DB2® IBM Informix

Version 10.0/8.5

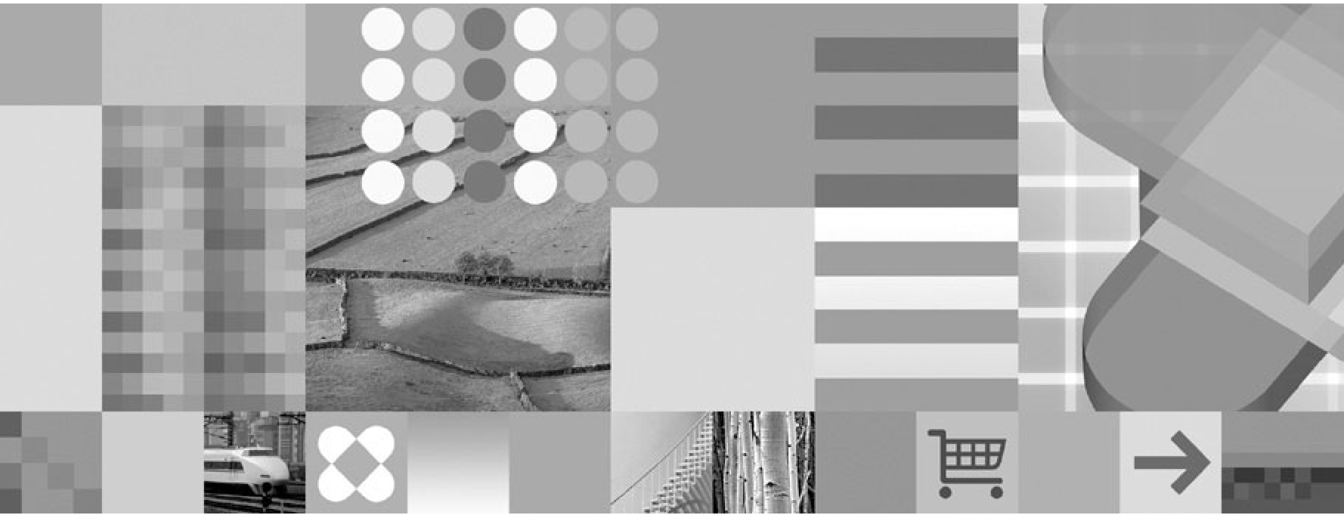
IBM Informix SNMP Subagent Guide

G251-2298-00

DB2® IBM Informix

IBM

Version 10.0/8.5



IBM Informix SNMP Subagent Guide

G251-2298-00

Note!

Before using this information and the product it supports, read the information in “Notices” on page C-1.

First Edition (December 2004)

Contents

Introduction

In This Introduction

This introduction provides an overview of the information in this manual and describes the conventions it uses.

About This Manual

This manual describes the Simple Network Management Protocol (SNMP) and the software that you need to use SNMP to monitor and manage Informix database servers, coservers, and databases.

Types of Users

This manual is written 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

Software Dependencies

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

- IBM Informix Extended Parallel Server, Version 8.5
- IBM Informix Dynamic Server, Version 10.0

You must install additional software to use the IBM Informix implementation of SNMP. For specific requirements, see Chapter 2, "IBM Informix Implementation of SNMP," on page 2-1.

Assumptions About Your Locale

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

This manual assumes that you use the U.S. 8859-1 English locale as the default locale. The default is **en_us.8859-1** (ISO 8859-1) on UNIX platforms or **en_us.CP1252** (Microsoft **1252**) for Windows environments. This locale supports U.S. English format conventions for dates, times, and currency, and also supports the ISO 8859-1 or Microsoft **1252** 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 *IBM Informix: GLS User's Guide*.

Important: SNMPv1 and SNMPv2 do not recognize non-English code sets. For more information, see "GLS and SNMP" on page 2-18.

Demonstration Databases

The DB–Access utility, which is provided with your Informix database server products, includes one or more of the following demonstration databases:

- The **stores_demo** database illustrates a relational schema with information about a fictitious wholesale sporting-goods distributor. Many examples in IBM Informix manuals are based on the **stores_demo** database.

Extended Parallel Server

- The **sales_demo** database illustrates a dimensional schema for data-warehousing applications. For conceptual information about dimensional data modeling, see the *IBM Informix: Database Design and Implementation Guide*.

End of Extended Parallel Server

Dynamic Server

- The **superstores_demo** database illustrates an object-relational schema. The **superstores_demo** database includes examples of extended data types, type and table inheritance, and user-defined routines.

End of Dynamic Server

For information about how to create and populate the demonstration databases, see the *IBM Informix: DB–Access User's Guide*. For descriptions of the databases and their contents, see the *IBM 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 in the **%INFORMIXDIR%\bin** directory in Windows environments.

New Features

For a comprehensive list of new features for your database server, see the *IBM Informix: Getting Started Guide*.

Documentation Conventions

This section describes the conventions that this manual uses. These conventions make it easier to gather information from this and other volumes in the documentation set.

The following conventions are discussed:

- Typographical conventions
- Other conventions
- Syntax diagrams

- Command-line conventions
- Example code conventions

Typographical Conventions

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

Convention	**Meaning**
KEYWORD	All primary elements in a programming language statement (keywords) appear in uppercase letters in a serif font.
italics ***italics*** *`italics`*	Within text, new terms and emphasized words appear in italics. Within syntax and code examples, variable values that you are to specify appear in italics.
boldface ***boldface***	Names of program entities (such as classes, events, and tables), environment variables, file and pathnames, and interface elements (such as icons, menu items, and buttons) appear in boldface.
`monospace` *`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 a menu item. For example, "Choose **Tools > Options**" means choose the **Options** item from the **Tools** menu.

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

Feature, Product, and Platform

Feature, product, and platform markup identifies paragraphs that contain feature-specific, product-specific, or platform-specific information. Some examples of this markup follow:

Dynamic Server

Identifies information that is specific to IBM Informix Dynamic Server

End of Dynamic Server

Extended Parallel Server

Identifies information that is specific to IBM Informix Extended Parallel Server

End of Extended Parallel Server

UNIX Only

Identifies information that is specific to UNIX platforms

End of UNIX Only

Windows Only

Identifies information that is specific to the Windows environment

End of Windows Only

This markup can apply to one or more paragraphs within a section. When an entire section applies to a particular product or platform, this is noted as part of the heading text, for example:

Table Sorting (Linux Only)

Syntax Diagrams

This guide uses syntax diagrams built with the following components to describe the syntax for statements and all commands other than system-level commands.

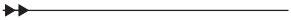
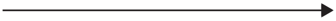
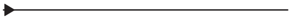
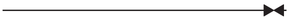
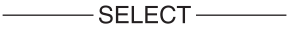
Note: Starting in 2004, syntax diagrams have been reformatted to conform to the IBM standard.

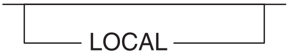
Syntax diagrams depicting SQL and command-line statements have changed in the following ways:

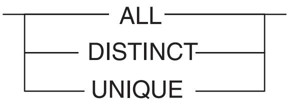
- The symbols at the beginning and end of statements are now double arrows instead of a vertical line at the end.
- The symbols at the beginning and end of syntax segment diagrams are now vertical lines instead of arrows.

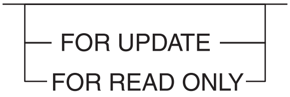
- How many times a loop can be repeated is now explained in a diagram footnote instead of a number in a gate symbol.
- Syntax statements that are longer than one line now continue on the next line instead of looping down with a continuous line.
- Product or condition-specific paths are now explained in diagram footnotes instead of icons.

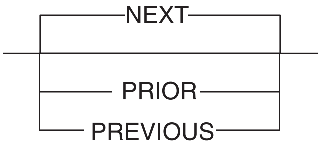
The following table describes syntax diagram components.

Component represented in PDF	Component represented in HTML	Meaning
	`>>----------------------`	Statement begins.
	`---------------------->`	Statement continues on next line.
	`>-----------------------`	Statement continues from previous line.
	`----------------------><`	Statement ends.
	`--------SELECT----------`	Required item.

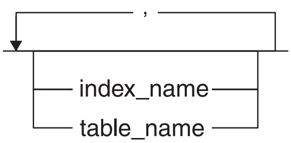
	`--+-----------------+---`
	`   '------LOCAL------'`
		Optional item.

	`---+-----ALL-------+---`
	`   +--DISTINCT-----+`
	`   '---UNIQUE------'`
		Required item with choice. One and only one item must be present.

	`---+-----------------+---`
	`    +--FOR UPDATE-----+`
	`    '--FOR READ ONLY--'`
		Optional items with choice are shown below the main line, one of which you might specify.

	`    .---NEXT---------.`
	`----+----------------+---`
	`    +---PRIOR--------+`
	`    '---PREVIOUS-----'`
		The values below the main line are optional, one of which you might specify. If you do not specify an item, the value above the line will be used as the default.

Component represented in PDF	Component represented in HTML	Meaning

	`.-------,-----------.`
	`V                   |`
	`---+-----------------+---`
	`   +---index_name---+`
	`   '---table_name---'`
		Optional items. Several items are allowed; a comma must precede each repetition.

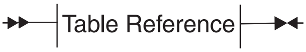
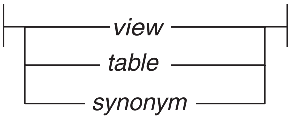
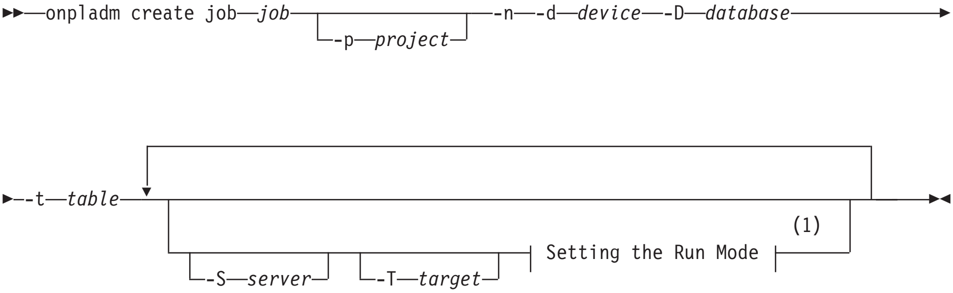
	`>>-| Table Reference |-><`	Reference to a syntax segment.

