriptions provided in this appendix are intended for database administrators who use a network manager. The MIBs themselves include detailed information for each item. As a rule, a database administrator who uses a network manager does not need to refer to the MIBs.

For more information, refer to the documentation supplied by the vendor of the network manager. For an introduction to MIBs, refer to Chapter 3, “The Management Information Bases.”

Developers who are preparing interfaces that access the information provided by the MIB tables should refer to the MIB tables for detailed information. The MIB tables are available in the directory **$INFORMIXDIR/snmp**.

Interpreting the Information in a Network Manager

Network managers accept information that is specified by MIB tables. MIB tables follow SNMP protocols; the tables do *not* represent Informix relational database tables. The Informix MIB is *not* an Informix database schema, and you cannot use a SELECT statement to access the information that a MIB table describes.

Row and Column Information

When you query an Informix database, the information is returned row by row. That is, if you use **dbaccess** to query the **syschunks** table of the **sysmaster** database (`SELECT * FROM syschunks`), **dbaccess** displays all of the information about the first chunk (**chknum**=1), then all of the information about the second chunk (**chknum**=2), and so on.

A network manager, on the other hand, typically displays information in a format that is analogous to a column-by-column display. That is, if you request information about the **onChunkTable**, the network manager displays a list of all of the chunks, followed by a list of the offsets for all of the chunks, and so on.

Indexing of the MIB Tables

The OID for each MIB table is specified by its position with the numerical hierarchy described in "The MIB Numerical Hierarchy" on page 3-6. The definition of each MIB includes a heading with information that allows the network manager to derive the basic table OID.

In addition, the header for each table specifies how each individual item that the table specifies should be indexed. The indexes are analogous to, but not the same as, the primary keys of a relational database table. A table might have one, two, or more indexes.

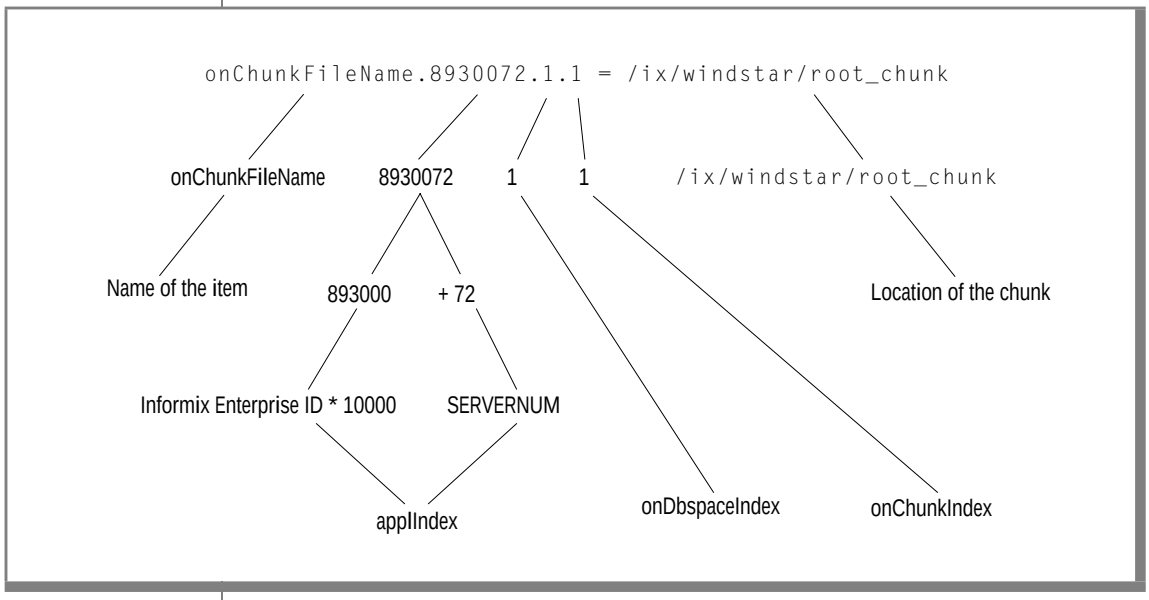
Tip: *In each table in this chapter, the entries above the dotted line refer to the values that form the index for each item in the table. Different network managers display this index in different formats. The entries below the dotted line describe the individual item values that* ***onsnmp*** *provides to the network manager.*

Each index value is concatenated to the basic OID with periods between each value. If an MIB table has several indexes, the indexes are concatenated one after the other. Most network managers display only the final portion of the OID that relates to the table being displayed. Some network managers display the OID as part of the information about each individual item; other network managers display the OID as part of a header for a list of values. Figure 1-2 on page 1-5 and Figure 1-3 on page 1-5 in Chapter 1 illustrate two types of displays.

