Informix® SNMP Subagent Guide

Informix Dynamic Server, Version 7.3
Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options, Version 8.2

Informix Dynamic Server, Developer Edition, Version 7.3
Informix Dynamic Server, Workgroup Edition, Version 7.3

Version 2.0
February 1998
Part No. 000-4371

Published by INFORMIX® Press

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025-1032

Documentation Team: Lynne Casey, Diana Chase, Barbara Nomiyama

Table of Contents

Chapter 2 Informix Implementation of SNMP

Chapter 3 MIB Reference

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

This manual describes the simple network management protocol (SNMP) and the software that you need in order to use SNMP. Each Informix database server provides some of this software, and you must buy more software from another vendor.

Types of Users

This manual is for the following users:

- Database server administrators
- Backup operators
- Performance engineers

This manual assumes that you have the following background:

- A working knowledge of your computer, your operating system, and the utilities that your operating system provides
- Some experience with database server administration, operating-system administration, or network administration

If you have limited experience with relational databases, SQL, or your operating system, see your *Getting Started* manual for a list of supplementary titles.

Software Dependencies

This manual assumes that you are using one of the following database servers:

- Informix Dynamic Server, Version 7.3
- Informix Dynamic Server, Developer Edition, Version 7.3
- Informix Dynamic Server, Workgroup Edition, Version 7.3
- Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options, Version 8.2

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.

For instructions on how to specify a nondefault locale, additional syntax, and other considerations related to GLS locales, see the *Informix Guide to GLS Functionality*.

Important: *SNMPv1 and SNMPv2 do not recognize non-English code sets. For more information, see* *"GLS and SNMP" on page 2-26*.

Demonstration Databases

The DB-Access utility, which is provided with your Informix database server products, includes a demonstration database called **stores7** that contains information about a fictitious wholesale sporting-goods distributor. You can use SQL scripts provided with DB-Access to derive a second database, called **sales_demo**. This database illustrates a dimensional schema for data-warehousing applications. Sample command files are also included for creating and populating these databases.

Many examples in Informix manuals are based on the **stores7** demonstration database. The **stores7** database is described in detail and its contents are listed in the *Informix Guide to SQL: Reference*.

The scripts that you use to install the demonstration databases reside in the **$INFORMIXDIR/bin** directory on UNIX platforms and the **%INFORMIXDIR%\bin** directory in Windows NT environments. For a complete explanation of how to create and populate the **stores7** demonstration database, see the *DB-Access User Manual*. For an explanation of how to create and populate the **sales_demo** database, see the *Informix Guide to Database Design and Implementation*.

New Features

For a comprehensive list of new features, see the release notes for your database server.

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.
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 within a table or section.

Tip: *When you are instructed to "enter" characters or to "execute" a command, immediately press* RETURN *after you type the indicated information on your keyboard. When you are instructed to "type" the text or to "press" other keys, you do not need to press* RETURN.

Icon Conventions

Throughout the documentation, you will find text that is identified by 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.

Feature, Product, and Platform Icons

Feature, product, and platform icons identify paragraphs that contain feature-specific, product-specific, or platform-specific information.

Icon	Description
AD/XP	Identifies information that is specific to Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options.
GLS	Identifies information that relates to the Informix Global Language Support (GLS) feature.

(1 of 2)

Icon	Description
IDS	Identifies information that is specific to Dynamic Server and its editions. However, in some cases, the identified section applies only to Informix Dynamic Server and not to Informix Dynamic Server, Workgroup and Developer Editions. Such information is clearly identified.
UD	Identifies information that is specific to Informix Dynamic Server with Universal Data Option.
UNIX	Identifies information that is specific to UNIX.
W/D	Identifies information that is specific to Informix Dynamic Server, Workgroup and Developer Editions.
WIN NT	Identifies information that is specific to Windows NT environments.

(2 of 2)

These icons can apply to a row in a table, one or more paragraphs, or an entire section. If an icon appears next to a section heading, the information that applies to the indicated feature, product, or platform ends at the next heading at the same or higher level. A ♦ symbol indicates the end of the feature-, product-, or platform-specific information that appears within a table or a set of paragraphs within a section.

Additional Documentation

For more information, see the following types of documentation:

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

On-Line Manuals

An Answers OnLine 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 the installation insert that accompanies Answers OnLine.

Printed Manuals

To order printed manuals, call 1-800-331-1763 or send email to moreinfo@informix.com. Please provide the following information when you place your order:

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

Error Message Files

Informix software products provide ASCII files that contain all of the Informix error messages and their corrective actions. For a detailed description of these error messages, see *Informix Error Messages* in Answers OnLine.

UNIX

To read the error messages on UNIX, you can use the following commands.

Command	Description
finderr	Displays error messages on-line
rofferr	Formats error messages for printing

♦

WIN NT

To read error messages and corrective actions in Windows NT environments, use the **Informix Find Error** utility. To display this utility, choose **Start→Programs→Informix** from the Task Bar. ♦

Documentation Notes, Release Notes, Machine Notes

In addition to printed documentation, the following sections describe the on-line files that supplement the information in this manual. Please examine these files before you begin using your database server. They contain vital information about application and performance issues.

UNIX

On UNIX, the following on-line files appear in the **$INFORMIXDIR/release/en_us/0333** directory.

On-Line File	Purpose
SNMPDOC_*x.y*	The documentation-notes file for your version of this manual describes features that are not covered in the manual or that have been modified since publication. Replace ***x.y*** in the filename with the version number of your database server to derive the name of the documentation-notes file for this manual.
SERVERS_*x.y*	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. Replace ***x.y*** in the filename with the version number of your database server to derive the name of the release-notes file.
IDS_*x.y*	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. Replace ***x.y*** in the filename with the version number of your database server to derive the name of the machine-notes file.

♦

WIN NT

In Windows NT environments, the following items appear in the Informix folder. To display this folder, choose **Start→Programs→Informix** from the Task Bar.

Item	Description
Documentation Notes	This item includes additions or corrections to manuals, along with information about features that may not be covered in the manuals or that have been modified since publication.
Release Notes	This item 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.

Machine notes do not apply to Windows NT platforms. ♦

Related Reading

The following publications provide additional information about the topics that are discussed in this manual. For a list of publications that provide an introduction to database servers and operating-system platforms, see your *Getting Started* manual.

SNMP Information

For general information about SNMP, see 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)

ASN.1 Information

Abstract syntax notation one (ASN.1) is the standard language for describing data structures. For information about ASN.1, see 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 RFCs describe the Application and RDBMS MIBs:

- 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.

WIN NT

Windows NT Information

Informix manuals assume that you are familiar with your computer operating system. If you are using Windows NT and have limited Windows NT system experience, consult your operating-system manual or a good introductory text before you read this manual. The following texts provide an introduction to Windows NT:

- *Using Windows NT Workstation 3.51* by Paul Sanna (Que, 1996)
- *Microsoft Windows NT Resource Kit* by Russ Blake (Microsoft Press, 1995)
- *NT Server Management and Control* by Kenneth L. Spencer (Prentice-Hall, 1995)

- *Windows NT Administration* by Marshall Brain and Shay Woodard (Prentice-Hall, 1994)
- *Windows NT Network Programming* by Ralph Davis (Addison-Wesley, 1994)
- *Inside Windows NT* by Helen Custer (Microsoft Press, 1993)

Compliance with Industry Standards

The Informix subagent OnSNMP complies with the SNMPV1 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:

650-926-6571

We appreciate your feedback.

Chapter

1

SNMP Concepts

Simple Network Management Protocol (SNMP) is a published, open standard for network management. SNMP lets hardware and software components on networks provide information to network administrators. This chapter provides a brief introduction to SNMP.

Purpose of SNMP

The original purpose of SNMP was to let network administrators remotely manage an Internet system. However, the design of SNMP lets network administrators manage applications as well as systems. SNMP does the following:

- Hides the underlying system network
- Lets you manage and monitor all network components from one console

Event Notification

When an event occurs that affects the performance or availability of a managed component, SNMP software can alert you to that condition. The following list describes some of the decisions that you can make about event notification:

- Define the conditions that should be monitored.
- Specify how frequently to poll for each condition.

  When you determine the polling frequency, you must balance the need for prompt notification of an undesirable condition and the burden that polling puts on the network.
- Specify how the SNMP software notifies you of an event.

  You might choose to have an icon blink or change colors when an event occurs.

Data Requests

A data request can be a one-time request or a periodic request. A one-time request is useful for comparing the data for two managed components. Periodic requests are useful for accumulating statistical information about a managed component.

Traps

You can configure SNMP software to detect extraordinary events and notify you when they occur. The following list describes some of the decisions that you can make about traps:

- Define the conditions that should generate a trap.
- Specify how the SNMP software notifies you of a trap.

  You might choose to have an icon blink or change colors when a trap occurs.
- Specify how the SNMP software responds to a trap.

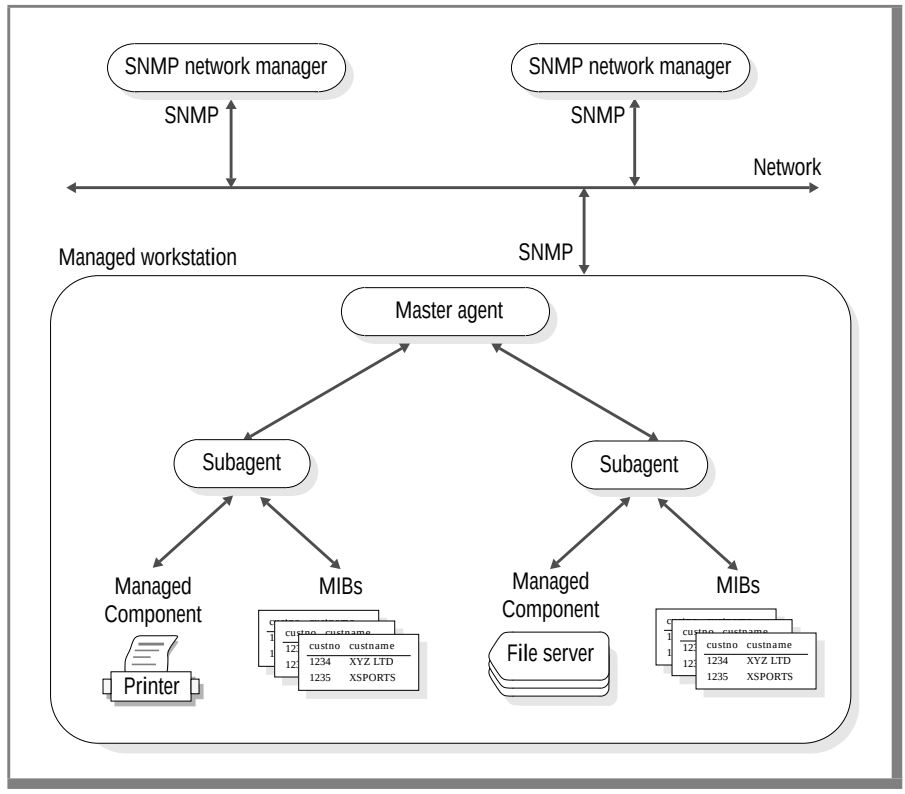
  The SNMP software can query the managed component to determine the cause and extent of the problem.

SNMP Architecture

As Figure 1-1 illustrates, the SNMP architecture includes the following layers:

- SNMP network managers
- Master agents
- Subagents
- Managed components

Figure 1-1
SNMP Architecture



A network can have multiple SNMP network managers. Each workstation can have one master agent. The SNMP network managers and master agents use SNMP protocols to communicate with each other. Each managed component has a corresponding subagent and MIBs. SNMP does not specify the protocol for communications between master agents and subagents.

SNMP Network Managers

An SNMP network manager is a program that asks for information from master agents and displays that information. Most SNMP network managers let you select the items to monitor and the form in which to display the information. An SNMP network manager typically provides the following features:

- Remote monitoring of managed components
- Low-impact sampling of the performance of a managed component
- Correlation of managed component metrics with related system and network metrics
- Graphical presentation of information

Many vendors of hardware and network services have created SNMP network managers. The following list names some SNMP network managers:

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

SNMP network managers use a connectionless protocol, which means that each exchange between an SNMP network manager and a master agent is a separate transaction. A connectionless protocol lets the SNMP network manager perform the following actions:

- Gather information without putting an excessive load on the network
- Function in an environment where heavy traffic can cause network problems

Most SNMP network managers provide graphical user interfaces such as the one illustrated in Figure 1-2. With this SNMP network manager, you select a node to monitor and then choose specific information from a menu.