Table Reference

	`Table Reference`
	`|--+-----view--------+--|`
	`   +------table------+`
	`   '----synonym------'`
		Syntax segment.

How to Read a Command-Line Syntax Diagram

The following command-line syntax diagram uses some of the elements listed in the table in the previous section.

Creating a No-Conversion Job

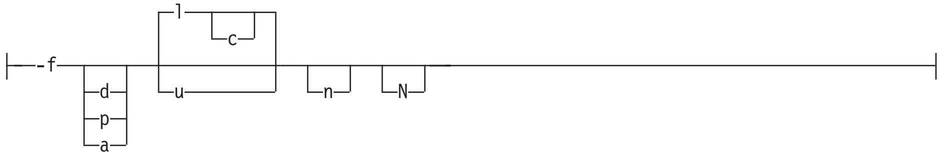


Notes:

1 See page 17-4

The second line in this diagram has a segment named "Setting the Run Mode," which according to the diagram footnote, is on page 17-4. This segment is shown in the following segment diagram (the diagram uses segment start and end components).

Setting the Run Mode:



To construct a command correctly, start at the top left with the command. Follow the diagram to the right, including the elements that you want. The elements in the diagram are case sensitive.

The Creating a No-Conversion Job diagram illustrates the following steps:

1. Type **onpladm create job** and then the name of the job.
2. Optionally, type **-p** and then the name of the project.
3. Type the following required elements:
   - **-n**
   - **-d** and the name of the device
   - **-D** and the name of the database
   - **-t** and the name of the table
4. Optionally, you can choose one or more of the following elements and repeat them an arbitrary number of times:
   - **-S** and the server name
   - **-T** and the target server name
   - The run mode. To set the run mode, follow the Setting the Run Mode segment diagram to type **-f**, optionally type **d**, **p**, or **a**, and then optionally type **l** or **u**.
5. Follow the diagram to the terminator.

Your diagram is complete.

Keywords and Punctuation

Keywords are words reserved for statements and all commands except system-level commands. When a keyword appears in a syntax diagram, it is shown in uppercase letters. When you use a keyword in a command, you can write it in uppercase or lowercase letters, but you must spell the keyword exactly as it appears in the syntax diagram.

You must also use any punctuation in your statements and commands exactly as shown in the syntax diagrams.

Identifiers and Names

Variables serve as placeholders for identifiers and names in the syntax diagrams and examples. You can replace a variable with an arbitrary name, identifier, or literal, depending on the context. Variables are also used to represent complex syntax elements that are expanded in additional syntax diagrams. When a variable appears in a syntax diagram, an example, or text, it is shown in *lowercase italic*.

The following syntax diagram uses variables to illustrate the general form of a simple SELECT statement.

```
►►──SELECT──column_name──FROM──table_name──────────────────────────────────►◄
```

When you write a SELECT statement of this form, you replace the variables *column_name* and *table_name* with the name of a specific column and table.

Example Code Conventions

Examples of SQL code occur throughout this manual. Except as noted, the code is not specific to any single IBM Informix application development tool.

If only SQL statements are listed in the example, they are not delimited by semicolons. For instance, you might see the code in the following example:

```
CONNECT TO stores_demo
...

DELETE FROM customer
   WHERE customer_num = 121
...

COMMIT WORK
DISCONNECT CURRENT
```

To use this SQL code for a specific product, you must apply the syntax rules for that product. For example, if you are using DB–Access, you must delimit multiple statements with semicolons. If you are using an SQL API, you must use EXEC SQL at the start of each statement and a semicolon (or other appropriate delimiter) at the end of the statement.

Tip: Ellipsis points in a code example indicate that more code would be added in a full application, but it is not necessary to show it to describe the concept being discussed.

For detailed directions on using SQL statements for a particular application development tool or SQL API, see the manual for your product.

Additional Documentation

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

- Installation guides
- Online notes
- Informix error messages
- Manuals
- Online help

Installation Guides

Installation guides are located in the **/doc** directory of the product CD or in the **/doc** directory of the product‘s compressed file if you downloaded it from the IBM Web site. Alternatively, you can obtain installation guides from the IBM Informix Online Documentation site at http://www.ibm.com/software/data/informix/pubs/library/.

Online Notes

The following sections describe the online files that supplement the information in this manual. Please examine these files before you begin using your IBM Informix product. They contain vital information about application and performance issues.

Online File	Description	Format
TOC Notes	The TOC (Table of Contents) notes file provides a comprehensive directory of hyperlinks to the release notes, the fixed and known defects file, and all the documentation notes files for individual manual titles.	HTML
Documentation Notes	The documentation notes file for each manual contains important information and corrections that supplement the information in the manual or information that was modified since publication.	HTML, text
Release Notes	The release notes file describes feature differences from earlier versions of IBM Informix products and how these differences might affect current products. For some products, this file also contains information about any known problems and their workarounds.	HTML, text
Machine Notes	(Non-Windows platforms only) The machine notes file describes any platform-specific actions that you must take to configure and use IBM Informix products on your computer.	text
Fixed and Known Defects File	This text file lists issues that have been identified with the current version. It also lists customer-reported defects that have been fixed in both the current version and in previous versions.	text

Locating Online Notes

Online notes are available from the IBM Informix Online Documentation site at http://www.ibm.com/software/data/informix/pubs/library/. Additionally you can locate these files before or after installation as described below.

Before Installation

All online notes are located in the **/doc** directory of the product CD. The easiest way to access the documentation notes, the release notes, and the fixed and known defects file is through the hyperlinks from the TOC notes file.

The machine notes file and the fixed and known defects file are only provided in text format.

After Installation

On UNIX platforms in the default locale, the documentation notes, release notes, and machine notes files appear under the **$INFORMIXDIR/release/en_us/0333** directory.

Dynamic Server

On Windows the documentation and release notes files appear in the **Informix** folder. To display this folder, choose **Start > Programs > IBM Informix Dynamic Server** *version* **> Documentation Notes** or **Release Notes** from the taskbar.

Machine notes do not apply to Windows platforms.

End of Dynamic Server

Online Notes Filenames

Online notes have the following file formats:

Online File	File Format	Examples
TOC Notes	*prod_os*_tocnotes_*version*.html	**ids_win_tocnotes_10.0.html**
Documentation Notes	*prod_bookname*_docnotes_*version*.html/txt	**ids_hpl_docnotes_10.0.html**
Release Notes	*prod_os*_relnotes_*version*.html/txt	**ids_unix_relnotes_10.0.txt**
Machine Notes	*prod*_machine_notes_*version*.txt	**ids_machine_notes_10.0.txt**
Fixed and Known Defects File	*prod*_defects_*version*.txt	**ids_defects_10.0.txt** **client_defects_2.90.txt**
	ids_win_fixed_and_known _defects_*version*.txt	**ids_win_fixed_and_known _defects_10.0.txt**

Informix Error Messages

This file is a comprehensive index of error messages and their corrective actions for the Informix products and version numbers.

On UNIX platforms, use the **finderr** command to read the error messages and their corrective actions.

Dynamic Server

On Windows, use the Informix Error Messages utility to read error messages and their corrective actions. To display this utility, choose **Start > Programs > IBM Informix Dynamic Server** *version* **> Informix Error Messages** from the taskbar.

End of Dynamic Server

You can also access these files from the IBM Informix Online Documentation site at http://www.ibm.com/software/data/informix/pubs/library/.

Manuals

Online Manuals

A CD that contains your manuals in electronic format is provided with your IBM Informix products. You can install the documentation or access it directly from the CD. For information about how to install, read, and print online manuals, see the installation insert that accompanies your CD. You can also obtain the same online manuals from the IBM Informix Online Documentation site at http://www.ibm.com/software/data/informix/pubs/library/.

Printed Manuals

To order hardcopy manuals, contact your sales representative or visit the IBM Publications Center Web site at http://www.ibm.com/software/howtobuy/data.html.

Online Help

IBM Informix online help, provided with each graphical user interface (GUI), displays information about those interfaces and the functions that they perform. Use the help facilities that each GUI provides to display the online help.

Accessibility

IBM is committed to making our documentation accessible to persons with disabilities. Our books are available in HTML format so that they can be accessed with assistive technology such as screen reader software. The syntax diagrams in our manuals are available in dotted decimal format, which is an accessible format that is available only if you are using a screen reader. For more information about the dotted decimal format, see the Accessibility appendix.

IBM Informix Dynamic Server Version 10.0 and CSDK Version 2.90 Documentation Set

The following tables list the manuals that are part of the IBM Informix Dynamic Server, Version 10.0 and the CSDK Version 2.90, documentation set. PDF and HTML versions of these manuals are available at http://www.ibm.com/software/data/informix/pubs/library/. You can order hardcopy versions of these manuals from the IBM Publications Center at http://www.ibm.com/software/howtobuy/data.html.