Displaying Numeric Index Values

Figure A-1 shows an example of indexing information. The figure shows how to interpret a single line from the information returned about the **onChunkTable**. The **onChunkTable** has three indexes: **applIndex**, **onDbspaceIndex**, and **onChunkIndex**.

Figure A-1
Interpreting Index Information



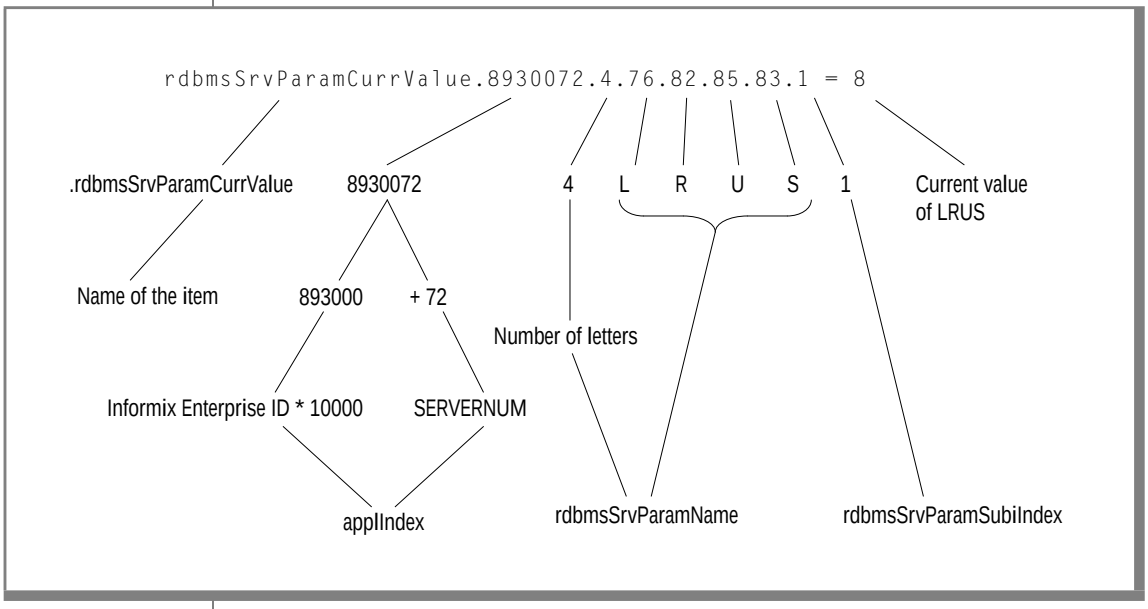
Displaying Alphabetic Index Values

When an index is an alphabetic string, such as the name of a configuration parameter, the OID for that index consists of the number of letters in the name, followed by the ASCII value for each letter, all separated by periods.

Figure A-2 shows an example of indexing information. The figure shows how to interpret a single line from the information returned about the **rdbmsSrvParamTable**. The **rdbmsSrvParamTable** has three indexes, **applIndex**, **rdbmsSrvParamName**, and **rdbmsSrvParamSubIndex**. The second index, **rdbmsSrvParamName**, is the name of a parameter (LRUS). Thus, the second index becomes the series of values: 4.76.82.85.83.

.

Figure A-2
Interpreting Index Information



The Application MIB

The application MIB defines a single table, the **applTable**, that provides a set of general-purpose attributes for each device that the SNMP network manager monitors.

The Application Table

The **applTable[applIndex]** provides general-purpose attributes for the Informix database servers. Every database server on the network is represented by a row in this table, even if the database server supports only local connections.

The following list shows the entries of the **applTable**.

Entry	Description
...Index	A unique integer index that identifies each database server on a host. For Informix, this index is the Informix Enterprise ID (893) *10000 plus SERVERNUM. (If SERVERNUM =0, the value 256 is used instead of 0.)
...Name	The name of the database server
...DirectoryName	The **onsnmp** subagent does not provide information for this attribute. It is left blank.
...Version	The version of the database server
...Uptime	The value of the operating-system time (sysUpTime) at the time the network service application (the database server) was last initialized. If the application was last initialized prior to the last initialization of the network-management subsystem (the **onsnmp** subagent), this entry contains a 0 value.

(1 of 2)

Entry	Description
...OperStatus	The database server operating status: up (1) down (2) halted (3) -- (4) (not used by **onsnmp**) Restarting (5)
...LastChange	The operating-system time stamp when the database server entered its current state
...InboundAssociations	Current number of active database server sessions (SQL CONNECT actions)
...OutboundAssociations	The **onsnmp** subagent does not provide information for this attribute. It is left blank.
...AccumulatedInboundAssociations	Total number of database server sessions started since the server came up
...AccumulatedOutboundAssociations	The **onsnmp** subagent does not provide information for this attribute. It is left blank.
...LastInboundActivity	The operating-system time stamp of the most recent session attempted or disconnected
...LastOutboundActivity	The **onsnmp** subagent does not provide information for this attribute. It is left blank.
...RejectedInboundAssociations	Number of sessions rejected by the database server for administrative reasons or resource limitations
...FailedOutboundAssociations	The **onsnmp** subagent does not provide information for this attribute. It is left blank.