Figure 1-2
SNMP Network Manager Example

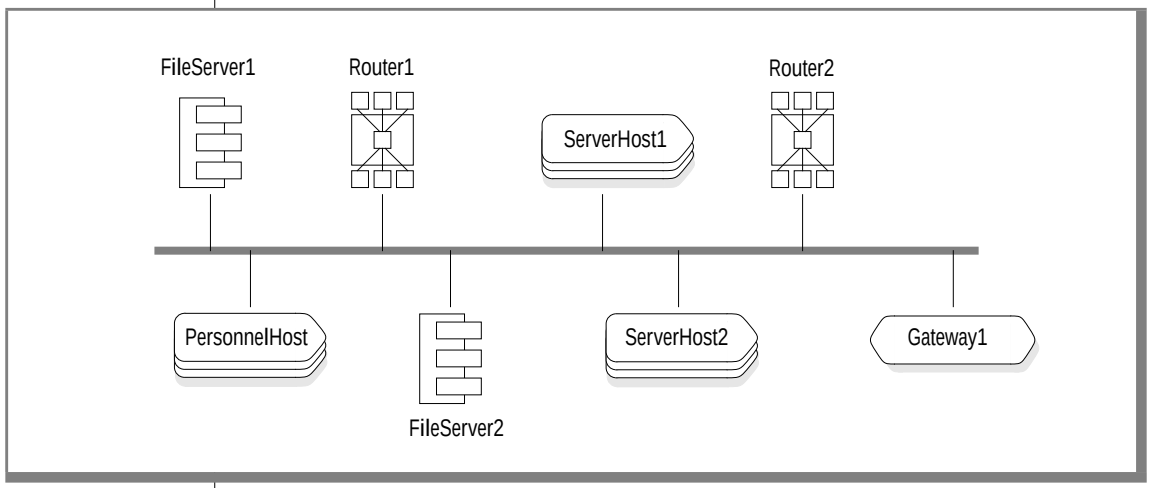


Figure 1-3 shows how an SNMP network manager might display information about the databases on a network. In this example, there is only one database on the network.

```
Jan 16 1998 [ smoke ] : RDBMS-MIB.rdbmsDbTable
KEY = 72000003
rdbmsDbName = CustomerData
rdbmsDbName.72000003 = AnotherData
```

Figure 1-3
Example of Monitoring Information

Figure 1-4 shows how a different SNMP network manager could display the same information.

```
rdbmsDbPrivateMibOID.72000003 = .1.3.6.1.4.1.893
rdbmsDbVendorName.72000003 = Informix Software, Inc.
rdbmsDbName.72000003 = CustomerData
rdbmsDbContact.72000003 = John Doe
```

Figure 1-4
Example of Monitoring Information

In addition to text, an SNMP network manager might also display graphs or charts, as Figure 1-5 illustrates.

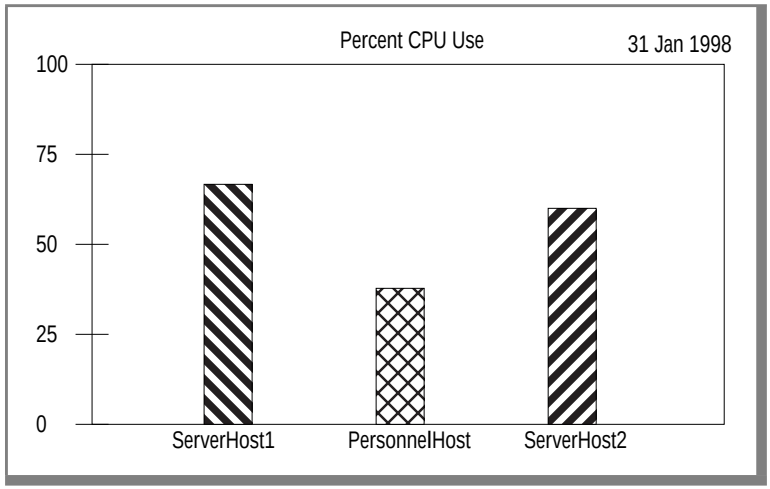


Figure 1-5
Example of Monitoring Information

Master Agents

A master agent is a software program that provides the interface between an SNMP network manager and a subagent. Each workstation that includes a managed component needs to have a master agent. Each managed workstation can have a different master agent. A master agent performs the following tasks:

1. Parses requests from the SNMP network manager
2. Routes requests from the SNMP network manager to the subagents
3. Collects and formats responses from the subagents
4. Returns the responses to the SNMP network manager
5. Notifies the SNMP network manager when a request is invalid or information is unavailable

Subagents

A subagent is a software program that provides information to a master agent. Each managed component has a corresponding subagent. A subagent performs the following tasks:

1. Receives requests from the master agent
2. Collects the requested information
3. Returns the information to the master agent
4. Notifies the master agent when a request is invalid or information is unavailable

Managed Components

A managed component is a hardware or software component that provides a subagent. For example, database servers, operating systems, routers, and printers can be managed components if they provide subagents.

MIBs

A Management Information Base (MIB) is a group of tables that specify the information that a subagent provides to a master agent. MIBs follow SNMP protocols.

MIBs use a common interface definition language. The Structure of Management Information (SMI) defines this language and dictates how to use Abstract Syntax Notation One (ASN.1) to describe each table in the MIBS. For more information about SMI and ASN.1, see the documents listed in"Related Reading" on page 11 of the Introduction.

Naming Conventions

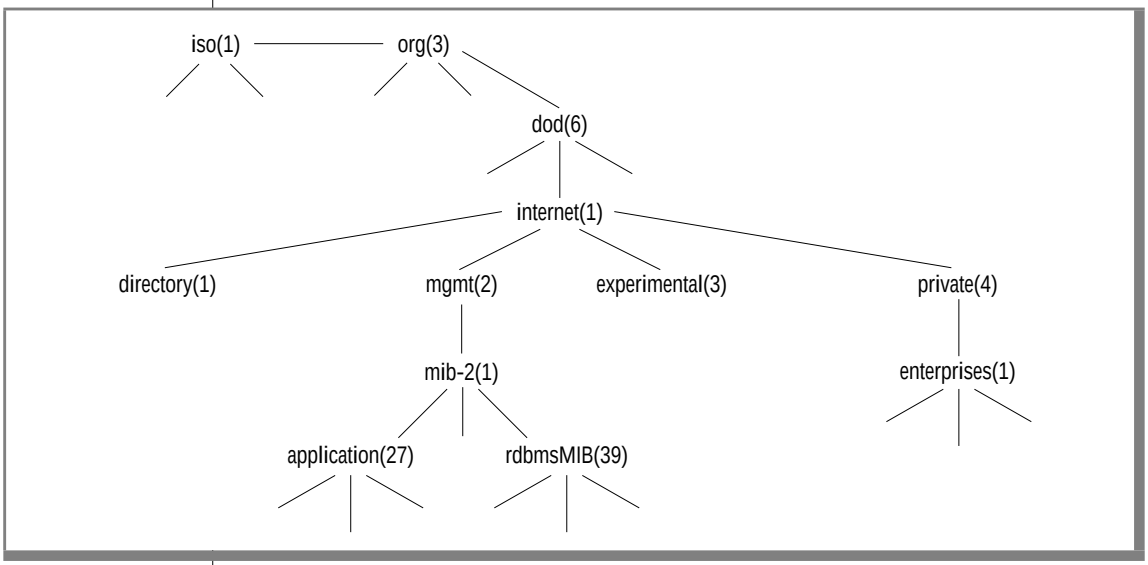
The name of each MIB table starts with the name of the MIB. Thus each table in the RDBMS MIB starts with **rdbms**. For example, the RDBMS MIB includes tables that are named **rdbmsSrvTable** and **rdbmsDbInfoTable**.

The name of each column in a MIB table starts with the name of the table, excluding **Table**. Thus, each column in **rdbmsSrvTable** starts with **rdbmsSrv**. For example, **rdbmsSrvVendorName** and **rdbmsSrvProductName** are columns in **rdbmsSrvTable**.

Hierarchy

All MIBs are part of an information hierarchy that the Internet Assigned Numbers Authority (IANA) defines. The hierarchy defines how to name tables and columns and how to derive the numerical object identifiers (OIDs). For information about IANA, see the documents listed in "Related Reading" on page 11 of the Introduction. Figure 1-6 shows the MIB hierarchy.

Figure 1-6
MIB Hierarchy



Even though you rarely see the full path to a table, column, or value, the path is important because the SNMP components use the numerical equivalent of the path to locate data. For example, the following value is the path to the Application MIB:

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

An OID is the numerical equivalent of a path. The OID uniquely describes each piece of data that an SNMP network manager can obtain. An OID is written as a string of numbers separated by periods (.). For example, the following value is the OID for the Application MIB:

```
.1.3.6.1.2.1.27
```

The following value is the OID for a value in the Application MIB:

```
.1.3.6.1.2.1.27.1.1.8.2
```

The first part of this OID is the OID for the Application MIB. The final part of the OID assigns values sequentially to each table in the MIB, each column in the table, and each value in a column.

Chapter

2

Informix Implementation of SNMP

This chapter describes how Informix uses SNMP to manage Informix database servers. The Informix implementation consists of the following components:

- Master agent

  UNIX

  On UNIX, Informix provides a master agent. ♦

  WIN NT

  In Windows NT environments, you need to install the Microsoft SNMP Extendible master agent. ♦

- Subagent

  The subagent for Informix database servers and coservers is OnSNMP.

  AD/XP

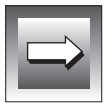
  Only Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options provides coservers. ♦

- Managed components

  In the Informix implementation of SNMP, each database server and each coserver is a managed component.

- MIBs

  OnSNMP uses several MIBs.



Important: *Informix does not provide an SNMP network manager. You need to buy one from another vendor as described in* *"SNMP Network Managers" on page 1-6*.

Purpose of SNMP

The Informix implementation of SNMP lets database administrators monitor and manage Informix database servers, coservers, and databases.

Event Notification

You can configure an SNMP network manager to notify you when a specific event occurs. An event usually has a corresponding column in a MIB table. The following table describes four possible events and the MIB columns that correspond to them.

Event	MIB Column
A database server or coserver is not available.	**onServerMode**
Database availability changed.	**rdbmsRelState**
A chunk failed.	**onChunkStatus**
A table is running out of space.	**onTablePagesAllocated** **onTablePagesUsed**

For example, a user can discover that an application that uses an Informix database server stopped responding. The user can send email to the help desk to report this problem. The help desk can tell you about the problem, and you can look at **onSessionTable** to determine the cause of the problem.

Data Requests

You can issue a one-time data request to compare the configuration parameters of two database servers. You can issue periodic data requests to provide statistical information for assessing database performance or resource allocation.

For example, a person who is using a database that is on a local host can call a remote technical support representative to report a problem. The problem might be that the throughput for the transactions running in a particular situation is less than expected. From the remote location, the technical support representative can query an SNMP network manager to determine the database server configuration, monitor the database server performance, and identify the bottleneck. OnSNMP provides this information to SNMP network managers via the master agent.

Traps

When the status of the database server changes from its current status to any status that is less available, OnSNMP sends a message to the SNMP network managers. For example, if a dbspace goes down, the database server status changes from full to limited availability. The message that OnSNMP sends is **rdbmsStateChange**, which is an unsolicited trap. When an SNMP network manager notifies you that it received an **rdbmsStateChange** trap, you can query the database server that generated the trap to determine the cause and extent of the problem.