Table 1. Database Server Manuals

Manual	**Subject**
Administrator's Guide	Understanding, configuring, and administering your database server.
Administrator's Reference	Reference material for Informix Dynamic Server, such as the syntax of database server utilities **onmode** and **onstat**, and descriptions of configuration parameters, the **sysmasters** tables, and logical-log records.
Backup and Restore Guide	The concepts and methods you need to understand when you use the **ON-Bar** and **ontape** utilities to back up and restore data.
DB-Access User's Guide	Using the **DB-Access** utility to access, modify, and retrieve data from Informix databases.
DataBlade API Function Reference	The DataBlade API functions and the subset of ESQL/C functions that the DataBlade API supports. You can use the DataBlade API to develop client LIBMI applications and C user-defined routines that access data in Informix databases.
DataBlade API Programmer's Guide	The DataBlade API, which is the C-language application-programming interface provided with Dynamic Server. You use the DataBlade API to develop client and server applications that access data stored in Informix databases.
Database Design and Implementation Guide	Designing, implementing, and managing your Informix databases.
Enterprise Replication Guide	How to design, implement, and manage an Enterprise Replication system to replicate data between multiple database servers.
Error Messages file	Causes and solutions for numbered error messages you might receive when you work with IBM Informix products.
Getting Started Guide	Describes the products bundled with IBM Informix Dynamic Server and interoperability with other IBM products. Summarizes important features of Dynamic Server and the new features for each version.
Guide to SQL: Reference	Information about Informix databases, data types, system catalog tables, environment variables, and the stores_demo demonstration database.
Guide to SQL: Syntax	Detailed descriptions of the syntax for all Informix SQL and SPL statements.
Guide to SQL: Tutorial	A tutorial on SQL, as implemented by Informix products, that describes the basic ideas and terms that are used when you work with a relational database.
High-Performance Loader User's Guide	Accessing and using the High-Performance Loader (HPL), to load and unload large quantities of data to and from Informix databases.
Installation Guide for Microsoft Windows	Instructions for installing IBM Informix Dynamic Server on Windows.
Installation Guide for UNIX and Linux	Instructions for installing IBM Informix Dynamic Server on UNIX and Linux.

Table 1. Database Server Manuals (continued)

Manual	Subject
J/Foundation Developer's Guide	Writing user-defined routines (UDRs) in the Java programming language for Informix Dynamic Server with J/Foundation.
Large Object Locator DataBlade Module User's Guide	Using the Large Object Locator, a foundation DataBlade module that can be used by other modules that create or store large-object data. The Large Object Locator enables you to create a single consistent interface to large objects and extends the concept of large objects to include data stored outside the database.
Migration Guide	Conversion to and reversion from the latest versions of Informix database servers. Migration between different Informix database servers.
Optical Subsystem Guide	The Optical Subsystem, a utility that supports the storage of BYTE and TEXT data on optical disk.
Performance Guide	Configuring and operating IBM Informix Dynamic Server to achieve optimum performance.
R-Tree Index User's Guide	Creating R-tree indexes on appropriate data types, creating new operator classes that use the R-tree access method, and managing databases that use the R-tree secondary access method.
SNMP Subagent Guide	The IBM Informix subagent that allows a Simple Network Management Protocol (SNMP) network manager to monitor the status of Informix servers.
Storage Manager Administrator's Guide	Informix Storage Manager (ISM), which manages storage devices and media for your Informix database server.
Trusted Facility Guide	The secure-auditing capabilities of Dynamic Server, including the creation and maintenance of audit logs.
User-Defined Routines and Data Types Developer's Guide	How to define new data types and enable user-defined routines (UDRs) to extend IBM Informix Dynamic Server.
Virtual-Index Interface Programmer's Guide	Creating a secondary access method (index) with the Virtual-Index Interface (VII) to extend the built-in indexing schemes of IBM Informix Dynamic Server. Typically used with a DataBlade module.
Virtual-Table Interface Programmer's Guide	Creating a primary access method with the Virtual-Table Interface (VTI) so that users have a single SQL interface to Informix tables and to data that does not conform to the storage scheme of Informix Dynamic Server.

Table 2. Client/Connectivity Manuals

Manual	Subject
Client Products Installation Guide	Installing IBM Informix Client Software Developer's Kit (Client SDK) and IBM Informix Connect on computers that use UNIX, Linux, and Windows.
Embedded SQLJ User's Guide	Using IBM Informix Embedded SQLJ to embed SQL statements in Java programs.

Table 2. Client/Connectivity Manuals (continued)

Manual	Subject
ESQL/C Programmer's Manual	The IBM Informix implementation of embedded SQL for C.
GLS User's Guide	The Global Language Support (GLS) feature, which allows IBM Informix APIs and database servers to handle different languages, cultural conventions, and code sets.
JDBC Driver Programmer's Guide	Installing and using Informix JDBC Driver to connect to an Informix database from within a Java application or applet.
.NET Provider Reference Guide	Using Informix .NET Provider to enable .NET client applications to access and manipulate data in Informix databases.
ODBC Driver Programmer's Manual	Using the Informix ODBC Driver API to access an Informix database and interact with the Informix database server.
OLE DB Provider Programmer's Guide	Installing and configuring Informix OLE DB Provider to enable client applications, such as ActiveX Data Object (ADO) applications and Web pages, to access data on an Informix server.
Object Interface for C++ Programmer's Guide	The architecture of the C++ object interface and a complete class reference.

Table 3. DataBlade Developer's Kit Manuals

Manual	Subject
DataBlade Developer's Kit User's Guide	Developing and packaging DataBlade modules using BladeSmith and BladePack.
DataBlade Module Development Overview	Basic orientation for developing DataBlade modules. Includes an example illustrating the development of a DataBlade module.
DataBlade Module Installation and Registration Guide	Installing DataBlade modules and using BladeManager to manage DataBlade modules in Informix databases.

Compliance with Industry Standards

The American National Standards Institute (ANSI) and the International Organization of Standardization (ISO) have jointly established a set of industry standards for the Structured Query Language (SQL). IBM Informix SQL-based products are fully compliant with SQL-92 Entry Level (published as ANSI X3.135-1992), which is identical to ISO 9075:1992. In addition, many features of IBM Informix database servers comply with the SQL-92 Intermediate and Full Level and X/Open SQL Common Applications Environment (CAE) standards.

IBM Welcomes Your Comments

We want to know about any corrections or clarifications that you would find useful in our manuals, which will help us improve future versions. Include the following information:

- The name and version of the manual that you are using
- Section and page number
- Your suggestions about the manual

Send your comments to us at the following email address:

docinf@us.ibm.com

This email address is reserved for reporting errors and omissions in our documentation. For immediate help with a technical problem, contact IBM Technical Support.

We appreciate your suggestions.

Chapter 1. SNMP Concepts

In This Chapter

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. For SNMP terms and definitions, see the Glossary.

Purpose of SNMP

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

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

SNMP Architecture

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

- SNMP Network Managers
- Master agents
- Subagents
- Managed components

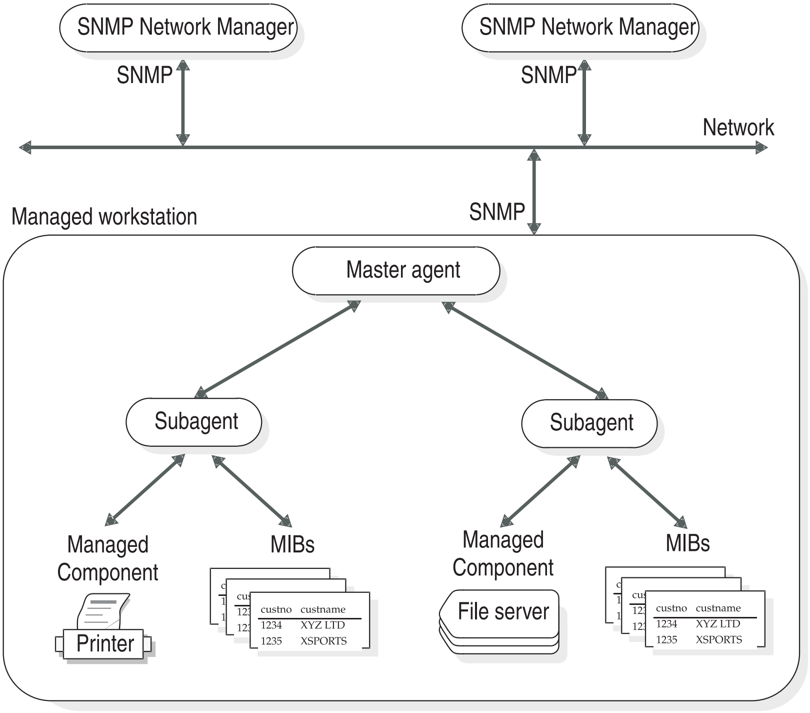


Figure 1-1. SNMP Architecture

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 hardware and network services have created SNMP Network Managers. For example:

- 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 allows the SNMP Network Manager to 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 a graphical user interface (GUI) such as the one that Figure 1-2 illustrates. With this SNMP Network Manager, you select a node to monitor and then choose specific information from a menu.

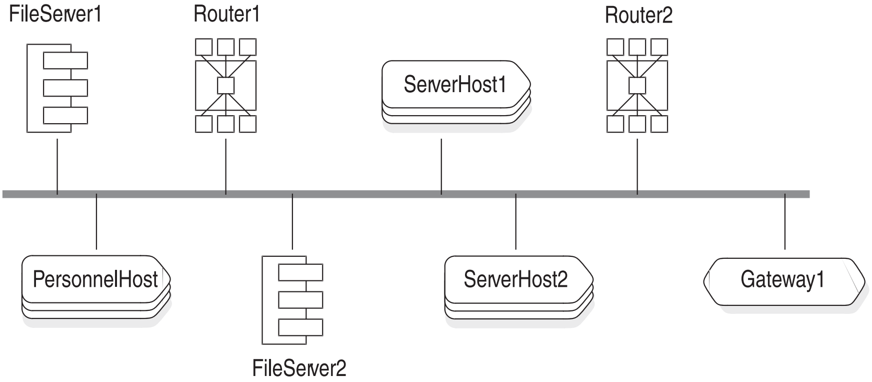


Figure 1-2. SNMP Network Manager Example

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

```
Feb 17 1999 [ smoke ] : RDBMS-MIB.rdbmsDbTable
KEY = 72000003
rdbmsDbName = CustomerData
rdbmsDbName.72000003 = AnotherData
rdbmsDbPrivateMibOID = 1.3.6.1.4.1.893
rdbmsDbVendorName = IBM Corporation
rdbmsDbName = CustomerData
rdbmsDbContact = John Doe
```

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 = IBM Corporation
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.