(2 of 2)

In the preceding list, the ellipses (...) indicate that **appl** has been omitted. That is, the formal name for the **Index** entry is **applIndex**.

The RDBMS Public MIB

The RDBMS MIB provides information that is of interest to the administrator of any database server. The items in this MIB are specified by the RDBMS working group of the IETF.

The RDBMS MIB consists of the following tables plus the **rdbmsStateChange** trap.

rdbmsDbTable
rdbmsDbInfoTable
rdbmsDbParamTable
rdbmsDbLimitedResourceTable
rdbmsSrvTable
rdbmsSrvInfoTable
rdbmsSrvParamTable
rdbmsSrvLimitedResourceTable
rdbmsRelTable

The Database Table

The **rdbmsDbTable[rdbmsDbIndex]** lists all databases on the network. For Informix databases, the list includes only databases from active database servers. If an Informix database server is off-line, its databases are not included in this table. Each entry of the **rdbmsDbTable** has the following items.

Entry	Description
...Index	A unique index for this database. The value is SERVERNUM * 10000 plus a value assigned by the subagent. (If SERVERNUM =0, the value 256 is used instead of 0.)
...PrivateMibOID	The Informix OID (.1.3.6.1.4.1.893)
...VendorName	Informix Software, Inc.
...Name	The name of the database
...Contact	The login name of the person who created the database

In the preceding table, the ellipses (...) indicate that **rdbmsDb** has been omitted. That is, the formal name for the **Index** entry in this table is **rdbmsDbIndex**.

The Database Information Table

The **rdbmsDbInfoTable[rdbmsDbIndex]** table provides additional information about a database. This table includes an entry for each database that is open, as indicated by the **rdbmsRelState** attribute for this database in the **rdbmsRelTable**.

Each entry of the **rdbmsDbInfoTable** has the following items.

Entry	Description
rdbmsDbIndex	Refer to "The Database Table" on page A-7.
...ProductName	The name of the product (for example, `INFORMIX-Universal Server`)
...Version	The version number of the database server that created or last restructured this database
...SizeUnits	The size units used for **SizeAllocated** and **SizeUsed**: bytes (1) kbytes (2) mbytes (3) gbytes (4) tbytes (5)
...SizeAllocated	The estimated size allocated for this database
...SizeUsed	The estimated size in use for this database
...LastBackup	The date and time when the latest backup of the database was performed. If a database has never been backed up, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

In the preceding list, the ellipses (...) indicate that **rdbmsDbInfo** has been omitted. That is, the formal name for the **ProductName** entry in this table is **rdbmsDbInfoProductName**.

The Database Parameters Table

The SNMP specification for the RDBMS MIB includes definitions for the **rdbmsDbParamTable**. Informix does not provide any information for this table. The configuration parameters that you might expect (from reading the table name) to find in **rdbmsDbParamTable** are included in the **rdbmsSrvParamTable**.

If you attempt to access the **rdbmsDbParamTable**, the table returns a **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1) error.

The Database Limited-Resources Tables

The SNMP specification for the RDBMS MIB includes definitions for the **rdbmsDbLimitedResourcesTable**. Informix does not provide any information for this table. All information about database limited resources is included in the **rdbmsSrvLimitedResourceTable**.

If you attempt to access the **rdbmsDbLimitedResourcesTable**, the table returns a **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1) error.

The RDBMS Server Table

The **rdbmsSrvTable[applIndex]** contains general information about the database server.

The following table shows the entries of the **rdbmsSvrTable**.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...PrivateMibOID	The Informix OID (`.1.3.6.1.4.1.89`)
...VendorName	This value is always set to `Informix Software, Inc.`
...ProductName	The product name of the database server (for example, `INFORMIX-Universal Server`)
...Contact	This value is always set to `informix`.

In the preceding table, the ellipses (...) indicate that **rdbmsSrv** has been omitted. That is, the formal name for the **Contact** entry is **rdbmsSrvContact**.

The Server Information Table

All of the entries in the **rdbmsSrvInfoTable[applIndex]** table, except **rdbmsStartupTime**, refer to actions that have affected the database server files since the server was last started.