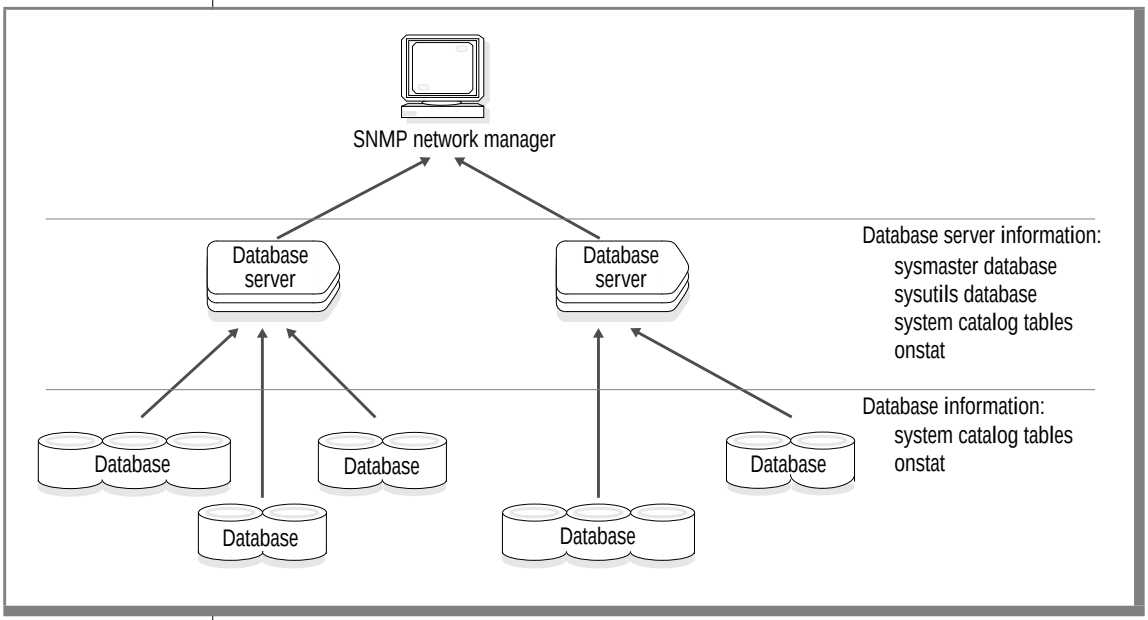
For example, the logical logs for a database server might become full and cause the database server to become unavailable. OnSNMP can notice that the database server is unavailable and send an **rdbmsStateChange** trap to an SNMP network manager. The SNMP network manager can make an icon blink to notify you of the problem. You can then send data requests to determine the cause of the failure.

For information about traps and the EMANATE master agent, see "Installing and Configuring a Master Agent" on page 2-11.

Information That OnSNMP Provides

Most of the information that OnSNMP provides is 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 or coserver. OnSNMP provides information that lets an SNMP network manager monitor all of the Informix databases that are on a network. Figure 2-1 illustrates this point.

Figure 2-1
Monitoring Informix Databases



SNMP Standard

The SNMP standard has two versions: SNMPv1 and SNMPv2. The following table lists the versions of the SNMP standard with which OnSNMP for each platform complies. For information about these standards, see the documents listed in "Related Reading" on page 11 of the Introduction.

Platform	Version of the SNMP Standard
UNIX	SNMPv1 AND SNMPv2
Windows NT	SNMPv1

SNMP Architecture

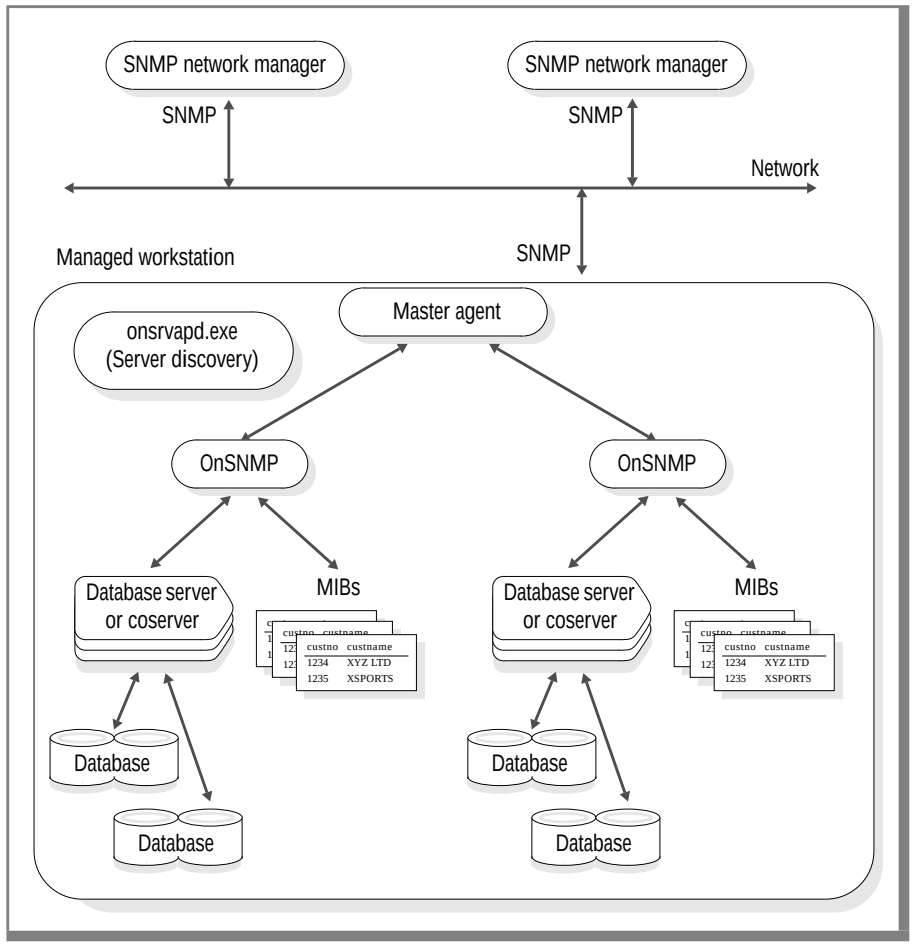
The architecture for the Informix implementation of SNMP depends on your platform.

UNIX

Informix SNMP Architecture on UNIX

Figure 2-2 shows the SNMP architecture for the Informix database servers on UNIX. Each managed workstation runs one master agent and one server discovery process. There is one OnSNMP process for each database server or coserver.

Figure 2-2
Informix SNMP Architecture on UNIX

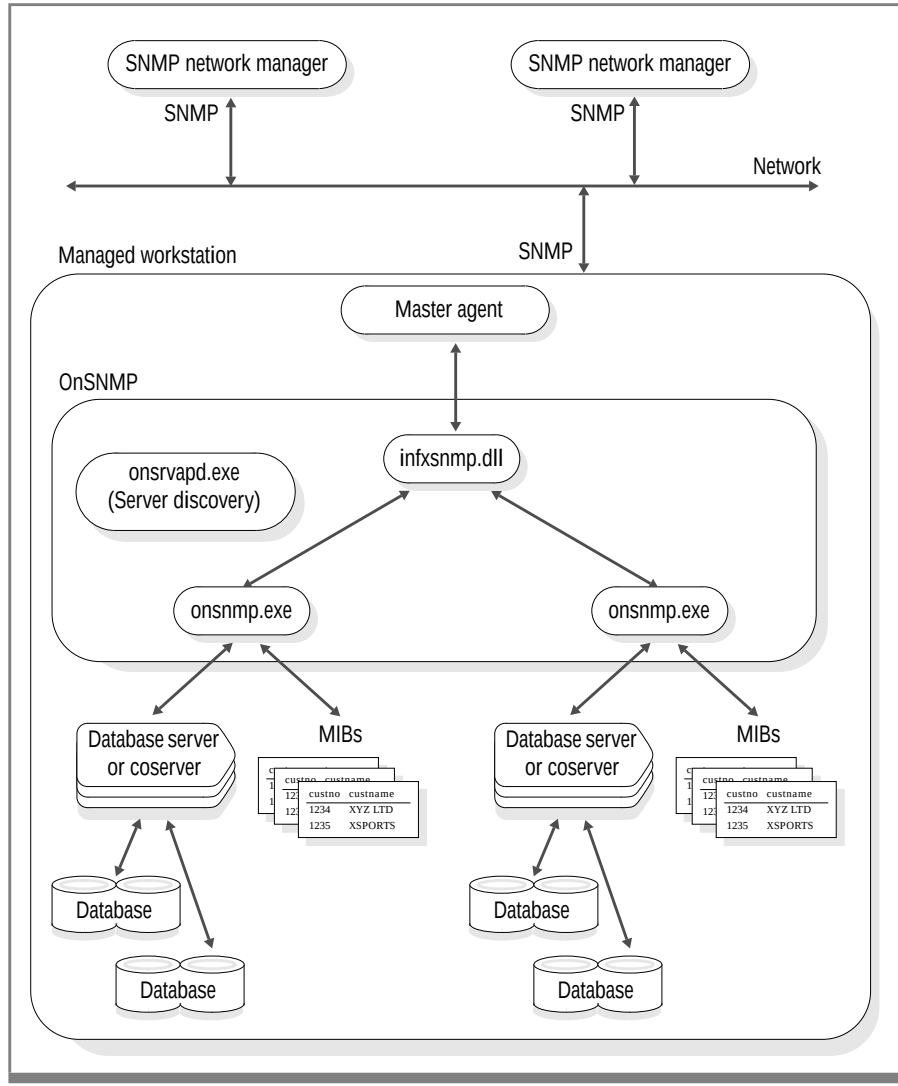


WIN NT

Informix SNMP Architecture on Windows NT

Figure 2-3 on page 2-10 shows the SNMP architecture for the Informix database servers in Windows NT environments. There is one SNMP network manager for the entire network. Each managed workstation runs one master agent. The master agent and the SNMP network manager use SNMP to communicate with each other. Each managed workstation runs one server discovery process and one **infxsnmp.dll**. There is one **onsnmp** executable file for each database server or coserver. OnSNMP and the master agent do not need to use SNMP to communicate with each other.

Figure 2-3
Informix SNMP Architecture on Windows NT



Working with SNMP

To use the Informix implementation of SNMP, you must install and start the following software:

- An SNMP network manager on a network management workstation
- A master agent on each workstation that includes an Informix database server or coserver
- An Informix database server or coserver

  When you install an Informix database server or coserver, the installation procedure installs the OnSNMP subagent and the server discovery process.

Important: *Informix does not provide an SNMP network manager. You need to buy one from another vendor, as described in* *"SNMP Network Managers" on page 1-6*.

UNIX

UNIX Master Agents

On UNIX, Informix provides master agents through licensing agreements with master-agent vendors. The following table lists these master agents. You can also buy and use any other master agent that is suitable for your platform.

Master Agent	Company	Web Site
EMANATE, Version 14.2	SNMP Research	www.snmp.com/emanateintro.html
Patrol, Version 2.3	BMC Software	www.bmc.com/products/snmp/ma.html

Additionally, each DEC workstation provides a master agent. For information about the DEC master agent, see the DEC documentation.

Installing and Configuring a Master Agent

The Informix installation procedure installs the master agents that Informix provides. If you bought a master agent from another vendor, follow the installation instructions that the vendor provides.

To configure the EMANATE master agent

1. Set the following environment variables:
   - Make sure that the **PATH** environment variable includes **$INFORMIXDIR/bin**.
   - Set **SR_AGT_CONF_DIR** to the directory for the EMANATE configuration file. For the location of the EMANATE configuration file, see the Release Notes.
   - Set **SR_LOG_DIR** to the directory for the EMANATE log file. For the location of the EMANATE configuration file, see the Release Notes.
2. Make sure that either the Network Information Services or the **/etc/services** file configures UDP ports 161 and 162 as the SNMP ports.
   a. You can use the **grep** command to search **/etc/services** for `snmp`. The output from **grep** should be similar to the following lines:
   ```
   snmp      161/udp
   snmp-trap 162/udp
   ```
   b. Make sure that UDP port 161 is available so that the master agent can be the owner of the port.
3. If necessary, modify the configuration parameters. For the location of the EMANATE configuration file, see the Release Notes. For information about the EMANATE configuration values, see the SNMP Research Web site, which is listed in "UNIX Master Agents" on page 2-11.
4. To configure a workstation to receive an SNMP trap, create a transport entry and a notify entry in the EMANATE configuration file. For the location of the EMANATE configuration file, see the Release Notes. For information about the EMANATE configuration values, see the SNMP Research Web site, which is listed in "UNIX Master Agents" on page 2-11.