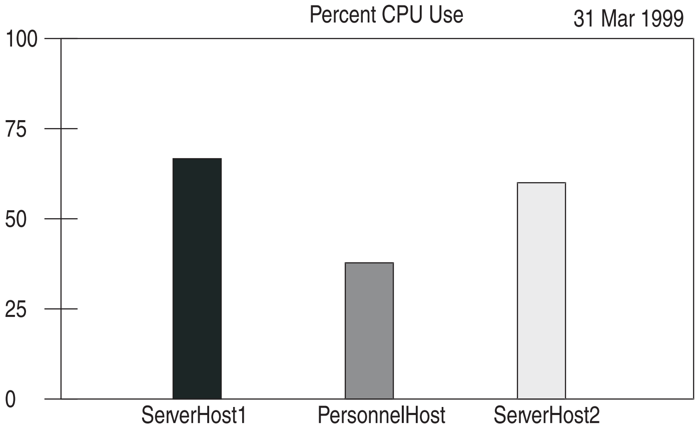


Figure 1-5. Example of Monitoring Information

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

Event Notification

When an event occurs that affects the performance or availability of a managed component, the SNMP Network Manager 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 Network Manager 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 the SNMP Network Manager 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 Network Manager notifies you of a trap.

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

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

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.

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 an 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). Figure 1-6 shows the MIB hierarchy.

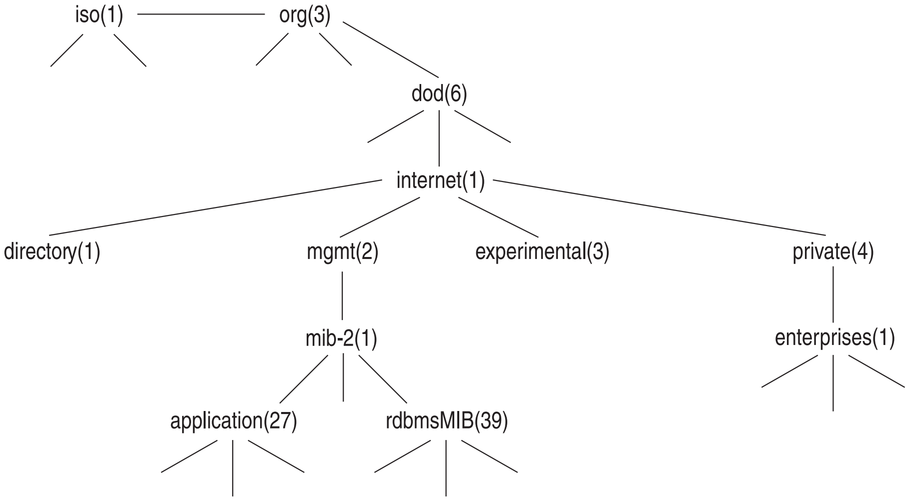


Figure 1-6. MIB Hierarchy

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. It uniquely describes each piece of data that an SNMP Network Manager can obtain and 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. IBM Informix Implementation of SNMP

In This Chapter

This chapter describes how SNMP manages Informix database servers. The IBM Informix implementation consists of the following components:

- Master agent

UNIX Only

– On UNIX, a master agent is provided through licensing agreements with vendors. See "UNIX Master Agents" on page 2-8.

End of UNIX Only

Windows 2000 Only

– On Windows, install the Microsoft SNMP Extendible master agent.

End of Windows 2000 Only

- Subagent

  The subagent for Informix database servers and coservers is OnSNMP.
- Managed components

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

  OnSNMP uses several MIBs.

Extended Parallel Server

Only Extended Parallel Server provides coservers.

End of Extended Parallel Server

Purpose of IBM Informix SNMP

The IBM Informix implementation of SNMP lets database administrators monitor 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 object in an MIB table. The following table describes four possible events and the MIB objects that correspond to them.

Event	MIB Object
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, you might discover that an application that uses an Informix database server stopped responding. You 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, even if you use a database that is on a local host, you can call a remote technical support representative to report a problem. The problem might be that the data 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 through 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 Manually" on page 2-9.

Information That OnSNMP Provides

All the information that OnSNMP provides is available from other sources, such as the system catalog tables, the **sysmaster** and **sysutils** databases, dbaccess calls, 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 the Informix databases that are on a network. Figure 2-1 illustrates this concept.

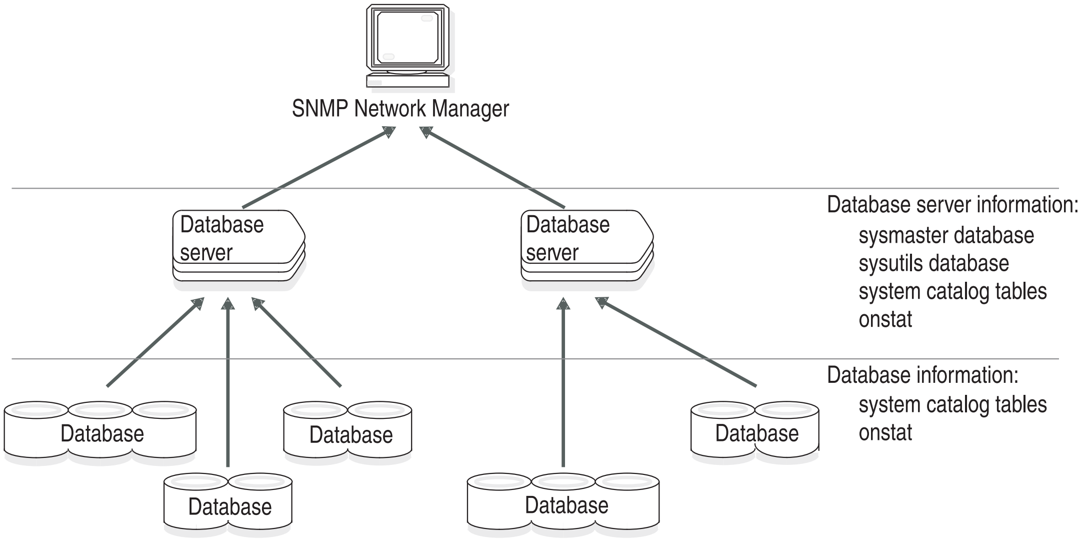


Figure 2-1. Monitoring Informix Databases

SNMP Standard

The SNMP standard has two versions: SNMPv1 and SNMPv2. The following table lists the versions of the SNMP standard with which OnSNMP complies.

Operating System	Version of the SNMP Standard
UNIX	SNMPv1 and SNMPv2
Windows	SNMPv1

SNMP Architecture

The architecture for the IBM Informix implementation of SNMP depends on your operating system.

IBM Informix SNMP Architecture on UNIX (UNIX)

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

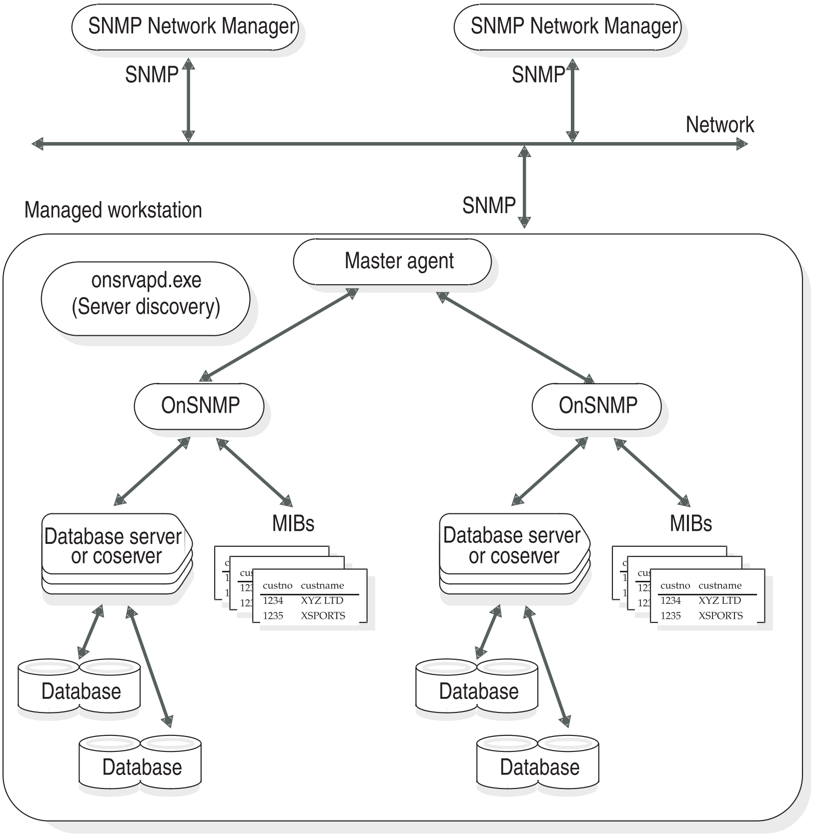


Figure 2-2. IBM Informix SNMP Architecture on UNIX

IBM Informix SNMP Architecture on Windows (Windows)

Figure 2-3 on page 2-6 shows the SNMP architecture for Informix database servers on Windows. 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**. One instance of the **onsnmp** subagent is started for each instance of Dynamic Server that runs on the managed workstation. OnSNMP and the master agent do not need to use SNMP to communicate with each other.