The **rdbmsSrvInfoTable** has the following entries.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...StartupTime	Date and time at which this database server was last started
...FinishedTransactions	Number of transactions completed, either by commit or by abort
...DiskReads	Number of reads
...LogicalReads	Number of logical reads
...DiskWrites	Number of writes
...LogicalWrites	Number of logical writes
...PageReads	Number of page reads
...PageWrites	Number of page writes
...DiskOutOfSpaces	Number of times the database server has been unable to obtain disk space that it wanted
...HandledRequests	Number of requests made to the database server on inbound associations
...RequestRecvs	Number of receive operations that the database server made while it was processing any requests on inbound associations

(1 of 2)

Entry	Description
...RequestSends	Number of send operations made while processing requests handled on inbound associations
...HighwaterInboundAssociations	The greatest number of inbound associations that have been simultaneously open
...MaxInboundAssociations	The greatest number of inbound associations that can be open at the same time

(2 of 2)

In the preceding table, the ellipses (...) indicate that **rdbmsSrvInfo** has been omitted. That is, the formal name for the **StartupTime** entry is **rdbmsSrvInfoStartupTime**.

The Server Parameters Table

The **rdbmsSrvParamTable[applIndex, Name, SubIndex]** table has an entry for each parameter in the **$INFORMIXDIR/etc/$ONCONFIG** file. The **Name** item shows the name of the configuration parameter. The **SubIndex** is 1 for every parameter except DATASKIP, DBSPACETEMP, DBSERVERALIASES, and NETTYPE.

Each entry contains the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...Name	The name of a configuration parameter from the ONCONFIG file. For information about interpreting the index value, refer to "Displaying Alphabetic Index Values" on page A-4.
...Subindex	Value is 1 for every parameter except DATASKIP, DBSPACETEMP, DBSERVERALIASES, and NETTYPE.

(1 of 2)

Entry	Description
...ID	The Informix OID (1.3.6.1.4.1.893) or informix
...CurrValue	The value of the parameter from the ONCONFIG file. This value does not reflect any dynamic changes that you might have made to the configuration parameter.
...Comment	Purpose of the parameter

(2 of 2)

In the preceding table, the ellipses (...) indicate that **rdbmsSrvParam** has been omitted. That is, the formal name for the **Name** entry is **rdbmsSrvParamName**.

The Server Limited-Resource Table

The **rdbmsSrvLimitedResourceTable [applIndex, Name]** has an entry for each limited resource for each database server.

The **SrvLimitedResourceTable** has the following entries.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...Name	The name of the resource. For information about interpreting the index value, refer to "Displaying Alphabetic Index Values" on page A-4.
...ID	The Informix OID (.1.3.6.1.4.1.893) or informix
...Limit	The maximum value that this resource may attain
...Current	The current value for this resource
...Highwater	The maximum value of the resource since the **applUpTime** value was reset
...Failures	Number of times that the system tried to exceed the limit since applUpTime was reset
...Description	A description of the resource, the units used, and so on

In the preceding table, the ellipses (...) indicate that **rdbmsSrvLimited-Resource** has been omitted. That is, the formal name for the **Highwater** entry is **rdbmsSrvLimitedResourceHighwater**.

The **Name** item refers to the name of the resource. An entry (a row) exists in the **rdbmsSrvLimtedResourceTable** for each of the following limited resources.

Resource Name	Comment
BUFFERS	None.
DS_MAX_QUERIES	None.
DS_MAX_SCANS	None.
DS_TOTAL_MEMORY	None.
LOCKS	None.
LTXEHWM	None.
LTXHWM	None.
STACKSIZE	None.
LOGFILES	None.
DBSPACES	The **...Highwater** and **...Failures** entries always show 0 for this resource.
CHUNKS	The **...Highwater** and **...Failures** entries always show 0 for this resource.

The Relative Table

The **rdbmsRelTable[rdbmsDbIndex, applIndex]** table relates the databases and the database servers present on a database server.

Each entry in the **rdbmsRelTable** has the following items.

Entry	Description
rdbmsDbIndex	Refer to "The Database Table" on page A-7.
applIndex	Refer to "The Application Table" on page A-5.
...State	The state of the access of the database server to this database. The possible states are described in the table that follows.
...ActiveTime	The date and time that the database was made active by the database server. If the state is not active, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

In the preceding table, the ellipses (...) indicate that **rdbmsRel** has been omitted. That is, the formal name for the **State** entry is **rdbmsRelState**.

The following table describes the possible values of **rdbmsRelState**.

Value of rdbmsRelState	Description
other (1)	The database server is online, but one of the dbspaces of the database is down.
active (2)	The database server is actively using the database. The database server is online, and a user has opened the database.
available (3)	The database server could use the database, if requested. The database server is online, but no user has opened the database.
restricted (4)	The database is not completely available. The database server is online, and a user has opened the database in exclusive mode.

The Traps Table

The **rdbmsTraps** table describes the traps that the **onsnmp** agent can send to the Network Manager through the Master Agent. The **rdbmsTraps** table contains only one item, as follows.