To configure the Patrol master agent

1. Make sure that the **PATH** environment variable includes **$INFORMIXDIR/bin**.
2. Make sure that either the Network Information Services or the **/etc/services** file configures UDP ports 161 and 162 as the SNMP ports and port 199 as the SMUX port.
   a. You can use the **grep** command to search **/etc/services** for `snmp` and `smux`. The output from **grep** should be similar to the following lines:

      ```
      snmp       161/udp
      snmp-trap  162/udp
      smux       199
      ```

   b. Make sure that UDP port 161 is available so that the master agent can be the owner of the port.
3. If necessary, modify the configuration parameters. For the location of the Patrol configuration file, see the Release Notes. For information about the Patrol configuration values, see the BMC Software Web site, which is listed in "UNIX Master Agents" on page 2-11.

Starting and Stopping a Master Agent

OnSNMP expects a master agent to be running when it starts and during the entire time that it is running. Therefore, start the master agent before you start an Informix database server or coserver, and stop all Informix database servers and coservers on a workstation before you stop the master agent.

The best way to start a master agent is to include it in the startup procedure for the workstation. However, you can start a master agent manually if you desire. Similarly, the best way to stop a master agent is to let the workstation stop it during the shut down procedure. However, you can stop a master agent manually if you desire. Additionally, while a master agent is running, you can make sure that it is running correctly.

If you bought a master agent from another vendor, follow the instructions that the vendor provides.

Starting EMANATE Automatically at Startup

To start EMANATE automatically at startup, add the following command to the startup procedure:

```
snmpdm &
```

The following table describes the command-line options that you can include.

Option	Description
-apall	Turn on all messages.
-aperror	Turn on error messages. Error messages are already turned on by default.
-aptrace	Turn on trace messages.
-apwarn	Turn on warning messages. Warning messages are already turned on by default.
-d	Run the master agent in the foreground.

Starting EMANATE Manually

To start EMANATE manually, perform the following steps:

1. Log in as **root**.

   If you do not have **root** user privileges, ask your system administrator to start the master agent.

2. Stop or kill any master agents and daemons that are running on the workstation.

3. Enter the following command:

```
snmpdm &
```

The following table describes the command-line options that you can include.

Option	Description
-apall	Turn on all messages.
-aperror	Turn on error messages. Error messages are already turned on by default.
-aptrace	Turn on trace messages.
-apwarn	Turn on warning messages. Warning messages are already turned on by default.
-d	Run the master agent in the foreground.

Making Sure That EMANATE Is Running Correctly

To make sure that EMANATE is running correctly, perform the following steps:

1. Check the log file to make sure that the master agent has not generated any errors. For the location of the log file, see the Release Notes.
2. Verify that **snmpdm** is running.

Stopping EMANATE Manually

To stop EMANATE manually, perform the following steps:

1. Log in as **root**.

   If you do not have **root** user privileges, ask your system administrator to stop the master agent.
2. Kill the **snmpdm** process.

Starting Patrol Automatically at Startup

To start Patrol automatically at startup, add the following command to the startup procedure:

```
snmpdp $INFORMIXDIR/sqldist/snmp/peer/CONFIG NOV &
```

Starting Patrol Manually

To start Patrol manually, perform the following steps:

1. Log in as **root**.

   If you do not have **root** user privileges, ask your system administrator to start the master agent.
2. Stop or kill any master agents and daemons that are running on the workstation.
3. Enter the following command:

   ```
   snmpdp $INFORMIXDIR/sqldist/snmp/peer/CONFIG NOV &
   ```

Making Sure That Patrol Is Running Correctly

To make sure that Patrol is running correctly, perform the following steps:

1. Check the log file to make sure that the master agent has not generated any errors. For the location of the log file, see the Release Notes.
2. Verify that **snmpdp** is running.

Stopping Patrol Manually

To stop Patrol manually, perform the following steps:

1. Log in as **root**.

   If you do not have **root** user privileges, ask your system administrator to stop the master agent.
2. Kill the **snmpdp** process.

UNIX

UNIX Subagent

On UNIX, OnSNMP consists of the **onsnmp** program. When you install an Informix database server or coserver on UNIX, the installation procedure installs OnSNMP, which provides the programs listed in the following table.

Program	Description
onsrvapd daemon	When you start an Informix database server or coserver that is on this workstation, **onsrvapd** detects this event and starts OnSNMP for the database server or coserver. When the database server or coserver halts, **onsrvapd** stops OnSNMP for that database server or coserver.
runsnmp.ksh script	This script starts **onsrvapd**. It also starts the EMANATE master agent if this workstation is on a platform that EMANATE supports. If you install Informix Enterprise Command Center, the IECC installation program configures the workstation to automatically run **runsnmp.ksh** each time that you reboot. If you do not install IECC and if you want to run OnSNMP, you need to run **runsnmp.ksh** each time that you reboot.

Under normal circumstances, you do not need to start or stop OnSNMP explicitly. If you are experiencing abnormal circumstances and need to start or stop OnSNMP explicitly, contact the Informix Technical Support Department. In North America, call toll-free (800) 274-8184 or send a fax to (913) 599-8590. Outside North America, contact your distributor or the nearest Informix subsidiary.

Tip: *The following section, "UNIX Server Discovery Process," includes instructions on how to configure OnSNMP.*

UNIX

UNIX Server Discovery Process

The principles for starting and stopping **onsrvapd** are the same as those for a master agent: start **onsrvapd** before you start an Informix database server or coserver, and stop all Informix database servers and coservers on a workstation before you stop **onsrvapd**. As with a master agent, the best way to start and stop **onsrvapd** is to let the workstation start and stop it during the startup and shutdown procedures.

The rest of this section provides procedures for working with **onsrvapd**. Some of these procedures include instructions on how to configure OnSNMP.

Preparing onsrvapd

To prepare **onsrvapd**, perform the following steps:

1. Make sure that the owner of **onsrvapd** is root and that the group is **informix**.
2. Make sure that the setuid (sticky) bit is set for the **onsrvapd** file.

Starting onsrvapd Automatically at Startup

To start **onsrvapd** automatically at startup, add the following command to the startup procedure:

```
onsrvapd
```

The following table describes the command-line options that you can include. Some of these options affect OnSNMP.

Option	Description
-d	Flag that tells UNIX to run **onsrvapd** once and terminate it instead of starting it as a daemon
-g *logginglevel*	Logging level to which OnSNMP logs debug information. The default value is 32. The **onsrvapd** daemon passes this value to OnSNMP.
-k *lingermnts*	Number of minutes that **onsrvapd** waits after a database server or coserver goes down before **onsrvapd** kills the corresponding OnSNMP. If *lingermnts* is 0, **onsrvapd** waits indefinitely.
-l *pathname*	Directory for the error log files. The OnSNMP error log filename is **onsnmp.*servername*.log**. For example, if your server name is MyServer, the OnSNMP error log filename is **onsnmp.MyServer.log**. The **onsrvapd** error log filename is **onsrvapd.log**.

(1 of 2)

Option	Description
-p *pollsecs*	Frequency, in seconds, with which OnSNMP polls the database server or coserver. The default value is 5 seconds. The **onsrvapd** daemon passes this value to OnSNMP.
-r *level*	Refresh control value. For a description, see "Refresh Control Value" on page 2-21.
-V	Print the OnSNMP version number.

(2 of 2)

Starting onsrvapd Manually

To start **onsrvapd** manually, perform the following steps:

1. Stop or kill any daemons that are running on the workstation.
2. Enter the following command:

   ```
   onsrvapd
   ```

   The following table describes the command-line options that you can include. Some of these options affect OnSNMP.

Option	Description
-d	Flag that tells UNIX to run **onsrvapd** once and terminate it instead of starting it as a daemon
-g *logginglevel*	Logging level to which OnSNMP logs debug information. The default value is 32. The **onsrvapd** daemon passes this value to OnSNMP.
-k *lingermnts*	Number of minutes that **onsrvapd** waits after a database server or coserver goes down before **onsrvapd** kills the corresponding OnSNMP. If *lingermnts* is 0, **onsrvapd** waits indefinitely.
-l *pathname*	Directory for the error log files. The OnSNMP error log filename is **onsnmp.*servername*.log**. For example, if your server name is MyServer, the OnSNMP error log filename is **onsnmp.MyServer.log**. The **onsrvapd** error log filename is **onsrvapd.log**.

(1 of 2)

Option	Description
-p *pollsecs*	Frequency, in seconds, with which OnSNMP polls the database server or coserver. The default value is 5 seconds. The **onsrvapd** daemon passes this value to OnSNMP.
-r *level*	Refresh control value. For a description, see "Refresh Control Value" on page 2-21.
-V	Print the OnSNMP version number.

(2 of 2)

Making Sure That onsrvapd Is Running Correctly

To make sure that **onsrvapd** is running correctly, perform the following steps:

1. Check the log file to make sure that **onsrvapd** has not generated any errors. For the location of the log file, see the Release Notes.
2. Verify that **onsrvapd** is running.

Stopping onsrvapd Manually

To stop **onsrvapd** manually, kill the **onsrvapd** process.

Refresh Control Value

As a background task, OnSNMP periodically updates the contents of MIB tables that it derives from catalog information. The refresh control value determines the amount of time that OnSNMP spends refreshing these MIB tables versus the amount of time that it spends responding to queries from the master agent. To set the refresh control value, use the **-r** command-line option for the **onsrvapd** command. The following table lists the MIB tables that this value affects.

Database-Related MIB Tables	Table-Related MIB Tables
rdbmsDbInfoTable	**onActiveTable**
rdbmsDbTable	**onFragmentTable**
rdbmsRelTable	**onTableTable**
onBarTable	
onDatabaseTable	

The following table describes the possible values for the refresh control value.

Value	Description
`a` or `all`	Refresh the database-related and table-related tables periodically.
`n` or `none`	Do not fill or refresh any of the catalog-based tables. Instead, leave the catalog-based tables empty.
`o` or `once`	Fill the database-related and table-related tables once at startup.

The following table lists the default refresh control value for each platform.

Platform	Default Refresh Control Value
Windows NT	`all`
UNIX	`once`

The best value to use depends on the environment and how you use OnSNMP. If the list of tables and databases changes frequently, it is probably best to use a value of `all` to make sure that the MIB tables are accurate. If the environment includes many tables and databases, it is probably best to use a value of `once` to let OnSNMP respond to queries.

WIN NT

Windows NT Master Agent

The Microsoft TCP/IP installation procedure installs the Microsoft SNMP Extendible master agent. For information about this master agent, see the Microsoft TCP/IP Help.

To start the Microsoft TCP/IP Help

1. Choose **Help** from the Windows NT **Start** menu.

   This command starts the Windows NT Help.
2. Choose the **Index** tab.
3. Enter the following phrase in search box:

   `SNMP`

   In response to this search request, the help system displays a **Topics Found** dialog box.
4. Choose **TCP/IP Procedures Help**.

Important: *To start or stop the Microsoft SNMP Extendible master agent, you must be a member of the Administrator Group on the host workstation.*

WIN NT

Windows NT Subagent

The following table describes the files that comprise OnSNMP in Windows NT environments. The table also lists the directories into which the Informix installation procedure installs each file.

File	Description	Directory
infxsnmp.dll	Library that provides the interface between **onsnmp.exe** and the master agent. The Informix installation procedure installs one **infxsnmp.dll** on each workstation. The initialization process for the master agent loads **infxsnmp.dll**. The library starts and stops **onsrvapd.exe**.	**%WINNT%\system32**
onsnmp.exe	Subagent program. The Informix installation procedure installs an **onsnmp.exe** file for each database server or coserver.	**%INFORMIXDIR%\bin**
onsrvapd.exe	Server discovery process, which starts **onsnmp.exe** for each database server or coserver that starts. The Informix installation procedure performs the following tasks for **onsrvapd.exe**: ■ Installs one **onsrvapd.exe** on each workstation. ■ Configures **onsrvapd.exe** to start automatically at system initialization. The **infxsnmp.dll** library starts and stops **onsrvapd.exe**.	**%WINNT%\system32**

When you install an Informix database server or coserver, the installation procedure automatically installs OnSNMP. When you start an Informix database server or coserver that is on a network that uses SNMP, **onsrvapd.exe** detects this event and starts OnSNMP for the database server or coserver. When the database server or coserver halts, **onsrvapd.exe** stops OnSNMP for that database server or coserver.