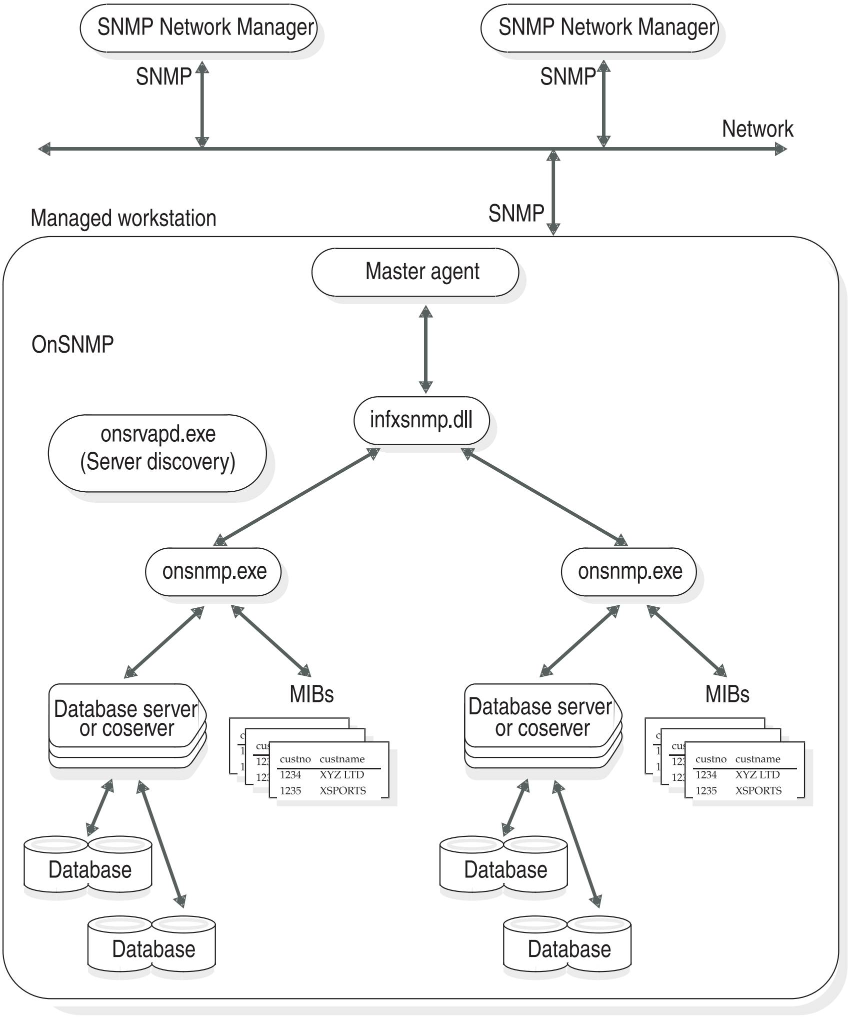


Figure 2-3. IBM Informix SNMP Architecture on Windows

Using SNMP on UNIX

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

- **runsnmp.ksh**
- 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 as well as the files needed for SNMP support.

  The discovery process discovers multiple server instances running on the host. These instances might belong to different versions that are installed on different directories. Whenever a server instance is brought online, the discovery process detects it and spawns an instance of OnSNMP to monitor the database server.

For information on how **runsnmp.ksh** automatically sets up and starts SNMP on UNIX, see "The runsnmp.ksh Script" on page 2-7. The rest of this section describes how to do a manual setup and provides background information that you can use to diagnose setup problems.

The runsnmp.ksh Script

The **runsnmp.ksh** script on UNIX ensures that both the SNMP master agent and the **onsrvapd** server-discovery daemon are running on a host. The **runsnmp.ksh** file is in the **$INFORMIXDIR/snmp** directory. You must correctly set the **INFORMIXDIR** environment variable to the latest installed version of the product and run the script as **root**.

```
►►──runsnmp.ksh──┬───────────────────────┬──┬──────┬──┬───────┬──────────────►◄
                 ├─-m──master_agent_args─┤  └─stop─┘  └─start─┘
                 └─-s──server_disc_args──┘
```

Issue the **runsnmp.ksh** commands that the following diagram shows.

Option	**Description**
-m *master_agent_args*	The master-agent arguments can be either **stop** or valid master-agent arguments.
-s *server_disc_args*	The server-discovery arguments can be either **stop** or valid **onsrvapd** arguments.
start	Starts **snmpdm** or **snmpdp** and **onsrvapd** if they are not running. This option is the default.
stop	Stops **snmpdm** or **snmpdp** and **onsrvapd** if they are already running and exits.

The *master_agent_args* and the *server_disc_args* are not checked for correctness.

The following examples illustrate how to use **runsnmp.ksh**:

- Start **snmpdm** or **snmpdp** and **onsrvapd** if they are not running.

  ```
  runsnmp.ksh
  ```

- Stop **onsrvapd** and **subagents** and then exit.

```
runsnmp.ksh -s stop
```

- Stop **onsrvapd** and any **subagents** and then restart **onsrvapd**.

```
runsnmp.ksh -s stop start
```

- Stop **snmpdm** or **snmpdp**, **onsrvapd**, and any **subagents** and then exit.

```
runsnmp.ksh stop
```

- Stop **snmpdm** or **snmpdp**, **onsrvapd**, and any **subagents** and then restart **snmpdm** or **snmpdp** and **onsrvapd**.

```
runsnmp.ksh stop start
```

- Start **snmpdm** or **snmpdp**, if it is not running, and then start **onsrvapd** with the none option, if it is not running.

```
runsnmp.ksh -s "-rnone"
```

UNIX Master Agents

On UNIX, master agents are provided through licensing agreements. The following table lists these master agents.

Master Agent	Company	Web Site Home Page
EMANATE, Version 14.2	SNMP Research	www.snmp.com
Patrol, Version 2.3	BMC Software	www.bmc.com

For some UNIX platforms, you might be able to use a master agent other than the one provided with the database server. To see whether this applies to your platform, see your release notes.

Assuring Compatibility

The following guidelines assure master agent compatibility:

- Only one master agent is provided, usually EMANATE or Patrol, for each UNIX platform type.
- The subagent that works with the master agent is also provided with the database server.
- In some cases, the platform vendor also supplies a master agent that works with the subagent provided with the database server. This is generally true only if the platform vendor supplies the same type of master agent as that provided with the database server and if the version number of the vendor-supplied master agent is greater than or equal to that of the version provided with the database server.
- You should run only one instance of a master agent on a platform. You can run multiple instances of subagents, including multiple instances of **onsnmp**, if multiple database server or coserver instances exist.
- Informix subagents can coexist with subagents that platform or third-party vendors supply if all the subagents share a common, compatible master agent.

Installing and Configuring a Master Agent Manually

The **runsnmp.ksh** script automatically performs the steps in this section for the master agents provided with the database server. 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.
   - Set **SR_LOG_DIR** to the directory for the EMANATE log file.

   The EMANATE configuration files are located in the **$INFORMIXDIR/snmp/snmpr** directory. The log files are located in the **/tmp** directory. The **/tmp** directory is the default location if the variable is not set.
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. 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.

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. 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.

   The Patrol configuration file is located in the **$INFORMIXDIR/snmp/peer** directory. For information about the Patrol configuration values, see the

BMC Software Web site, which is listed in “UNIX Master Agents” on page 2-8. The Patrol log files are located in the **/tmp** directory.

Starting and Stopping a Master Agent

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 run the **runsnmp.ksh** script as part of the startup procedure for the system. Similarly, the best way to stop a master agent is to run the **runsnmp.ksh** script as part of the shutdown procedure. However, you can start or stop a master agent manually if you prefer. 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 and Stopping a Master Agent Automatically: The **runsnmp.ksh** script automatically starts the EMANATE or Patrol master agent at startup and stops it at shutdown.

Starting and Stopping a Master Agent Manually: This section describes how to start or stop a master agent if you do not use the **runsnmp.ksh** script.

To start a master agent manually:

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:

   For EMANATE:

   ```
   snmpdm &
   ```

   For Patrol:

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

To stop a master agent manually:

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 following process:

   For EMANATE, **snmpdm**

For Patrol, **snmpdp**

The following table describes the command-line options that you can include in the `snmpdm` command for the EMANATE master agent.

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.

To make sure that a master agent is running correctly:

1. Check the master agent log file to verify that the master agent has not generated any errors. The log file is located in the **/tmp** directory unless the environment variable mentioned in on page 2-9 is set to a different directory.
2. Verify that the process is running:

   For EMANATE, **snmpdm**

   For Patrol, **snmpdp**

UNIX Subagent

When you install an Informix database server or coserver on UNIX, the installation procedure installs OnSNMP. OnSNMP consists of the **onsnmp** program.

Under normal circumstances, you do not need to start or stop OnSNMP explicitly. If you experience abnormal circumstances and need to start or stop OnSNMP explicitly, contact Technical Support. For contact information, refer to your *IBM Informix: Installation Guide*.

The following additional files are provide with the database server for SNMP support.

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. See “UNIX Server Discovery Process” on page 2-12.

runsnmp.ksh script

This script starts **onsrvapd**. It also starts the master agent that is appropriate for the platform. If you want to run OnSNMP, you need to run **runsnmp.ksh** each time that you reboot. See “The runsnmp.ksh Script” on page 2-7.

UNIX Server Discovery Process

The **runsnmp.ksh** script automatically starts the UNIX server discovery process, as “The runsnmp.ksh Script” on page 2-7 describes. This section provides procedures for working manually with **onsrvapd**. Some of these procedures include instructions on how to configure OnSNMP.

The principles for starting and stopping **onsrvapd** manually 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**.

Preparing onsrvapd Manually

If you do not use **runsnmp.ksh** to automatically prepare and start **onsrvapd**, perform the steps in this section.

To prepare onsrvapd:

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.

Issuing the onsrvapd Command

You can specify the **onsrvapd** command-line options that Figure 2-4 shows. Some of these options affect OnSNMP.

```
►►──onsrvapd──┬─────────────────────┬──────────────────────────────────────────►◄
              ├─-d──────────────────┤
              ├─-g──logginglevel────┤
              ├─-k──lingermnts──────┤
              ├─-l──pathname────────┤
              ├─-p──pollsecs────────┤
              ├─-r──server_disc_args┤
              ├─-s──level───────────┤
              └─-V──────────────────┘
```

Figure 2-4. onsrvapd Command

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. Valid values are `2`, `4`, `8`, `16`, `32`, and `64`. The default value is `32`. The lower the value, the higher the amount of logging. 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 filename of the OnSNMP error log is **onsnmp.*servername*.log**. For example, if your server name is MyServer, the filename of the OnSNMP error log is **onsnmp.MyServer.log**. The filename of the **onsrvapd** error log is **onsrvapd.log**.