EntrY	Description
...StateChange	The value of the **rdbmsRelState** has changed to a less accessible state. When the state of the database server changes, a trap is generated for each database that the server manages.

In the preceding list, the ellipses (...) indicate that **rdbmsTraps** has been omitted. That is, the formal name for the **StateChange** entry is **rdbmsTrapsStateChange**.

The Informix Private MIB

The Informix Private MIB (usually referred to as the Informix MIB) provides information that is not included in the Application MIB or the RDBMS MIB. The RDBMS MIB describes database servers and their databases in a vendor-neutral way. The Informix MIB provides information that is specifically relevant for Informix database servers.

The Informix MIB includes the following tables.

onServerTable
onDatabaseTable
onTableTable
onActiveTableTable
onFragmentTable
onDbspaceTable
onChunkTable
onLogicalLogTable
onPhysicalLogTable
onSessionTable
onLockTable
onBarTable
onSQLHostTable

The Server Table

The **onServerTable[applIndex]** table provides status and profile information about each active Informix database server on the network. Each row of the table is associated with a single active Informix database server.

Each entry of the **onServerTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5
...Mode	The current mode of the database server: initializing (1) quiescent (2) fastRecovery (3) backingUp (4) shuttingDown (5) online (6) aborting (7) onlineReadOnly (8)
...CheckpointInProgress	Indicates whether a checkpoint is in progress: yes (1) - checkpoint in progress no (2) - no checkpoint in progress
...PageSize	The size of a page, in bytes
...Threads	Number of active threads
...VPs	Number of virtual processors
...VirtualMemory	The total virtual memory used, in kilobytes
...ResidentMemory	The total resident memory used, in kilobytes
...MessageMemory	The total message memory used, in kilobytes
.. IsamCalls	The sum of all ISAM reads, writes, rewrites, deletes, commits, and rollbacks
... LatchWaits	Number of latch waits
...LockRequests	Number of lock requests
...LockWaits	Number of lock waits

(1 of 2)

Entry	Description
...BufferWaits	Number of buffer waits
...CheckpointWaits	Number of checkpoint waits
...DeadLocks	Number of deadlocks
...LockTimeouts	Number of deadlock time-outs
...LogicalLogRecords	Number of logical-log records
...LogicalLogPageWrites	Number of logical-log page writes
...LogicalLogWrites	Number of logical-log writes
...BufferFlushes	Number of buffer flushes
...ForegroundWrites	Number of foreground writes
...LRUWrites	Number of LRU writes
...ChunkWrites	Number of chunk writes
...ReadAheadPages	Number of read-ahead pages (includes both data and index read-ahead pages)
...ReadAheadPagesUsed	Number of read-ahead pages used
...SequentialScans	Number of sequential scans
...MemorySorts	Number of memory sorts
...DiskSorts	Number of disk sorts
...MaxSortSpace	Maximum disk space used by a sort, in pages
...NetworkReads	Number of network reads
...NetworkWrites	Number of network writes
...PDQCalls	Number of parallel-processing actions performed

(2 of 2)

In the preceding list, the ellipses (...) indicate that **onServer** has been omitted. That is, the formal name for the **Mode** entry in this table is **onServerMode**.

The Database Table

Each row in the **onDatabaseTable[applIndex, rdbmsDbIndex]** table contains additional information about a single active Informix database. No entry exists for a database in this table if the corresponding **rdbmsRelState** indicates that the database server is active but one of the dbspaces is down.

Each entry of the **onDatabaseTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
rdbmsDbIndex	Refer to "The Database Table" on page A-7.
...Dbspace	The default dbspace for this database
...Created	Creation date and time
...Logging	The logging status of the database: none (1) buffered (2) buffered (3) ansi (4)
...OpenStatus	The status of the database: notOpen (1) open (2) openExclusive (3)
...Users	Number of users of this database

In the preceding list, the ellipses (...) indicate that **onDatabase** has been omitted. That is, the formal name for the **Dbspace** entry is **onDatabaseMode**.

The Table Table

Each row in the **onTableTable[applIndex, rdbmsDbIndex, Index]** table contains information about a table in a database.

Each entry of the **onTableTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
rdbmsDbIndex	Refer to "The Database Table" on page A-7.
...Index	The table number, as specified by **tabid** in the system catalog table **systables**
...Name	The table name
...Owner	The owner of the table
...Type	The type of the table: table (1) view (2) privateSynonyn (3) synonym (4)
...LockLevel	The locking level of the table: page (1) row (2)
...Created	Creation date (in string format)
...FirstDbspace	Name of the first (or only) dbspace for this table
...RowSize	Length of a row
...Rows	Number of rows
...Columns	Number of columns
...Indices	Number of indexes
...Extents	Number of extents in use
...PagesAllocated	Total (extent) size allocated to the table, in pages
...PagesUsed	Number of pages in use

(1 of 2)

Entry	Description
...Fragments	Number of fragments for this table
...FragmentStrategy	Fragmentation strategy of this table: roundRobin byExpression tableBased If the table is not fragmented, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).
...ActiveFragments	Number of active fragments. If the table is not fragmented, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

(2 of 2)

In the preceding list, the ellipses (...) indicate that **onTable** has been omitted. That is, the formal name for the **Index** entry in this table is **onTableIndex**.