Under normal circumstances, you do not need to start or stop OnSNMP explicitly. If you are experiencing abnormal circumstances and need to start or stop OnSNMP explicitly, contact the Informix Technical Support Department. In North America, call toll-free (800) 274-8184 or send a fax to (913) 599-8590. Outside North America, contact your distributor or the nearest Informix subsidiary.

Configuring OnSNMP

The Informix installation procedure creates a new registry key, **OnSnmpSubagent**, under **HKEY_LOCAL_MACHINE\SOFTWARE\Informix**. The following table describes the **OnSnmpSubagent** arguments that you should not change.

Argument	Value	Description
Pathname	*pathname*	Complete path of **infxsnmp.dll**, including filename
MIBS\APPLMIB	*apploid*	OID for the Application MIB
MIBS\ONMIB	*onoid*	OID for the Online MIB
MIBS\RDBMSMIB	*rdbmsoid*	OID for the RDBMS MIB
MIBS\XPSMIB	*xpsoid*	OID for the XPS MIB

AD/XP

The following table describes the **OnSnmpSubagent** arguments that you can change.

Argument	Value	Description
Environment\ LINGER_TIME	*lingermnts*	Number of minutes that the master agent waits after a database server or coserver goes down before the master agent kills the corresponding OnSNMP. If *lingermnts* is 0, the master agent waits indefinitely.
Environment\ LOGDIR	*pathname*	Complete path of the OnSNMP error log file, including filename
Environment\ REFRESH_TIME	*pollsecs*	Frequency, in seconds, with which OnSNMP polls the database server or coserver

The Informix installation procedure also creates a new argument, **INFXSNMP**, under **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents**. This new argument specifies the location of the **OnSnmpSubagent** registry key, including the name of the key.

To change the OnSNMP configuration, change the values for these arguments.

Installing the Microsoft SNMP Extendible Master Agent

If you install the Microsoft SNMP Extendible master agent after you install the Informix database server or coserver, the Informix installation procedure cannot create **INFXSNMP**. To correct this problem, run a program called **inssnmp** to complete the OnSNMP installation.

To run inssnmp

1. Start a Command Prompt session.
2. Go to **%INFORMIXDIR%\bin**.
3. Enter the following command:

   ```
   inssnmp
   ```

AD/XP

SNMP and Dynamic Server with AD and XP Options

For Dynamic Server with AD and XP Options, the way that OnSNMP runs depends on whether or not it is running on a connection coserver.

On a connection coserver, OnSNMP is the XPS subagent. The XPS subagent responds to SNMP requests for information about the entire Dynamic Server with AD and XP Options database server, which means that it provides XPS MIB information. This instance of OnSNMP also responds to SNMP requests for information about the connection coserver, which means that it provides information from the Application, RDBMS, and Online MIBs. The **onXpsTable** MIB table specifies which OnSNMP instance is the XPS subagent.

On a coserver that is not a connection coserver, OnSNMP responds to SNMP requests for information about the coserver, which means that it provides information from the Application, RDBMS, and Online MIBs.

GLS

GLS and SNMP

Informix products provide GLS, which lets you work with languages that use code sets other than the standard English code set. However, the SNMP protocols that OnSNMP supports (SNMPv1 and SNMPv2) do not recognize these different code sets.

OnSNMP 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 with the following message:

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

OnSNMP sends only 7-bit characters. If an eighth bit is present, OnSNMP truncates it. Thus, when an SNMP network manager requests character information, OnSNMP return a value. However, the value might not reflect the name of the database or table.

OnSNMP sends numeric information correctly, regardless of the code set that the database uses.

MIBs

OnSNMP uses the following MIBs:

- Application MIB
- ★ RDBMS MIB
- Online MIB in the Informix Private MIB
- XPS MIB in the Informix Private MIB

Application MIB

The Application MIB is a public MIB, which means that the Internet Engineering Task Force (IETF) specifies the structure of the MIB and the MIB tables. A public MIB is the same for all managed components on an SNMP network, not just Informix products. OnSNMP uses only **applTable**, which is the portion of the Application MIB that the RDBMS MIB requires. Figure 1-6 on page 1-10 shows the position of the Application MIB in the MIB hierarchy.

The following value is the path to the Application MIB:

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

The following value is the OID for the Application MIB:

```
.1.3.6.1.2.1.27
```

RDBMS MIB

The RDBMS MIB is a public MIB, which means that the IETF specifies the structure of the MIB and the MIB tables. A public MIB is the same for all managed database components. However, some of the definitions in the RDBMS MIB are purposely vague to let each vendor tailor the entries to a specific database server. For example, **rdbmsSrvLimitedResourceTable** contains information about the resources that a database server uses. Each database server vendor can decide the resources to include in this table. Figure 1-6 on page 1-10 shows the position of the RDBMS MIB in the MIB hierarchy.

The following value is the path to the RDBMS MIB:

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

The following value is the OID for the RDBMS MIB:

```
.1.3.6.1.2.1.39
```

Informix Private MIB

The Informix Private MIB is a private MIB, which means that a private enterprise defines and uses it. The Internet Assigned Numbers Authority (IANA) assigns a unique enterprise identifier to each company that uses the SNMP protocol. The Informix Private MIB describes information that is relevant to the specific architecture and features of Informix database servers, coservers, and databases. Figure 2-4 shows the MIB hierarchy for the Informix Private MIB.

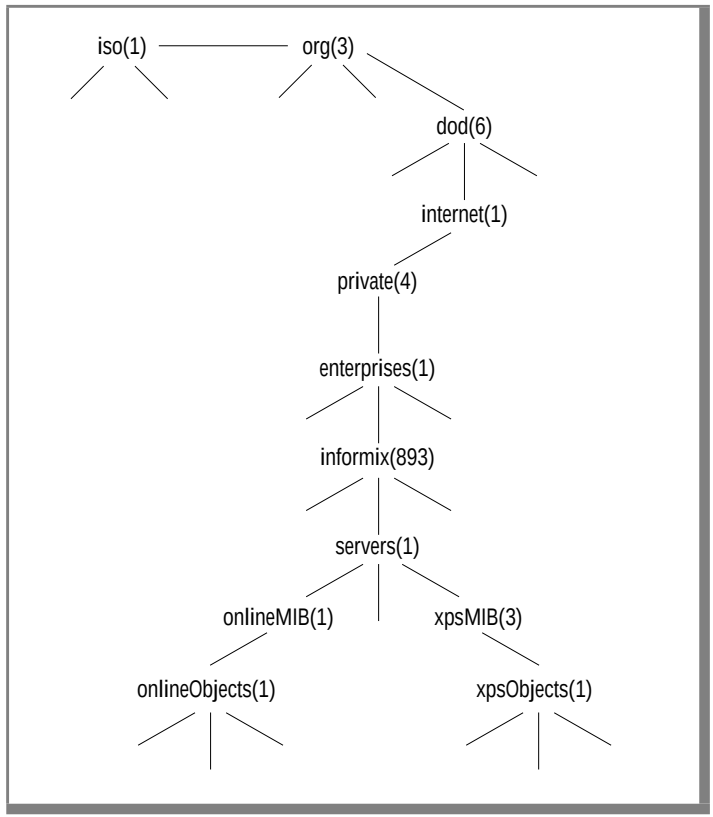


Figure 2-4
MIB Hierarchy for the Informix Private MIB

The following value is the path to the Informix Private MIB:

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

The following value is the OID for the Informix Private MIB:

```
.1.3.6.1.4.1.893
```

Online MIB

The Online MIB is in the Informix Private MIB. The Online MIB contains information for all Informix database servers and coservers. In the Online MIB, all tables are below the following node:

```
servers.onlineMIB.onlineObjects
```

The OID for each table in the Online MIB starts with the following value:

```
.1.3.6.1.4.1.893.1.1.1
```

AD/XP

XPS MIB

The XPS MIB is in the Informix Private MIB. The XPS MIB contains information for Dynamic Server with AD and XP Options database servers and coservers. In the XPS MIB, all tables are below the following node:

```
servers.xpsMIB.xpsObjects
```

The OID for each table in the XPS MIB starts with the following value:

```
.1.3.6.1.4.1.893.1.3.1
```

Table Indexing

In the OnSNMP implementation of the MIBs, the header for each table specifies how each row in the table is indexed. A table can have one, two, or more indexes. For example, the header for **rdbmsSrvTable** is **rdbmsSrvTable[applIndex]**, which means that the table has one index called **applIndex**.

Each index value is concatenated to the column OID with periods between each value. If a MIB table has several indexes, the indexes are concatenated one after the other. Most SNMP network managers display only the final portion of the OID that relates to the table being displayed. Some SNMP network managers display the OID as part of the information about each individual item; other SNMP network managers display the OID as part of a header for a list of values.

Numeric Index Values

The following line is an example of indexed information:

```
rdbmsRelActiveTime.72000003.893072000 = 01/16/98 12:34:08
```

The following table describes how to interpret the example. For more information about these values, see "rdbmsRelTable" on page 3-10.

Index Subvalue	Description
rdbmsRelActiveTime	Name of the column
72000003	**rdbmsDbIndex**
893072000	**applIndex**

Alphabetical Index Values

When an index is an alphabetical string, such as the name of a configuration parameter, the OID for that index consists of the following elements, all separated by periods:

- Number of letters in the name
- ASCII value for each letter

The following line is an example of alphabetical indexed information:

```
rdbmsSrvParamCurrValue.893072000.4.76.82.85.83.1 = 8
```

The following table describes how to interpret this example. For more information about these values, see "rdbmsSrvParamTable" on page 3-14.

Index Subvalue	Description
rdbmsSrvParamCurrValue	Name of the column
893072000	**applIndex**
4.76.82.85.83	**rdbmsSrvParamName**: 4 = Number of letters 76 = L 82 = R 85 = U 83 = S
1	**rdbmsSrvParamSubIndex**

Chapter

3

MIB Reference

This chapter describes the MIBs that OnSNMP uses. An SNMP network manager hides most of the structure of the MIBs. However, an understanding of this structure can help you understand the information that an SNMP network manager displays.

The descriptions in this chapter are brief. For detailed descriptions, see the on-line MIB files. The following table lists the directories for the MIB files on each platform.

Platform	MIB Directory
UNIX	**$INFORMIXDIR/snmp**
Windows NT	**%INFORMIXDIR%\etc**

Many MIB values are for database servers or coservers, depending on the types of servers that you are using.

AD/XP

Only Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options provides coservers. ♦

This chapter presents the MIB tables in alphabetical order. For the logical order, see MIB files. The following table summarizes the MIB tables that OnSNMP uses.