-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 2-23.

-V Prints the OnSNMP version number.

To start onsrvapd manually:

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

   ```
   onsrvapd
   ```

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

To make sure that onsrvapd is running correctly:

1. Check the log file to verify that **onsrvapd** has not generated any errors. The log file is located in the **/tmp** directory.
2. Verify that **onsrvapd** is running.

Choosing an Installation Directory

When you have multiple Informix installation directories on a host computer, you must set the latest installation directory as **INFORMIXDIR** before you run the **runsnmp.ksh** script to start OnSNMP. If all the directories are for the same type of database server, use the installation directory that has the latest database server version number.

One way to determine the latest directory to use with different types of database server lines is to find the latest version of the SNMP master agent.

- The EMANATE master agent displays the version when you run it.
- The BMC Patrol agent also displays a version number.

Using SNMP on Windows

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

- Microsoft's SNMP service 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 as well as the files needed for SNMP support.

Windows Master Agent

The Microsoft TCP/IP custom 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 **Start > Help**.
2. Choose the **Index** tab.
3. Enter the following phrase in the text 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.

Windows Subagent

On Windows, OnSNMP comprises the following files. The table also lists the directories in which the IBM Informix installation procedure installs each file.

File	Description	Directory
infxsnmp.dll	Library that provides the interface between **onsnmp.exe** and the master agent. The IBM Informix installation procedure installs one **infxsnmp.dll** on each workstation. The initialization process for the master agent loads **infxsnmp.dll**.	**%Windows%\system32**
onsnmp.exe	Subagent program. The IBM 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 IBM Informix installation procedure performs the following tasks for **onsrvapd.exe**: • Installs one **onsrvapd.exe** on each workstation • Creates the Informix Server Discovery Process for SNMP in the control panel and configures it to start automatically when the system reboots	**%Windows%\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.

Starting and Stopping OnSNMP

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 Technical Support. For contact information, refer to your *IBM Informix: Installation Guide*.

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 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
Environment LOGLEVEL	*loglevel*	Logging level to which OnSNMP logs debugging information. The default value is `3`. The **onsrvapd** daemon passes this value to OnSNMP.

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 (XPS)	xpsoid	OID for the XPS MIB

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.

Windows Server Discovery Process

The Informix Server Discovery Process for SNMP is known as **onsrvapd**. It is installed as a Windows service that runs under the Informix user. The discovery process discovers multiple server instances running on the host. These instances might belong to different versions that are installed on different directories. Whenever a server instance is brought online, the discovery process detects it and spawns an instance of OnSNMP to monitor the database server.

Starting and Stopping onsrvapd

You can start **onsrvapd** from the services folder in the control panel or from a command prompt. To start and stop **onsrvapd** from a command prompt, enter the following commands:

- To start **onsrvapd**, enter:

  ```
  net start onsrvapd
  ```

- To stop **onsrvapd**, enter:

  ```
  net stop onsrvapd
  ```

The OnSNMP Discovery Process (onsrvapd.exe) is installed as an Windows service and starts and stops automatically. You do not need to issue commands at the command line. In the event you want to issue commands from the command line, see the command-line syntax listed in “Issuing the onsrvapd Command” on page 2-12.

To make sure that onsrvapd is running correctly:

1. Check the log file to verify that **onsrvapd** has not generated any errors. For location of the log files, see your release notes.
2. Verify that **onsrvapd** is running.

Installing the IBM Informix SNMP 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
   ```

Tip: If you install a Windows service pack on your computer before you install the Microsoft SNMP Extendible master agent, you might need to reinstall the service pack.

SNMP and Extended Parallel Server (XPS)

For Extended Parallel Server, the MIB objects that OnSNMP supports depend on whether or not it is running on coserver 1.

Coserver 1 supports the following tables:

- Application (all MIB objects)

- RDBMS (all MIB objects)
- Online (all MIB objects except Enterprise Replication objects)
- XPS (all MIB objects)

All other coservers support the following tables:

- Application (all MIB objects)
- RDBMS
  - **rdbmsSrvInfoTable**
  - **rdbmsSrvLimitedResourceTable**
  - **rdbmsSrvParamTable**
  - **rdbmsSrvTable**
  - **rdbmsTraps**
- Online
  - **onActiveBarTable**
  - **onChunkTable**
  - **onDbspaceTable**
  - **onLockTable**
  - **onLogicalLogTable**
  - **onPhysicalLogTable**
  - **onServerTable**
  - **onSessionTable**
  - **onSqlHostTable**
  - **onXpsTable**

These tables and MIB objects are described in Chapter 3, "MIB Reference," on page 3-1.

The **onXpsTable** MIB table specifies which OnSNMP instance is the subagent on coserver 1 for Extended Parallel Server. The subagent responds to SNMP requests for XPS MIB information about the entire database server as well as about coserver 1.

GLS and SNMP

IBM Informix products include a Global Language Support (GLS) feature, 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 returns 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

This section describes the types of MIBs and the types of MIB objects that the Informix database server uses. For a description of MIBs, see page 1-6.

OnSNMP uses the following MIBs:

- Application MIB
- Relational Database Management System (RDBMS) MIB
- Informix Private MIB
- Online MIB in the Informix Private MIB

Extended Parallel Server

- XPS MIB in the Informix Private MIB

End of Extended Parallel Server

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 for IBM Informix products. OnSNMP uses only **applTable**, which is the portion of the Application MIB that the RDBMS MIB requires. Figure 1-6 on page 1-7 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 which resources to include in this table. Figure 1-6 on page 1-7 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 2-5 shows the MIB hierarchy for the Informix Private MIB.

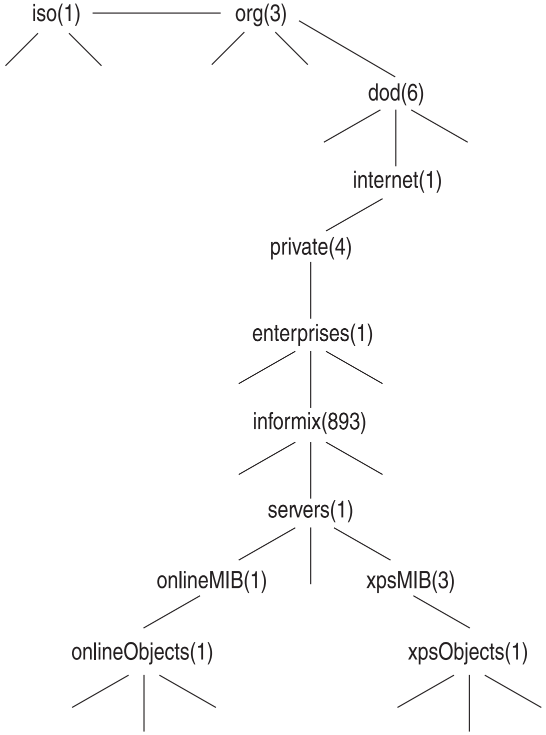


Figure 2-5. MIB Hierarchy for the Informix Private MIB

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
```

XPS MIB

The XPS MIB is in the Informix Private MIB. The XPS MIB contains information for database servers and coservers on Extended Parallel Server. 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
```

MIB Objects

An MIB object is similar to a column in a table. The IBM Informix implementation of SNMP recognizes the following types of MIB objects:

- Traps are defined as MIB objects, but they cannot be retrieved. Instead, when a certain condition is detected, OnSNMP issues an event that includes the object ID that the trap defines.
- Catalog-based MIB objects exist only if the refresh control value (described on page 2-23) is `once` or `all`.

Extended Parallel Server

- Catalog-based MIB objects exist only if the refresh control value is `once` or `all` and they are only on coserver 1.
- XPS-only MIB objects exist only on the node that hosts coserver 1 for Extended Parallel Server.

End of Extended Parallel Server

- Enterprise Replication objects are tables that exist only if a database server is configured to participate in Enterprise Replication.

Table Indexing

In the description of the MIBs in Chapter 3, the header for each table specifies how each row in the table is indexed. A table can have one 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 = 11/16/98 12:34:08
```

The following table describes how to interpret the example. For more information about these values, see “rdbmsRelTable” on page 3-7.

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 3-10.

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**

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.

Specify the refresh control value with the **runsnmp.ksh -s -r** command-line option or the **onsrvapd -r** command-line option. The following table lists the MIB tables that this value affects. See also "Issuing the onsrvapd Command" on page 2-12.

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

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 operating system.

Operating System	Default Refresh Control Value
UNIX	once
Windows	all

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.

Files Installed for SNMP

This section lists the files that are typically installed for the IBM Informix implementation of SNMP on UNIX and Windows.

Files Installed on UNIX

The **runsnmp.ksh** file exists for all UNIX versions of SNMP support.

The following files are installed in **$INFORMIXDIR/bin**.

Filename	Description
onsnmp	OnSNMP executable
onsrvapd	Server discovery process

snmpdm	EMANATE executable or a dummy file for UNIX platforms that EMANATE does not support
snmpdp	PATROL executable or a dummy file for UNIX platforms that PATROL does not support

The following files are installed in **$INFORMIXDIR/snmp**.

Filename	**Description**
Files for the MIBs:	
./snmpr/snmpd.cnf	EMANATE configuration file or a dummy file for UNIX platforms that EMANATE does not support
./peer/CONFIG	Patrol configuration file or a dummy file for UNIX platforms that Patrol does not support
.runsnmp.ksh	Script that starts the master agent and **onsrvapd**

OnSNMP uses the following log files by default.

Filename	Description
snmp.log	Log file for EMANATE; not installed on UNIX platforms that EMANATE does not support
onsrvapd.log	Log file for onsrvapd.
onsnmp.*.log For Extended Parallel Server, the path is **onsnmp.servername**.**coserver#.log** The *coserver#* is optional. For Dynamic Server, the path is **onsnmp.servername.log**	Log file for onsnmp.