The Active Table

The **onActiveTableTable[applIndex, rdbmsDbIndex, onTableIndex]** table contains profile information on all open and active tables. For a table that is fragmented, the values in this table are summary versions of the corresponding values from all the fragments of the table.

Each entry of the **onActiveTableTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
rdbmsDbIndex	Refer to "The Database Table" on page A-7.
onTableIndex	Refer to "The Table Table" on page A-19.
...Status	The status of the table: not Busy (1) - The table is not currently in use. busy (2) - The table is in use. dirty (3) - The table has been modified. For a table that is fragmented, the status is derived from all the fragments of the table.

(1 of 2)

Entry	Description
...IsBeingAltered	The current state of the table: yes (1) - The table is being altered (add or drop index, ALTER TABLE, alter page count updated, or pages altered to the latest schema). no (2) - The table is not being altered. For a table that is fragmented, the status is derived from all the fragments of the table.
...Users	Number of users accessing this table
...LockRequests	Number of lock requests
...LockWaits	Number of lock waits
...LockTimeouts	Nubmer of lock timeouts
...IsamReads	Number of ISAM reads
...IsamWrites	Number of ISAM writes
...BufferReads	Number of buffer reads
...BufferWrites	Number of buffer writes

(2 of 2)

In the preceding list, the ellipses (...) indicate that **onActiveTable** has been omitted. That is, the formal name for the **Status** entry in this table is **onActiveStatusIndex**.

The Fragment Table

The **onFragmentTable[applIndex, rdbmsDbIndex, onTableIndex, Index]** table contains profile information on all open and active tables. For a table that is fragmented, the values in this table are summary versions of the corresponding values from all the fragments of the table.

Each entry of the **onFragmentTableTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
rdbmsDbIndex	Refer to "The Database Table" on page A-7.
onTableIndex	Refer to "The Table Table" on page A-19.
...Index	A unique integer index for this fragment
...Type	The type of the table: fragmentedIndex (1) fragmentedTable (2)
...Dbspace	Dbspace name for this fragment
...Expression	The expression text used for fragmentation of this table or index (blank if the fragmentation scheme is round-robin)
...IndexName	The index identifier
...Extents	Number of extents used
...PagesAllocated	Total (extent) size allocated to this fragment, in pages
...PagesUsed	Number of pages used
...IsamReads	Number of ISAM reads. If the fragment is not active, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).
...IsamWrites	Number of ISAM writes. If the fragment is not active, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

In the preceding list, the ellipses (...) indicate that **onFragment** has been omitted. That is, the formal name for the **Index** entry in this table is **onFragmentIndex**.

The DbspaceTable

The **onDbspaceTable[applIndex, Index]** table contains information about all dbspaces.

The **onDbspaceTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...Index	A unique integer identifier for this dbspace, generated by the database server
...Name	Name of the dbspace
...Owner	Login name of the owner
...Created	Creation date
...Chunks	Number of chunks in this dbspace
...Type	The type of the dbspace: regularDbspace (1) temporaryDbspace (2) blobDbspace (3)
...MirrorStatus	The mirroring status of the dbspace: notMirrored (1) mirrored (2) mirrorDisabled (3) newlyMirrored (4)
...RecoveryStatus	The recovery status of the dbspace: noRecoveryNeeded (1) doneRecovery (2) physicallyRecovered (3) logicallyRecovering (4)
...BackupStatus	Backup status of the dbspace: yes (1) - Dbspace is backed up. no (2) - Dbspace is not backed up.

(1 of 2)

Entry	Description
...MiscStatus	Additional dbspace information: none (1) - no additional information aTableDropped (2)
...PagesAllocated	The size of all the primary chunks in this dbspace
...PagesUsed	Number of pages used in all the primary chunks in this dbspace
...BackupDate	The date that the latest backup was performed. If the dbspace has never been backed up, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

(2 of 2)

In the preceding list, the ellipses (...) indicate that **onDbspace** has been omitted. That is, the formal name for the **Index** entry in this table is **onDbspaceIndex**.

The Chunk Table

The **onChunkTable[applIndex, onDbspaceIndex, onChunkIndex]** table contains information about all of the chunks that the Informix database servers use.