MIB	Table	Description
Application	**applTable**	Attributes for each database server or coserver
RDBMS	**rdbmsDbInfoTable**	Information about databases
	rdbmsDbTable	Information about databases
	rdbmsRelTable	Information about the relationship between a database and the database server or coserver with which it is associated
	rdbmsSrvInfoTable	Information about the database server or coserver since the server was started
	rdbmsSrvLimite-dResourceTable	Information about the limited resources for each database server or coserver
	rdbmsSrvParamTable	Information about the configuration parameters for each database server or coserver
	rdbmsSrvTable	Information about a database server or coserver
	rdbmsTraps	Information about the traps that OnSNMP can send to the SNMP network manager
Online	**onActiveBarTable**	Information about the current ON-Bar activity
	onActiveTableTable	Information about the open and active database tables
	onBarTable	Information about the backup and restore history
	onChunkTable	Information about the chunks that the database servers and coservers use
	onDatabaseTable	Information about active databases
	onDbspaceTable	Information about dbspaces
	onFragmentTable	Information about the fragments that are in fragmented database tables

(1 of 2)

MIB	Table	Description
	onLockTable	Information about the active locks that database servers and coservers are using
	onLogicalLogTable	Information about logical logs
	onPhysicalLogTable	Information about physical logs
	onServerTable	Status and profile information about each active database server and coserver
	onSessionTable	Information about each session
	onSqlHostTable	Copy of the connection information
	onTableTable	Information about a database table
	onXpsTable	Indicates the OnSNMP instance that is the XPS subagent
XPS	**xpsCogroupMemberTable**	Information about coservers as members of cogroups
	xpsCogroupTable	Information about cogroups as members of Dynamic Server with AD and XP Options database servers
	xpsCoserverTable	Information about coservers as members of Dynamic Server with AD and XP Options database servers
	xpsDbsliceDbspaceTable	Information about each dbspace that is in a Dynamic Server with AD and XP Options database server
	xpsDbsliceTable	Information about each dbslice that is in a Dynamic Server with AD and XP Options coserver

(2 of 2)

Application MIB

Informix uses one table from the application MIB. This table provides general-purpose attributes for each database server or coserver.

applTable

The following list summarizes this table:

Contents: Attributes for each database server or coserver

Index: **applIndex**

Scope of a row: One database server or coserver

The table has the following columns.

Column	Description
applIndex	Unique integer index that identifies each database server or coserver. This value is the sum of the following values: ■ Informix Enterprise ID * 1,000,000 The Informix Enterprise ID is 893. Therefore, Informix Enterprise ID * 1,000,000 is 893,000,000. ■ SERVERNUM * 1000 ■ Coserver ID for a Dynamic Server with AD and XP Options database server or 0 for any other type of database server
applName	Name of the database server or coserver
applDirectoryName	OnSNMP does not support this column.
applVersion	Version of the database server or coserver
applUptime	Time when the database server or coserver was last initialized. This time is the system time according to the master agent. If the database server or coserver was last initialized before OnSNMP was last initialized, this value is 0.

(1 of 2)

Column	Description
applOperStatus	Operating status of the database server or coserver: ■ up (1) ■ down (2) ■ halted (3) ■ -- (4): OnSNMP does not use this value. ■ restarting (5)
applLastChange	Time when the database server or coserver entered its current state. This time is the system time according to the master agent. If the database server or coserver was last initialized before OnSNMP was last initialized, this value is 0.
applInboundAssociations	Number of current SQLCONNECT actions
applOutboundAssociations	OnSNMP does not support this column.
applAccumulatedInboundAssociations	Number of SQLCONNECT actions that have occurred so far
applAccumulatedOutboundAssociations	OnSNMP does not support this column.
applLastInboundActivity	Time for the most recent attempt to start or stop a session with a database server or coserver. This time is the system time according to the master agent.
applLastOutboundActivity	OnSNMP does not support this column.
applRejectedInboundAssociations	Number of times that the database server or coserver rejected an input connection due to administrative reasons or resource limitations
applFailedOutboundAssociations	OnSNMP does not support this column.

(2 of 2)

RDBMS MIB

The RDBMS MIB defines several tables that provide information about managed database servers or coservers and their databases.

rdbmsDbInfoTable

The following list summarizes this table:

Contents: Information about databases

Index: **rdbmsDbIndex**

Scope of a row: One database that does not have an access state of **unavailable**

(The **rdbmsRelState** value indicates the access state for the database.)

The table has the following columns.

Column	Description
rdbmsDbIndex	See "rdbmsDbTable" on page 3-9.
rdbmsDbInfoProductName	Name of the database product. For example, this value might be Dynamic Server.
rdbmsDbInfoVersion	Version number of the database server or coserver that created or last restructured this database
rdbmsDbInfoSizeUnits	Units for **rdbmsDbInfoSizeAllocated** and **rdbmsDbInfoSizeUsed**: ■ bytes (1) ■ kbytes (2) ■ mbytes (3) ■ gbytes (4) ■ tbytes (5)

(1 of 2)

Column	Description
rdbmsDbInfoSizeAllocated	Estimated size allocated for this database in the units that **rdbmsDbInfoSizeUnits** specifies
rdbmsDbInfoSizeUsed	Estimated size in use for this database in the units that **rdbmsDbInfoSizeUnits** specifies
rdbmsDbInfoLastBackup	Date and time when the latest backup of the database was performed. If the database has never been backed up, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).

(2 of 2)

rdbmsDbLimitedResourcesTable

OnSNMP does not support this table.

rdbmsDbParamTable

OnSNMP does not support this table.

rdbmsDbTable

The following list summarizes this table:

Contents: Information about databases

Index: **rdbmsDbIndex**

Scope of a row: One database

The table has the following columns.

Column	Description
rdbmsDbIndex	Unique integer index that identifies a database. This value is the sum of the following values: ■ SERVERNUM * 1,000,000 If SERVERNUM is 0, OnSNMP uses 256 instead of 0. ■ Database number
rdbmsDbPrivateMibOID	OID for the Informix Private MIB: .1.3.6.1.4.1.893
rdbmsDbVendorName	Name of the database vendor: Informix Software, Inc.
rdbmsDbName	Name of the database
rdbmsDbContact	Login name of the person who created the database

rdbmsRelTable

The following list summarizes this table:

Contents: Information about the relationship between a database and the database server or coserver with which it is associated

Index: **rdbmsDbIndex, applIndex**

Scope of a row: One database

AD/XP

For Dynamic Server with AD and XP Options, a row covers one database coserver pair. ♦

The table has the following columns.

Column	Description
rdbmsDbIndex	See "rdbmsDbTable" on page 3-9.
applIndex	See "applTable" on page 3-6.
rdbmsRelState	Access state between the database server or coserver and the database: ■ other (1): The database server or coserver is on-line, but one of the dbspaces of the database is down. ■ active (2): The database server or coserver is actively using the database. The database server or coserver is on-line, and a user opened the database. ■ available (3): The database server or coserver could use the database if asked to do so. The database server or coserver is on-line, but the database is not open. ■ restricted (4): The database is not completely available. The database server or coserver is on-line, and a user opened the database in exclusive mode. ■ unavailable (5)
rdbmsRelActiveTime	Date and time that the database server or coserver made the database active. If **rdbmsRelState** is not active, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).

rdbmsSrvInfoTable

The following list summarizes this table:

Contents: Information about the database server or coserver since the server was started

Index: **applIndex**

Scope of a row: One database server or coserver

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
rdbmsSrvInfoStartupTime	Date and time when the database server or coserver was last started
rdbmsSrvInfoFinishedTransactions	Number of transactions completed, either with a commit or with an abort
rdbmsSrvInfoDiskReads	Number of reads from the physical disk
rdbmsSrvInfoLogicalReads	Number of logical reads
rdbmsSrvInfoDiskWrites	Number of writes to the physical disk
rdbmsSrvInfoLogicalWrites	Number of logical writes
rdbmsSrvInfoPageReads	Number of page reads
rdbmsSrvInfoPageWrites	Number of page writes
rdbmsSrvInfoDiskOutOfSpaces	Number of times that the database server or coserver has been unable to obtain disk space that it wanted
rdbmsSrvInfoHandledRequests	Number of requests made to the database server or coserver on inbound associations
rdbmsSrvInfoRequestRecvs	Number of receive operations that the database server or coserver made while it was processing requests on inbound associations
rdbmsSrvInfoRequestSends	Number of send operations that the database server or coserver made while it was processing requests on inbound associations
rdbmsSrvInfoHighwaterInboundAssociations	Greatest number of inbound associations that have been open at the same time
rdbmsSrvInfoMaxInboundAssociations	Greatest number of inbound associations that can be open at the same time

rdbmsSrvLimitedResourceTable

The following list summarizes this table:

Contents: Information about the limited resources for each database server or coserver

Index: **applIndex, rdbmsSrvLimitedResourceName**

Scope of a row: One limited resource

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
rdbmsSrvLimitedResourceName	Name of the limited resource: ■ BUFFERS ■ DS_MAX_QUERIES ■ DS_MAX_SCANS ■ DS_TOTAL_MEMORY ■ LOCKS ■ LTXEHWM ■ LTXHWM ■ STACKSIZE ■ LOGFILES ■ DBSPACES ■ CHUNKS
rdbmsSrvLimitedResourceID	OID or vendor name for the Informix Private MIB: .1.3.6.1.4.1.893 or informix
rdbmsSrvLimitedResourceLimit	Maximum value that this limited resource can attain
rdbmsSrvLimitedResourceCurrent	Current value for this limited resource

(1 of 2)

Column	Description
rdbmsSrvLimitedResourceHighwater	Maximum value that this limited resource has attained since **applUptime** was reset. This value is 0 for DBSPACES and CHUNKS.
rdbmsSrvLimitedResourceFailures	Number of times that the database server or coserver tried to exceed the maximum value for this limited resource since **applUptime** was reset. This value is 0 for DBSPACES and CHUNKS.
rdbmsSrvLimitedResourceDescription	Description of the limited resource. This description includes the units for the value for the limited resource.

(2 of 2)

rdbmsSrvParamTable

The following list summarizes this table:

Contents: Information about the configuration parameters for each database server or coserver

Index: **applIndex, rdbmsSrvParamName, rdbmsSrvParamSubIndex**

Scope of a row: One configuration parameter that is listed in the configuration file for the database server or coserver

The **ONCONFIG** environment variable specifies the filename of the configuration file. The following table lists the location of the configuration file for each platform. For more information about the configuration file, see your *Administrator's Guide*. For more information about the **ONCONFIG** environment variable, see the *Informix Guide to SQL: Reference*.

Platform	Location of Configuration File
UNIX	**$INFORMIXDIR/etc/$ONCONFIG**
Windows NT	**%INFORMIXDIR%\etc\%ONCONFIG%**

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
rdbmsSrvParamName	Name of a configuration parameter
rdbmsSrvParamSubindex	Subindex for the configuration parameter. This value is 1 for every configuration parameter except DATASKIP, DBSPACETEMP, DBSERVERALIASES, and NETTYPE.
rdbmsSrvParamID	OID or vendor name for the Informix Private MIB: .1.3.6.1.4.1.893 or informix
rdbmsSrvParamCurrValue	Value of the configuration parameter. OnSNMP obtains this value from the configuration file. Therefore, it does not reflect dynamic changes that you might make to the configuration parameter.
rdbmsSrvParamComment	Purpose of the configuration parameter

rdbmsSrvTable

The following list summarizes this table:

Contents: Information about a database server or coserver

Index: **applIndex**

Scope of a row: One database server or coserver

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
rdbmsSrvPrivateMibOID	OID for the Informix Private MIB: `.1.3.6.1.4.1.893`
rdbmsSrvVendorName	Name of the database server vendor: `Informix Software, Inc`.
rdbmsSrvProductName	Name of the database server product. For example, this value could be Dynamic Server.
rdbmsSrvContact	Name of the database server contact: `informix`

rdbmsTraps

See the Documentation Notes.

Online MIB in the Informix Private MIB

The Online MIB defines several tables that provide information that is specifically relevant for Informix database servers, coservers, and their databases.

onActiveBarTable

The following list summarizes this table:

Contents: Information about the current ON-Bar activity

Index: **applIndex, onActiveBarIndex**

Scope of a row: One ON-Bar activity

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onActiveBarIndex	A number that OnSNMP assigns
onActiveBarActivityType	Type of activity: ■ dbspaceBackup (1) ■ dbspaceRestore (2) ■ logBackup (3) ■ logRestore (4) ■ systemBackup (5) ■ systemRestore (6)
onActiveBarActivityLevel	Level of activity: ■ completeBackup (1) ■ incrementalLevelOne (2) ■ incrementalLevelTwo (3)
onActiveBarElapsedTime	Length of time since the activity started, in hundredths of seconds

(1 of 2)

Column	Description
onActiveBarActivitySize	Total number of used pages to scan. OnSNMP updates this value as the activity progresses.
onActiveBarActivityScanned	Number of used pages that the activity has scanned so far
onActiveBarActivityCompleted	Number of scanned pages that the activity has transferred for archiving so far
onActiveBarActivityStatus	Status of the activity

(2 of 2)

onActiveTableTable

The following list summarizes this table:

Contents: Information about the open and active database tables

Index: **applIndex, rdbmsDbIndex, onTableIndex**

Scope of a row: One open and active database table

For a fragmented database table, the values in this table are summaries of the values from all the fragments of the database table. The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-9.
onTableIndex	See "onDbspaceTable" on page 3-24.
onActiveTableStatus	Status of the table: ■ not Busy (1): The table is not in use. ■ busy (2): The table is in use. ■ dirty (3): The table has been modified.