Files Installed on Windows

The following files are created in **%Windows%\system32**.

Filename	**Description**
infxsnmp.dll	DLL for OnSNMP
onsrvapd.exe	Server discovery process

The following file is created in **%INFORMIXDIR%\bin**.

Filename **Description**

onsnmp.exe	OnSNMP executable

In addition, log files are created in the directories that are specified in the registry.

Chapter 3. MIB Reference

In This Chapter

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 comprehend the information that an SNMP Network Manager displays.

The descriptions in this chapter are brief. For detailed descriptions, see the online MIB files. The following table lists the directories for the MIB files.

Operating System	MIB Directory
UNIX	**$INFORMIXDIR/snmp**
Windows	**%INFORMIXDIR%\etc**

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

Extended Parallel Server

Only Extended Parallel Server provides coservers.

End of Extended Parallel Server

This chapter presents the MIB tables in alphabetical order. For the logical order, see the MIB files. The following table summarizes the MIB tables that OnSNMP uses and indicates the page that contains more information.

MIB	Table	Description
Application (See page 3-3)	**applTable**	Attributes for each database server or coserver
RDBMS (See page 3-5)	**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 it was started
	rdbmsSrvLimited-ResourceTable	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 (See page 3-12)	**onActiveBarTable**	Information about the current ON–Bar activity
	onActiveTableTable	Information about the open and active database tables

MIB	Table	Description
	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
	onErQueueTable	Information about the Enterprise Replication queue
	onErSiteTable	Information about the Enterprise Replication site
	onFragmentTable	Information about the fragments that are in fragmented database tables
	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 subagent for Extended Parallel Server
XPS (See page 3-31)	**xpsCogroupMember-Table**	Information about coservers as members of cogroups
	xpsCogroupTable	Information about cogroups as members of database servers
	xpsCoserverTable	Information about coservers as members of database servers
	xpsDbsliceDbspaceTable	Information about each dbspace that is in a database server
	xpsDbsliceTable	Information about each dbslice that is in a coserver

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 MIB objects.

MIB Object **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,
  Enterprise ID * 1,000,000 is 893,000,000.
- SERVERNUM * 1000
- Coserver ID for Extended Parallel Server or 0 for any other type of database server

applName
Name of the database server or coserver

applDirectoryName
No OnSNMP support for this MIB object

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.

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 MIB object.

applAccumulatedInboundAssociations
Number of SQLCONNECT actions that have occurred so far

applAccumulatedOutboundAssociations
OnSNMP does not support this MIB object.

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 MIB object.

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 MIB object.

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 MIB objects.

MIB Object **Description**

rdbmsDbIndex See “rdbmsDbTable” on page 3-6.

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) • Kilobytes (2) • Megabytes (3) • Gigabytes (4) • Terabytes (5)
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).

rdbmsDbLimitedResourceTable

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 MIB objects.

MIB Object	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: IBM Corporation
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: (XPS)	**rdbmsDbIndex**, **applIndex**
Scope of a row: (XPS)	One database
	One database coserver pair.

The table has the following MIB objects.

MIB Object	**Description**
rdbmsDbIndex	See “rdbmsDbTable” on page 3-6.
applIndex	See “applTable” on page 3-3.
rdbmsRelState	Access state between the database server or coserver and the database: • Other (1): The database server or coserver is online, 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 online, 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 online, but the database is not open. • Restricted (4): The database is not completely available. The database server or coserver is online, 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 it was started
Index:	**applIndex**
Scope of a row:	One database server or coserver

The table has the following MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
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 the desired disk space
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
rdbmsSrvInfoHighwaterInbound-Associations	Greatest number of inbound associations that have been open at the same time

MIB Object	Description
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 MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
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
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.

MIB Object	Description
rdbmsSrvLimitedResourceDescription	Description of the limited resource. This description includes the units for the value for the limited resource.

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 operating system. For more information about the configuration file, see your *IBM Informix: Administrator's Guide* and the *IBM Informix: Administrator's Reference*. For more information about the **ONCONFIG** environment variable, see the *IBM Informix: Guide to SQL Reference*.

Operating System	Location of Configuration File
UNIX	$INFORMIXDIR/etc/$ONCONFIG
Windows	%INFORMIXDIR%\etc\%ONCONFIG%

The table has the following MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
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 MIB objects.

MIB Object	Description
applIndex	See “applTable” on page 3-3.
rdbmsSrvPrivateMibOID	OID for the Informix Private MIB: 1.3.6.1.4.1.893
rdbmsSrvVendorName	Name of the database server vendor: IBM Corporation
rdbmsSrvProductName	Name of the database server product. For example, this value might be Dynamic Server.
rdbmsSrvContact	Name of the database server contact: informix

rdbmsTraps

This MIB object contains information about traps that an SNMP subsystem that supports the RDBMS MIB can generate. In this case, the SNMP subsystem is OnSNMP.

frdbmsStateChange Trap

When a database server changes from its current status to any less-available status, OnSNMP sends a **rdbmsStateChange** trap message to configured network hosts through the master agent.

The following list summarizes this trap:

Contents:	The **rdbmsRelState** MIB object
Index:	**rdbmsDbIndex**, **applIndex**
Scope of a row:	If the status of an Informix database server becomes unavailable, it generates one trap for each database.

For Extended Parallel Server, if the entire database server becomes unavailable, each coserver causes as many traps to be generated as there are available databases.

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 MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
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
onActiveBarActivitySize	Total number of used pages to scanOnSNMP 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

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 MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-6.
onTableIndex	See "onDbspaceTable" on page 3-16.
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.
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

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 the *IBM Informix: Backup and Restore Guide*. The table has the following MIB objects.

MIB Object **Description**

applIndex	See “applTable” on page 3-3.
onBarActivityIndex	Index to the history
onBarObjectIndex	Index to the object
onBarName	Name of the object
onBarType	Type of object: • blobspace (1) (Only Dynamic Server provides blobspaces.) • rootDbspace (2) • criticalDbspace (3) • noncriticalDbspace (4) • logicalLog (5)
onBarLevel	Level of the backup action: • completeBackup (1) • incrementalLevelOne(2) • incrementalLevelTwo (3)
onBarStatus	Status of the action on the object: • 0 = successful • Nonzero = error number
onBarTimeStamp	Ending time stamp for the action

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 MIB objects.

MIB Object	**Description**
applIndex	See “applTable” on page 3-3.
onDbspaceIndex	See “rdbmsDbInfoTable” on page 3-5.
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

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).

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).

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** MIB object 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 MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-6.
onDatabaseDbspace	Default dbspace
onDatabaseCreated	Creation date and time
onDatabaseLogging	Logging status: • none (1) • buffered (2) • unbuffered (3) • ansi (4)
onDatabaseOpenStatus	Database status: • notOpen (1) • open (2) • openExclusive (3)
onDatabaseUsers	Number of users

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 MIB objects.

MIB Object	**Description**
applIndex	See "applTable" on page 3-3.
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
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

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 had a full backup, this value is noSuchInstance (SNMPv2) or noSuchName (SNMPv1).

onErQueueTable

The following list summarizes this table:

Contents: Information about the replication queues for all database servers that participate in Enterprise Replication

Index: **applIndex**, **onErQueueReplIndex**

Scope of a row: One replication queue

The table has the following MIB objects.

MIB Object **Description**

applIndex See "applTable" on page 3-3.

onErQueueReplIndex Unique integer index that identifies a replicant

onErQueueSiteIndex Unique integer that identifies a database server

onErQueueReplName Display string that describes the replicant or collection of replicants

onErQueueSiteName Name of the Enterprise Replication database server

onErQueueSize Current number of bytes in the send queue

onErQueueLastCommit Date and time when last transaction was committed

onErQueueLastAck Date and time when last data was acknowledged

onErSiteTable

The following list summarizes this table:

Contents:	Information about all the remote database servers that participate in Enterprise Replication
Index:	**applIndex**, **onErSiteIndex**
Scope of a row:	A single replication queue

The table has the following MIB objects.

MIB Object	**Description**
applIndex	See “applTable” on page 3-3.
onErSiteIndex	Integer that uniquely identifies a database server as defined in the group entry in **sqlhosts**
onErSiteName	Name of the replication site
onErSiteState	State of the replication activity for this site: • inactive (1) • active (2) • suspend (3) • quiescent (4) • hold (5) • delete (6) • failed (7) • unknown (8)
onErSiteConnectionState	State of the connection to this site: • idle (1) • connected (2) • disconnected (3) • timeout (4) • shutdown (5) • error (6) • unknown (7)
onErSiteConnectionChange	Date and time when the connection state last changed
onErSiteIdleTimeout	Time limit for Enterprise Replication to wait for new data to send or receive. Value is set

when database server is defined. Connection is closed if time limit is exceeded.

onErSiteOutMsgs Total number of messages transmitted from the current database server to this site

onErSiteOutBytes Total number of bytes transmitted from the current database server to this site

onErSiteInMsgs Total number of messages received by the current database server from this site

onErSiteInBytes Total number of bytes received by the current database server from this site

onErSiteTransactions Total number of transactions received from this site

onErSiteCommits Total number of transactions received and committed from this site

onErSiteAborts Total number of transactions aborted from this site

onErSiteLastReceived Date and time when the last transaction was processed from this site

onErSiteRowCommits Total number of rows committed from this site

onErSiteRowAborts Total number of rows aborted from this site

onErSiteRcvLatency Average latency between the source commit time and target receive time; performance measure of network queueing delay

onErSiteCommitLatency Average latency between source and target commit time; performance measure of network and database server delay

onErSiteClockErrors Number of transactions received from this site with a time that is ahead of our current time; indicates system clock synchronization problems

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 MIB objects.