The **onChunkTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
onDbspaceIndex	Refer to "The DbspaceTable" on page A-23.
...Index	A unique integer index for this chunk, generated by the database server
...FileName	The pathname for the chunk
...FileOffset	Offset into the device, in pages
...PagesAllocated	Chunk size, in pages

(1 of 3)

Entry	Description
...PagesUsed	Number of pages used
...Type	The type of the chunk: regularChunk (1) blobChunk (2) stageBlob (3)
...Status	The status of the chunk: offline (1) online (2) recovering (3) inconsistent (4) dropped (5)
...Mirroring	The mirroring status of the chunk: notMirrored (1) mirrored (2) newlyMirrored (3)
...Reads	Number of physical-read operations
...PageReads	Number of page reads
...Writes	Number of physical-write operations
...PageWrites	Number of page writes
...MirrorFileName	Pathname of the mirror chunk. If this chunk is not mirrored, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

(2 of 3)

Entry	Description
...MirrorFileOffset	Offset of the mirror, in pages. If this chunk is not mirrored, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).
...MirrorStatus	Current status of mirroring: offline (1) online (2) recovering (3) inconsistent (4) dropped (5) If the chunk is not mirrored, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

(3 of 3)

In the preceding list, the ellipses (...) indicate that **onChunk** has been omitted. That is, the formal name for the **Index** entry in this table is **onChunkIndex**.

The Logical-Log Table

The **onLogicalLogTable[applIndex, onLogicalLogIndex]** table contains information about logical-log files defined for Informix database servers.

The **onLogicalLogTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...Index	The index for this logical-log file
...ID	A unique integer identification number for this logical-log file
...Dbspace	The dbspace name where this log file was created

(1 of 2)

Entry	Description
...Status	The status of this logical-log file: newlyAdded (1) free (2) current (3) used (4) backedUpButNeeded (5)
...ContainsLastCheckpoint	Whether this logical-log file contains the last checkpoint: yes (1) no (2)
...IsTemporary	Whether this logical-log file contains the last checkpoint: yes (1) no (2)
...PagesAllocated	The size of the logical-log file, in pages
...PagesUsed	Number of pages used in this logical-log file
...FillTime	The date and time when this logical-log file last filled up. If the logfile has not been filled, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

(2 of 2)

In the preceding list, the ellipses (...) indicate that **onLogicalLog** has been omitted. That is, the formal name for the **Index** entry in this table is **onLogicalLogIndex**.

The Physical-Log Table

The **onPhysicalLogTable[applIndex]** table contains information about the physical-log files defined for Informix database servers.

The **onPhysicalLogTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...Dbspace	The dbspace name where this physical log was created
...BufferSize	The size of the physical-log buffer, in pages
...BufferUsed	Number of pages of the physical-log buffer that are used
...PageWrites	Number of pages written to the physical log
...Writes	Number of (disk) writes to the physical log
...PagesAllocated	Size of the physical log, in pages
...PagesUsed	Number of pages used

In the preceding list, the ellipses (...) indicate that **onPhysicalLog** has been omitted. That is, the formal name for the **Dbspace** entry in this table is **onPhysicalLogDbspace**.

The Session Table

The **onSessionTable[applIndex, onSessionIndex]** table contains information about each user who is connected to an Informix database server.

The **onSessionTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...Index	A unique integer index for this session
...UserName	Name of the user of the database server. The name is in the form *name@host*(tty).

(1 of 4)

Entry	Description
...UserProgramVersion	The version of the application program in use by the user
...UserProcessId	The process ID of the application program
...UserTime	The amount of time that the user has been connected to the database server, in hundredths of seconds
...State	The state of the user session: idle (1) active (2) waitingOnMutex (3) waitingOnCondition (4) waitingOnLock (5) waitingOnBuffer (6) waitingOnCheckPointing (7) waitingOnLogicalLogWrite (8) waitingOnTransaction (9)
...Database	Currently connected database
...CurrentMemory	Current memory usage, in bytes
...Threads	Number of active threads for this session
...CurrentStack	Average current size of the stack for all threads for this session
...HighwaterStack	Maximum amount of memory used so far by any thread of this session
...LockRequests	Number of lock requests
...LocksHeld	Number of locks held
...LockWaits	Number of lock waits
...LockTimeouts	Number of time-outs for locks
...LogRecords	Number of log records
...IsamReads	Number of ISAM reads
...IsamWrites	Number of ISAM writes
...PageReads	Number of page reads

(2 of 4)

Entry	Description
...PageWrites	Number of page writes
...LongTxs	Number of long transactions
...LogSpace	Logical-log space used, in bytes
...HighwaterLogSpace	Maximum logical-log space ever used by this session
...SqlStatement	The latest SQL statement (truncated to 250 characters)
...SqlIsolation	The SQL isolation level: noTransactions (1) dirtyReads (2) readCommitted (3) cursorRecordLocked (4) repeatableRead (5)
...SqlLockWaitMode	Action to take if isolation level requires a wait: -1 = Wait forever. 0 = Do not wait. >0 = Wait for specified number of seconds.
...SqlEstimatedCost	Estimated cost of the SQL statement as reported by SQLEXPLAIN
...SqlEstimatedRows	Estimated number of rows that will be selected by the SQL statement as reported by SET EXPLAIN
...SqlError	The error number for the last SQL statement
...SqlIsamError	The ISAM error number for the last SQL statement