(1 of 2)

Column	Description
onActiveTableIsBeingAltered	State of the table: ■ yes (1): The table is being altered. (An index is being added or dropped, an ALTER TABLE statement is being executed, the alter page count is being updated, or pages are being altered to conform to the latest schema.) ■ no (2): The table is not being altered.
onActiveTableUsers	Number of users accessing the table
onActiveTableLockRequests	Number of lock requests
onActiveTableLockWaits	Number of lock waits
onActiveTableLockTimeouts	Number of lock timeouts
onActiveTableIsamReads	Number of reads from the database table
onActiveTableIsamWrites	Number of writes to the database table
onActiveTableBufferReads	Number of buffer reads
onActiveTableBufferWrites	Number of buffer writes

(2 of 2)

onBarTable

The following list summarizes this table:

Contents:	Information about the backup and restore history
Index:	**applIndex, onBarActivityIndex, onBarObjectIndex**
Scope of a row:	One object that participated in a backup or restore activity

For information about backup and restore, see your *Backup and Restore Guide*. The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onBarActivityIndex	Index to the history
onBarObjectIndex	Index to the object
onBarName	Name of the object
onBarType	Type of object: ■ blobspace (1) Dynamic Server with AD and XP Options does not provide blobspaces. ■ rootDbspace (2) ■ criticalDbspace (3) ■ noncriticalDbspace (4) ■ logicalLog (5)
onBarLevel	Level of the backup action: ■ completeBackup (1) ■ incrementalLevelOne(2) ■ incrementalLevelTwo (3)

(1 of 2)

Column	Description
onBarStatus	Status of the action on the object: ■ 0 = successful ■ non-zero = error number
onBarTimeStamp	Ending time stamp for the action

(2 of 2)

onChunkTable

The following list summarizes this table:

Contents: Information about the chunks that the database servers and coservers use

Index: **applIndex, onDbspaceIndex, onChunkIndex**

Scope of a row: One chunk

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onDbspaceIndex	See "rdbmsDbInfoTable" on page 3-8.
onChunkIndex	Unique integer index for this chunk. The database server or coserver generates this value.
onChunkFileName	Pathname for the chunk
onChunkFileOffset	Offset into the device, in pages
onChunkPagesAllocated	Chunk size, in pages
onChunkPagesUsed	Number of pages used

(1 of 3)

Column	Description
onChunkType	Type of chunk: ■ regularChunk (1) ■ blobChunk (2) ■ stageBlob (3)
onChunkStatus	Status of the chunk: ■ offline (1) ■ online (2) ■ recovering (3) ■ inconsistent (4) ■ dropped (5)
onChunkMirroring	Mirroring status of the chunk: ■ notMirrored (1) ■ mirrored (2) ■ newlyMirrored (3)
onChunkReads	Number of physical-read operations
onChunkPageReads	Number of page reads
onChunkWrites	Number of physical-write operations
onChunkPageWrites	Number of page writes
onChunkMirrorFileName	Pathname of the mirror chunk. If the chunk is not mirrored, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).
onChunkMirrorFileOffset	Offset of the mirror, in pages. If the chunk is not mirrored, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).

(2 of 3)

Column	Description
onChunkMirrorStatus	Mirroring status: ■ 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)

onDatabaseTable

The following list summarizes this table:

Contents: Information about active databases

Index: **applIndex, rdbmsDbIndex**

Scope of a row: One active database

This table does not provide information about an active database if one of the dbspaces for the database is down. (The **rdbmsRelState** column for each database in **rdbmsRelTable** indicates whether or not a database is active and whether or not one of its dbspaces is down.)

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-9.
onDatabaseDbspace	Default dbspace
onDatabaseCreated	Creation date and time

(1 of 2)

Column	Description
onDatabaseLogging	Logging status: ■ none (1) ■ buffered (2) ■ unbuffered (3) ■ ansi (4)
onDatabaseOpenStatus	Database status: ■ notOpen (1) ■ open (2) ■ openExclusive (3)
onDatabaseUsers	Number of users

(2 of 2)

onDbspaceTable

The following list summarizes this table:

Contents: Information about dbspaces

Index: **applIndex, onDbspaceIndex**

Scope of a row: One dbspace

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onDbspaceIndex	Unique integer index for this dbspace. The database server or coserver generates this value.
onDbspaceName	Name of the dbspace
onDbspaceOwner	Login name of the owner
onDbspaceCreated	Creation date

(1 of 3)

Column	Description
onDbspaceChunks	Number of chunks in the dbspace
onDbspaceType	Type of dbspace: ■ regularDbspace (1) ■ temporaryDbspace (2) ■ blobDbspace (3)
onDbspaceMirrorStatus	Mirroring status: ■ notMirrored (1) ■ mirrored (2) ■ mirrorDisabled (3) ■ newlyMirrored (4)
onDbspaceRecoveryStatus	Recovery status: ■ noRecoveryNeeded (1) ■ doneRecovery (2) ■ physicallyRecovered (3) ■ logicallyRecovering (4)
onDbspaceBackupStatus	Backup status: ■ yes (1): The dbspace is backed up. ■ no (2): The dbspace is not backed up.
onDbspaceMiscStatus	Miscellaneous status: ■ none (1): no more information ■ aTableDropped (2)
onDbspacePagesAllocated	Size of all the primary chunks in the dbspace
onDbspacePagesUsed	Number of pages used in all the primary chunks in the dbspace

(2 of 3)

Column	Description
onDbspaceBackupDate	Date when the latest backup was performed. If the dbspace has never been backed up, this value is noSuchInstance (SNMPv2) or noSuchName (SNMPv1).
onDbspaceLastBackupLevel	Level of the last backup. If the dbspace has never been backed up, this value is noSuchInstance (SNMPv2) or noSuchName (SNMPv1).
onDbspaceLastFullBackupDate	Date and time of the last full backup (level 0). If the dbspace has never has a full backup, this value is noSuchInstance (SNMPv2) or noSuchName (SNMPv1).

(3 of 3)

onFragmentTable

The following list summarizes this table:

Contents: Information about the fragments that are in fragmented database tables

Index: **applIndex, rdbmsDbIndex, onTableIndex, onFragmentIndex**

Scope of a row: One fragment of a fragmented database table

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-9.
onTableIndex	See "onDbspaceTable" on page 3-24.
onFragmentIndex	Unique integer index for the fragment
onFragmentType	Type of database table: ■ fragmentedIndex (1) ■ fragmentedTable (2)

(1 of 2)

Column	Description
onFragmentDbspace	Dbspace name for the fragment
onFragmentExpression	Expression text used for fragmentation of the table or index. This value is blank if the fragmentation scheme is round-robin.
onFragmentIndexName	Index identifier
onFragmentExtents	Number of extents used
onFragmentPagesAllocated	Total (extent) size allocated to the fragment, in pages
onFragmentPagesUsed	Number of pages used
onFragmentIsamReads	Number of reads from the fragment. If the fragment is not active, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).
onFragmentIsamWrites	Number of writes to the fragment. If the fragment is not active, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).

(2 of 2)

onLockTable

The following list summarizes this table:

Contents: Information about the active locks that database servers and coservers are using

Index: **applIndex, onSessionIndex, onLockIndex**

Scope of a row: One lock

There is a row for each lock that the session is using and for each lock on which the session is waiting.

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onSessionIndex	See "onServerTable" on page 3-31.
onLockIndex	Index to this row
onLockDatabaseName	Name of the database using or waiting for this lock
onLockTableName	Name of the table using or waiting for this lock
onLockType	Type of the lock: ■ byte (1) ■ intentShared (2) ■ shared (3) ■ sharedByRepeatableRead (4) ■ update (5) ■ intentExclusive (6) ■ sharedIntentExclusive (7) ■ exclusive (8) ■ exclusiveByRepeatableRead (9) ■ waiting (10)
onLockGranularity	Granularity of the lock: ■ table (1) ■ page (2) ■ row (3) ■ index (4)
onLockRowId	**rowid** of the locked row
onLockWaiters	Number of sessions waiting for the lock
onLockGrantTime	Time when the lock was granted if the session is using the lock. If no transaction exists, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).

onLogicalLogTable

The following list summarizes this table:

Contents: Information about logical logs

Index: **applIndex, onLogicalLogIndex**

Scope of a row: One logical log

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onLogicalLogIndex	Index for the logical-log file
onLogicalLogID	Unique integer identification number for the logical-log file
onLogicalLogDbspace	Dbspace name where the log file was created
onLogicalLogStatus	Status of the logical-log file: ■ newlyAdded (1) ■ free (2) ■ current (3) ■ used (4) ■ backedUpButNeeded (5)
onLogicalLogContainsLastCheckpoint	Checkpoint status: ■ yes (1): The logical-log file contains the last checkpoint. ■ no (2): The logical-log file does not contain the last checkpoint.
onLogicalLogIsTemporary	Temporary status: ■ yes (1): The logical-log file is temporary. ■ no (2): The logical-log file is not temporary.
onLogicalLogPagesAllocated	Size of the logical-log file, in pages
onLogicalLogPagesUsed	Number of pages used in the logical-log file

(1 of 2)

Column	Description
onLogicalLogFillTime	Date and time when the logical-log file last filled up. If the log file has never been full, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).
onLogicalLogTimeUniqueIDChanged	Time stamp when a new unique ID was assigned to this logical-log entry
onLogicalLogTimeLastBackupDate	Date and time of the last backup for this logical-log entry

(2 of 2)

onPhysicalLogTable

The following list summarizes this table:

Contents: Information about physical logs

Index: **applIndex**

Scope of a row: One physical log

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onPhysicalLogDbspace	Dbspace name where the physical log was created
onPhysicalLogBufferSize	Size of the physical-log buffer, in pages
onPhysicalLogBufferUsed	Number of pages of the physical-log buffer that are used
onPhysicalLogPageWrites	Number of pages written to the physical log
onPhysicalLogWrites	Number of (disk) writes to the physical log
onPhysicalLogPagesAllocated	Size of the physical log, in pages
onPhysicalLogPagesUsed	Number of pages used

onServerTable

The following list summarizes this table:

Contents: Status and profile information about each active database server and coserver

Index: **applIndex**

Scope of a row: One database server or coserver

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6
onServerMode	Mode of the database server or coserver: ■ initializing (1) ■ quiescent (2) ■ fastRecovery (3) ■ backingUp (4) ■ shuttingDown (5) ■ online (6) ■ aborting (7) ■ onlineReadOnly (8)
onServerCheckpointInProgress	Checkpoint status: ■ yes (1): A checkpoint is in progress. ■ no (2): A checkpoint is not in progress.
onServerPageSize	Size of a page, in bytes
onServerThreads	Number of active threads
onServerVPs	Number of virtual processors
onServerVirtualMemory	Total virtual memory used, in kilobytes
onServerResidentMemory	Total resident memory used, in kilobytes

(1 of 3)

Column	Description
onServerMessageMemory	Total message memory used, in kilobytes
onServerIsamCalls	Sum of all reads, writes, rewrites, deletes, commits, and rollbacks to and from the database table
onServerLatchWaits	Number of latch waits
onServerLockRequests	Number of lock requests
onServerLockWaits	Number of lock waits
onServerBufferWaits	Number of buffer waits
onServerCheckpointWaits	Number of checkpoint waits
onServerDeadLocks	Number of deadlocks
onServerLockTimeouts	Number of deadlock time-outs
onServerLogicalLogRecords	Number of logical-log records
onServerLogicalLogPageWrites	Number of logical-log page writes
onServerLogicalLogWrites	Number of logical-log writes
onServerBufferFlushes	Number of buffer flushes
onServerForegroundWrites	Number of foreground writes
onServerLRUWrites	Number of LRU writes
onServerChunkWrites	Number of chunk writes
onServerReadAheadPages	Number of read-ahead pages. This value includes data and index read-ahead pages.
onServerReadAheadPagesUsed	Number of read-ahead pages used
onServerSequentialScans	Number of sequential scans
onServerMemorySorts	Number of memory sorts
onServerDiskSorts	Number of disk sorts
onServerMaxSortSpace	Maximum disk space that a sort uses, in pages
onServerNetworkReads	Number of network reads