MIB Object	Description
applIndex	See “applTable” on page 3-3.
rdbmsDbIndex	See “rdbmsDbTable” on page 3-6.
onTableIndex	See “onDbspaceTable” on page 3-16.
onFragmentIndex	Unique integer index for the fragment
onFragmentType	Type of database table: • fragmentedIndex (1) • fragmentedTable (2)
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).
onFragmentUsers	Number of user threads that access the fragment.
onFragmentLockRequests	Number of locks of any type requested for this fragment.
onFragmentLockWaits	Number of times an initial lock request failed because the lock could not be granted initially for the fragment.

onFragmentLockTimeouts
Number of deadlock timeouts for the fragment.

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

A row exists for each lock that the session is using and for each lock on which the session is waiting.

The table has the following MIB objects.

MIB Object **Description**

applIndex
See “applTable” on page 3-3.

onSessionIndex
See “onServerTable” on page 3-24.

onLockIndex
Index to this row

onLockDatabaseName
Name of the database that is using or waiting for this lock

onLockTableName
Name of the table that is 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 that are 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 MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
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

MIB Object	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

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 MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
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 MIB objects.

MIB Object	Description
applIndex	See “applTable” on page 3-3
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
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

MIB Object	Description
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
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 hundredths of second
onServerCPUSystemTime	Amount of CPU time that the database server has used in System Mode, in hundredths of second
onServerCPUUserTime	Amount of CPU time that the database server has used in User Mode, in hundredths of second

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 MIB objects.

MIB Object	Description
applIndex	See “applTable” on page 3-3.
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
onSessionUserTime	Length of time that the user has been connected to the database server or coserver, in hundredths of seconds

MIB Object	Description
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
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)

MIB Object	Description
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).
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 (XPS)	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 operating system. For details about the connection information, see your *IBM Informix: Administrator's Guide*.

Operating System	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 information about **INFORMIXSQLHOSTS**, see the *IBM Informix: Guide to SQL Reference*.
Windows	The connectivity information is in a key in the Windows registry called **HKEY_LOCAL_MACHINE\SOFTWARE\ Informix\SQLHOSTS**.

The table has the following MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
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 MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-6.
onTableIndex	Table number

This value is the same as **tabid** in the system catalog table **systables**

onTableName
Table name

onTableOwner
Table owner

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).

onXpsTable

The following list summarizes this table:

Contents:	The OnSNMP instance that is the subagent for Extended Parallel Server
Index:	**applIndex**
Scope of a row:	One coserver

For information about the subagent, see "SNMP and Extended Parallel Server (XPS)" on page 2-17. The table has the following MIB objects.

MIB Object	**Description**
applIndex	See "applTable" on page 3-3.
onXpsSnmpIndex	Unique integer index that identifies the 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 subagent runs

XPS MIB in the Informix Private MIB

The XPS MIB defines several tables that provide information that is specifically relevant for database servers, coservers, and their databases on Extended Parallel Server.

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 MIB objects.

MIB Object	**Description**

applIndex	See “applTable” on page 3-3.
xpsCogroupIndex	See “xpsCogroupTable” on page 3-32.
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 Extended Parallel Server
Index:	**applIndex**, **xpsCogroupIndex**
Scope of a row:	One cogroup

The table has the following MIB objects.

MIB Object	**Description**
applIndex	See “applTable” on page 3-3.
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 Extended Parallel Server
Index:	**applIndex**, **xpsCoserverIndex**
Scope of a row:	One coserver

The table has the following MIB objects.

MIB Object	Description
applIndex	See “applTable” on page 3-3.

MIB Object	Description
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 Extended Parallel Server
Index:	**applIndex**, **xpsDbsliceIndex**, **xpsDbsliceDbspaceIndex**
Scope of a row:	One dbspace

The table has the following MIB objects.

MIB Object	Description
applIndex	See "applTable" on page 3-3.
xpsDbsliceIndex	See "xpsDbsliceTable" on page 3-33.
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 Extended Parallel Server
Index:	**applIndex**, **xpsDbsliceIndex**
Scope of a row:	One dbslice

The table has the following MIB objects.

MIB Object	Description
applIndex	See “applTable” on page 3-3.
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	Indicates whether or not the dbslice is a blobslice: • yes (1) • no (2)
xpsDbsliceIsTemp	Indicates whether or not the dbslice is temporary: • yes (1) • no (2)

Appendix. Accessibility

The syntax diagrams in the HTML version of this manual are available in dotted decimal syntax format, which is an accessible format that is available only if you are using a screen reader.

Dotted Decimal Syntax Diagrams

In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), the elements can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, `3` or `3.1` or `3.1.1`. To hear these numbers correctly, make sure that your screen reader is set to read punctuation. All syntax elements that have the same dotted decimal number (for example, all syntax elements that have the number `3.1`) are mutually exclusive alternatives. If you hear the lines `3.1 USERID` and `3.1 SYSTEMID`, your syntax can include either `USERID` or `SYSTEMID`, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number `3` is followed by a series of syntax elements with dotted decimal number `3.1`, all the syntax elements numbered `3.1` are subordinate to the syntax element numbered `3`.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, the word or symbol is preceded by the backslash (\) character. The `*` symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element `*FILE` with dotted decimal number 3 is read as `3 \* FILE`. Format `3* FILE` indicates that syntax element `FILE` repeats. Format `3* \* FILE` indicates that syntax element `* FILE` repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol that provides information about the syntax elements. For example, the lines `5.1*`, `5.1 LASTRUN`, and `5.1 DELETE` mean that if you use more than one of the `LASTRUN` and `DELETE` syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this identifies a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line `2.1 %OP1` means that you should refer to a separate syntax fragment `OP1`.

The following words and symbols are used next to the dotted decimal numbers:

? Specifies an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element (for example, `5? NOTIFY`). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines `5 ?`, `5 NOTIFY`, and `5 UPDATE`, you know that syntax elements `NOTIFY` and `UPDATE` are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! Specifies a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicates that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines `2? FILE`, `2.1! (KEEP)`, and `2.1 (DELETE)`, you know that `(KEEP)` is the default option for the `FILE` keyword. In this example, if you include the `FILE` keyword but do not specify an option, default option `KEEP` is applied. A default option also applies to the next higher dotted decimal number. In this example, if the `FILE` keyword is omitted, default `FILE(KEEP)` is used. However, if you hear the lines `2? FILE`, `2.1`, `2.1.1! (KEEP)`, and `2.1.1 (DELETE)`, the default option KEEP only applies to the next higher dotted decimal number, `2.1` (which does not have an associated keyword), and does not apply to `2? FILE`. Nothing is used if the keyword `FILE` is omitted.

***** Specifies a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line `5.1* data-area`, you know that you can include more than one data area or you can include none. If you hear the lines `3*`, `3 HOST`, and `3 STATE`, you know that you can include `HOST`, `STATE`, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write `HOST STATE`, but you could not write `HOST HOST`.
3. The `*` symbol is equivalent to a loop-back line in a railroad syntax diagram.

+ Specifies a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times. For example, if you hear the line `6.1+ data-area`, you must include at least one data area. If you hear the lines `2+`, `2 HOST`, and `2 STATE`, you know that you must include `HOST`, `STATE`, or both. As for the * symbol, you can only repeat a particular item if it is the only item with that dotted decimal number. The `+` symbol, like the `*` symbol, is equivalent to a loop-back line in a railroad syntax diagram.

Glossary

agent. An interface to a managed object, such as a workstation or an application running on a workstation, that responds to a request from a manager.

ASN.1. Acronym for Abstract Syntax Notation One, a 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 “MIBs” on page 1-6.

coserver. The functional equivalent of a database server that operates on a single node. References to coserver in this manual apply only to Extended Parallel Server.

IANA. Acronym for Internet Assigned Numbers Authority, which defines a hierarchy for naming tables and columns and for deriving numerical object identifiers (OIDs). IANA assigns identifiers to companies that use the SNMP protocol.

IETF. Acronym for Internet Engineering Task Force, which specifies the structure of public MIBs and MIB tables.

infxsnmp.dll. Library for the Windows 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 a 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, which is a group of tables that contain the information that a subagent can provide to a master agent.

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

onsnmp. Subagent program for the UNIX version of OnSNMP.

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

OnSNMP. Subagent for an Informix database server or coserver. On UNIX, OnSNMP consists of **onsnmp**. In Windows, OnSNMP consists of **onsnmp.exe**, **infxsnmp.dll**, and **onsrvapd.exe**.

onsrvapd. Server discovery program for the UNIX version of OnSNMP.

onsrvapd.exe. Server discovery program for the Windows version of OnSNMP.

runsnmp.ksh. Helper script to ensure that both the **onsrvapd** server discovery daemon and the SNMP master agent are running on a host.

Server Discovery Process. The discovery process discovers multiple server instances running on the host. These instances might belong to different versions that are installed on different directories.

SMI. In the context of SNMP, acronym for the Structure of Management Information, which defines the common interface definition language.

SNMP. Acronym for Simple Network Management Protocol, a 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 "SNMP Standard" on page 2-4.

subagent. Program that lets a managed component and a master agent communicate with each other.

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

> © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. (enter the year or years). All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

AIX; DB2; DB2 Universal Database; Distributed Relational Database Architecture; NUMA-Q; OS/2, OS/390, and OS/400; IBM Informix®; C-ISAM®; Foundation.2000™; IBM Informix ® 4GL; IBM Informix®DataBlade®Module; Client SDK™; Cloudscape™; Cloudsync™; IBM Informix®Connect; IBM Informix®Driver for JDBC; Dynamic Connect™; IBM Informix®Dynamic Scalable Architecture™(DSA); IBM Informix®Dynamic Server™; IBM Informix®Enterprise Gateway Manager (Enterprise Gateway Manager); IBM Informix®Extended Parallel Server™; i.Financial Services™; J/Foundation™; MaxConnect™; Object Translator™; Red Brick™; IBM Informix® SE; IBM Informix® SQL; InformiXML™; RedBack®; SystemBuilder™; U2™; UniData®; UniVerse®; wintegrate®are trademarks or registered trademarks of International Business Machines Corporation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Windows, Windows NT, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names used in this publication may be trademarks or service marks of others.

Index

Printed in USA

G251-2298-00