(3 of 4)

Entry	Description
...TransactionStatus	The status of the transaction: none (1) committing (2) rollingBack (3) rollingHeuristically (4) waiting (5)
...TransactionBeginLog	The unique ID of the logical-log file in which the BEGIN WORK record was logged. If no transaction exists, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).
...TransactionLastLog	The unique ID of the logical-log file in which the last record was logged. If no transaction exists, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

(4 of 4)

In the preceding list, the ellipses (...) indicate that **onSession** has been omitted. That is, the formal name for the **Index** entry in this table is **onSessionIndex**.

The Lock Table

The **onLockTable[applIndex, onSessionIndex, onLockIndex]** table contains information on all the active locks that are currently in use by Informix database servers. For each session, entries exist for both the locks that are in use by the session as well as the locks on which the session is waiting.

The **onLockTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
onSessionIndex	Refer to "The Session Table" on page A-28.
...Index	The index to this entry
...DatabaseName	Name of the database using or waiting for this lock

(1 of 2)

Entry	Description
...TableName	Name of the table using or waiting for this lock
...Type	The type of the lock: byte (1) intentShared (2) shared (3) sharedByRepeatableRead (4) update (5) intentExclusive (6) sharedIntentExclusive (7) exclusive (8) exclusiveByRepeatableRead (9) waiting (10)
...Granularity	The granularity of this lock: table (1) page (2) row (3) index (4)
...RowId	The **rowid** of the locked row
...Waiters	Number of sessions waiting for this lock
...GrantTime	The time that the lock was granted, if the session is using the lock. If no transaction exists, this value is **noSuchInstance** (SNMPV2) or **noSuchName** (SNMPV1).

(2 of 2)

In the preceding list, the ellipses (...) indicate that **onLock** has been omitted. That is, the formal name for the **Index** entry in this table is **onLockIndex**.

The Bar Table

The **onBarTable[applIndex, onBarActivityIndex, onBarObjectIndex]** table contains all the objects that participated in a backup and restore activity. For more information about the items in this table, refer to the *INFORMIX-Universal Server Archive and Backup Guide*.

The **onBarTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...ActivityIndex	The index to this activity
...ObjectIndex	The index to this object
...Name	The name of the object
...Type	The type of the object: blobspace (1) rootDbspace (2) criticalDbspace (3) noncriticalDbspace (4) logicalLog (5)
...Level	The level of the backup action: completeBackup (1) incrementalLevelOne(2) incrementalLevelTwo (3)
...Status	The status of the action on this object 0 = successful non-zero = error number
...TimeStamp	The ending time stamp for this action

In the preceding list, the ellipses (...) indicate that **onBar** has been omitted. That is, the formal name for the **ActivityIndex** entry in this table is **onBarActivityIndex**.

The sqlhosts Table

The **onSqlHostTable[applIndex, onSqlHostIndex]** table contains all of the entries of the **sqlhosts** connectivity files for each active server.

The **onSqlHostTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...Index	The index to this connection entry
...Name	Host name of the database server
...NetType	Connection type
...ServerName	Name of the database server, or its alias
...ServiceName	Service name

In the preceding list, the ellipses (...) indicate that **onSqlHost** has been omitted. That is, the formal name for the **Index** entry in this table is **onSqlHostIndex**.

The SNMP Maintenance Table

The **onSnmpTable[applIndex, onSnmpIndex]** table is for maintenance. This table contains information about all the tables that the subagent maintains.

The end user cannot access this table.

The **onSnmpTable** has the following items.

Entry	Description
applIndex	Refer to "The Application Table" on page A-5.
...Index	A sequential index to an entry
...Name	Name of the ONSNMP table
...OID	The OID assigned to the table
...RefreshType	The refresh strategy for the table: oneShot (1) onDemand (2) onTimer (3)
...RefreshInterval	The time period between refreshes to this table, in hundredths of a second
...NextRefresh	Time that remains after which the table will be refreshed, in hundredths of a second
...Refreshes	Number of refreshes done
...HoldInterval	Time period after which the contents of the table become invalid, in hundredths of a second
...Rows	Number of rows in the table
...Columns	Number of columns in the table
...Retries	Number of retries for access
...Failures	Number of access failures

In the preceding list, the ellipses (...) indicate that **onSnmp** has been omitted. That is, the formal name for the **Index** entry in this table is **onSnmpIndex**.

Index