(2 of 3)

Column	Description
onServerNetworkWrites	Number of network writes
onServerPDQCalls	Number of parallel-processing actions performed
onServerTransactionCommits	Number of committed transactions
onServerTransactionRollbacks	Number of rolled-back transactions
onServerTimeSinceLastCheckpoint	Length of time since the last checkpoint, in centi-seconds
onServerCPUSystemTime	Amount of CPU time that the database server has used in System Mode, in hundredths of seconds
onServerCPUUserTime	Amount of CPU time that the database server has used in User Mode, in hundredths of seconds

(3 of 3)

onSessionTable

The following list summarizes this table:

Contents: Information about each session

Index: **applIndex, onSessionIndex**

Scope of a row: One session

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onSessionIndex	Unique integer index for the session
onSessionUserName	Name of the user, in the form `name@host(tty)`
onSessionUserProgramVersion	Version of the database server or coserver
onSessionUserProcessId	Process ID for the session

(1 of 4)

Column	Description
onSessionUserTime	Length of time that the user has been connected to the database server or coserver, in hundredths of seconds
onSessionState	State of the session: ■ idle (1) ■ active (2) ■ waitingOnMutex (3) ■ waitingOnCondition (4) ■ waitingOnLock (5) ■ waitingOnBuffer (6) ■ waitingOnCheckPointing (7) ■ waitingOnLogicalLogWrite (8) ■ waitingOnTransaction (9)
onSessionDatabase	Connected database
onSessionCurrentMemory	Memory usage, in bytes
onSessionThreads	Number of active threads
onSessionCurrentStack	Average size of the stack for all threads
onSessionHighwaterStack	Maximum amount of memory that any thread has used so far
onSessionLockRequests	Number of lock requests
onSessionLocksHeld	Number of locks held
onSessionLockWaits	Number of lock waits
onSessionLockTimeouts	Number of time-outs for locks
onSessionLogRecords	Number of log records
onSessionIsamReads	Number of reads from database tables
onSessionIsamWrites	Number of writes to database tables
onSessionPageReads	Number of page reads
onSessionPageWrites	Number of page writes

Column	Description
onSessionLongTxs	Number of long transactions
onSessionLogSpace	Logical-log space used, in bytes
onSessionHighwaterLogSpace	Maximum logical-log space that this session has ever used
onSessionSqlStatement	Latest SQL statement, truncated to 250 characters if necessary
onSessionSqlIsolation	SQL isolation level: ■ noTransactions (1) ■ dirtyReads (2) ■ readCommitted (3) ■ cursorRecordLocked (4) ■ repeatableRead (5)
onSessionSqlLockWaitMode	Action to take if the isolation level requires a wait: ■ -1 = Wait forever. ■ 0 = Do not wait. ■ >0 = Wait for specified number of seconds.
onSessionSqlEstimatedCost	Estimated cost of the SQL statement according to SQLEXPLAIN
onSessionSqlEstimatedRows	Estimated number of rows that the SQL statement will select according to SET EXPLAIN
onSessionSqlError	Error number for the last SQL statement
onSessionSqlIsamError	ISAM error number for the last SQL statement
onSessionTransactionStatus	Status of the transaction: ■ none (1) ■ committing (2) ■ rollingBack (3) ■ rollingHeuristically (4) ■ waiting (5)
onSessionTransactionBeginLog	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).

Column	Description
onSessionTransactionLastLog	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).
onSessionOriginatingCoserverId	This value applies only to Dynamic Server with AD and XP Options. It is the coserver ID of the coserver for which this local session runs.
onSessionOriginatingSessionId	Local session ID of the global session on the coserver for which this local session runs

onSqlHostTable

The following list summarizes this table:

Contents: Copy of the connection information

Index: **applIndex, onSqlHostIndex**

Scope of a row: One connectivity value

As the following table shows, the location of the connection information depends on the platform. For more information about the connection information, see your *Administrator's Guide*.

Platform	Location of Connectivity Information
UNIX	The **INFORMIXSQLHOSTS** environment variable specifies the full pathname and filename of the connection information. The default location is **$INFORMIXDIR /etc/sqlhosts**. For more information about **INFORMIXSQLHOSTS**, see the *Informix Guide to SQL: Reference*.
Windows NT	The connectivity information is in a key in the Windows registry called **HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SQLHOSTS**.

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onSqlHostIndex	Index to the entry in the connectivity information
onSqlHostName	Host name of the database server or coserver
onSqlHostNetType	Connection type
onSqlHostServerName	Name of the database server or coserver or its alias
onSqlHostServiceName	Service name
onSqlHostOptions	List server options in the form of `key=value` pairs

onTableTable

The following list summarizes this table:

Contents: Information about a database table

Index: **applIndex, rdbmsDbIndex, onTableIndex**

Scope of a row: One database table

For a fragmented database table, the values in this table are summaries of the values from all the database table fragments. The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-9.
onTableIndex	Table number. This value is the same as **tabid** in the system catalog table **systables**.
onTableName	Table name
onTableOwner	Table owner

(1 of 2)

Column	Description
onTableType	Type of table: ■ table (1) ■ view (2) ■ privateSynonyn (3) ■ synonym (4)
onTableLockLevel	Locking level of the table: ■ page (1) ■ row (2)
onTableCreated	Creation date, in string format
onTableFirstDbspace	Name of the first (or only) dbspace for the table
onTableRowSize	Length of a row
onTableRows	Number of rows
onTableColumns	Number of columns
onTableIndices	Number of indexes
onTableExtents	Number of extents in use
onTablePagesAllocated	Total (extent) size allocated to the table, in pages
onTablePagesUsed	Number of pages in use
onTableFragments	Number of fragments
onTableFragmentStrategy	Fragmentation strategy: ■ roundRobin (1) ■ byExpression (2) ■ tableBased (3) If the table is not fragmented, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).
onTableActiveFragments	Number of active fragments. If the table is not fragmented, this value is `noSuchInstance` (SNMPv2) or `noSuchName` (SNMPv1).

(2 of 2)

AD/XP

onXpsTable

The following list summarizes this table:

Contents: The OnSNMP instance that is the XPS subagent

Index: **applIndex**

Scope of a row: One coserver

For information about the XPS subagent, see "SNMP and Dynamic Server with AD and XP Options" on page 2-25. The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
onXpsSnmpIndex	Unique integer index that identifies the XPS subagent. This value is the sum of the following values: ■ Informix Enterprise ID * 1,000,000 The Informix Enterprise ID is 893. Therefore, Informix Enterprise ID * 1,000,000 is 893,000,000. ■ SERVERNUM * 1000 ■ Coserver ID
onXpsSnmpHost	Name of the workstation on which the XPS subagent runs

AD/XP

XPS MIB in the Informix Private MIB

The XPS MIB defines several tables that provide information that is specifically relevant for Dynamic Server with AD and XP Options database servers, coservers, and their databases.

xpsCogroupMemberTable

The following list summarizes this table:

Contents: Information about coservers as members of cogroups

Index: **applIndex, xpsCogroupIndex, xpsCogroupMemberIndex**

Scope of a row: One coserver

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
xpsCogroupIndex	See "xpsCogroupTable" on page 3-41.
xpsCogroupMemberIndex	Unique integer index that identifies the coserver
xpsCogroupMemberCoserverId	Coserver ID

xpsCogroupTable

The following list summarizes this table:

Contents: Information about cogroups as members of Dynamic Server with AD and XP Options database servers

Index: **applIndex, xpsCogroupIndex**

Scope of a row: One cogroup

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
xpsCogroupIndex	Unique integer index that identifies the cogroup
xpsCogroupName	Name of the cogroup
xpsCogroupMemberCount	Number of coservers that are in the cogroup
xpsCogroupIsClustered	Indicates whether or not the cogroup is clustered: ■ yes (1) ■ no (2)

xpsCoserverTable

The following list summarizes this table:

Contents: Information about coservers as members of Dynamic Server with AD and XP Options database servers

Index: **applIndex, xpsCoserverIndex**

Scope of a row: One coserver

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
xpsCoserverIndex	Unique integer index that identifies the coserver. This value is the sum of the following values: ■ Informix Enterprise ID * 1,000,000 The Informix Enterprise ID is 893. Therefore, Informix Enterprise ID * 1,000,000 is 893,000,000. ■ SERVERNUM * 1000 ■ Coserver ID
xpsCoserverHostname	Name of the workstation on which the coserver runs
xpsCoserverId	Coserver ID
xpsCoserverRootDbspaceId	ID of the root dbspace for the coserver

xpsDbsliceDbspaceTable

The following list summarizes this table:

Contents: Information about each dbspace that is in a Dynamic Server with AD and XP Options database server

Index: **applIndex, xpsDbsliceIndex, xpsDbsliceDbspaceIndex**

Scope of a row: One dbspace

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
xpsDbsliceIndex	See "xpsDbsliceTable" on page 3-44.
xpsDbsliceDbspaceIndex	Unique integer index that identifies the dbspace
xpsDbsliceDbspace	Coserver ID for the connecting coserver

xpsDbsliceTable

The following list summarizes this table:

Contents: Information about each dbslice that is in a Dynamic Server with AD and XP Options database server

Index: **applIndex, xpsDbsliceIndex**

Scope of a row: One dbslice

The table has the following columns.

Column	Description
applIndex	See "applTable" on page 3-6.
xpsDbsliceIndex	Unique integer index that identifies the dbslice
xpsDbsliceName	Name of the dbslice
xpsDbsliceDbspacesCount	Number of dbspaces in the dbslice
xpsDbsliceIsMirrored	Indicates whether or not the dbslice is mirrored: ■ yes (1) ■ no (2)
xpsDbsliceIsBlobSlice	Reserved
xpsDbsliceIsTemp	Indicates whether or not the dbslice is temporary: ■ yes (1) ■ no (2)

Glossary

ASN.1 Acronym for Abstract Syntax Notation One. Standard language for describing data structures. ASN.1 does not depend on computer architecture or implementation language. Therefore, it lets programs exchange structured data over networks. For more information, see the documents listed in "Related Reading" on page 11 of the Introduction.

infxsnmp.dll Library for the Windows NT version of OnSNMP.

managed component Hardware or software component that has a corresponding SNMP subagent. To manage a component, an SNMP network manager communicates with a master agent, and the master agent communicates with the subagent.

managed workstation Workstation that includes a managed component.

master agent Program that provides an interface between a subagent and an SNMP network manager. A managed workstation includes one master agent.

MIB Acronym for Management Information Base. Group of tables that contain the information that a subagent can provide to a master agent.

OID Acronym for Object Identifier. Numerical value that identifies a MIB, a MIB table, a parameter (column) in a MIB table, or an object (row) in a MIB table.

onsnmp Subagent program for the UNIX version of OnSNMP.

onsnmp.exe Subagent program for the Windows NT version of OnSNMP.

OnSNMP	Subagent for an Informix database server or coserver. On UNIX, OnSNMP consists of **onsnmp**. In Windows NT environments, OnSNMP consists of **onsnmp.exe**, **infxsnmp.dll**, and **onsrvapd.exe**.
onsrvapd.exe	Server discovery program for the Windows NT version of OnSNMP.
SNMP	Acronym for Simple Network Management Protocol. Communication protocol that lets you manage components on a network.
SNMP network manager	Program that monitors and controls managed components on a network.
SNMPv1, SNMPv2	Versions of the SNMP standard. For more information, see the documents listed in "Related Reading" on page 11 in the Introduction.
subagent	Program that lets a managed component and a master agent communicate with each other.

Index

R

S

T

X