Informix Product Family
Informix
Version 12.10

IBM Informix SNMP Subagent Guide

SC27-4533-00

Informix Product Family
Informix
Version 12.10

IBM Informix SNMP Subagent Guide

SC27-4533-00

Note
Before using this information and the product it supports, read the information in "Notices" on page B-1.

Contents

Introduction

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

About this publication

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

Types of users

This publication is written for the following users:

- Database server administrators
- Backup operators
- Performance engineers

This publication 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 compatibility

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

The **onsnmp** utility cannot be run on HDR secondary servers, remote standalone (RS) secondary servers, or shared disk (SD) secondary servers.

Assumptions about your locale

IBM Informix products can support many languages, cultures, and code sets. All the information related to character set, collation and representation of numeric data, currency, date, and time that is used by a language within a given territory and encoding is brought together in a single environment, called a Global Language Support (GLS) locale.

The IBM Informix OLE DB Provider follows the ISO string formats for date, time, and money, as defined by the Microsoft OLE DB standards. You can override that default by setting an Informix environment variable or registry entry, such as **DBDATE**.

If you use Simple Network Management Protocol (SNMP) in your Informix environment, note that the protocols (SNMPv1 and SNMPv2) recognize only English code sets. For more information, see the topic about GLS and SNMP in the *IBM Informix SNMP Subagent Guide*.

The examples in this publication are written with the assumption that you are using one of these locales: en_us.8859-1 (ISO 8859-1) on UNIX platforms or

en_us.1252 (Microsoft 1252) in Windows environments. These locales support U.S. English format conventions for displaying and entering date, time, number, and currency values. They also support the ISO 8859-1 code set (on UNIX and Linux) or the Microsoft 1252 code set (on Windows), which includes the ASCII code set plus many 8-bit characters such as é, è, and ñ.

You can specify another locale if you plan to use characters from other locales in your data or your SQL identifiers, or if you want to conform to other collation rules for character data.

For instructions about how to specify locales, additional syntax, and other considerations related to GLS locales, see the *IBM Informix GLS User's Guide*.

Demonstration databases

The DB-Access utility, which is provided with your IBM 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 publications are based on the **stores_demo** database.
- The **superstores_demo** database illustrates an object-relational schema. The **superstores_demo** database contains examples of extended data types, type and table inheritance, and user-defined routines.

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 are in the `$INFORMIXDIR/bin` directory on UNIX platforms and in the `%INFORMIXDIR%\bin` directory in Windows environments.

Example code conventions

Examples of SQL code occur throughout this publication. 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 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. If you are using DB–Access, you must delimit multiple statements with semicolons.

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 documentation for your product.

Additional documentation

Documentation about this release of IBM Informix products is available in various formats.

You can access Informix technical information such as information centers, technotes, white papers, and IBM Redbooks® publications online at http://www.ibm.com/software/data/sw-library/.

Compliance with industry standards

IBM Informix products are compliant with various standards.

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.

The IBM Informix Geodetic DataBlade® Module supports a subset of the data types from the *Spatial Data Transfer Standard (SDTS)—Federal Information Processing Standard 173*, as referenced by the document *Content Standard for Geospatial Metadata*, Federal Geographic Data Committee, June 8, 1994 (FGDC Metadata Standard).

Syntax diagrams

Syntax diagrams use special components to describe the syntax for statements and commands.

Table 1. Syntax Diagram Components

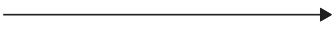
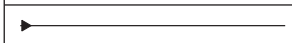
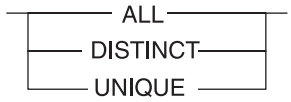
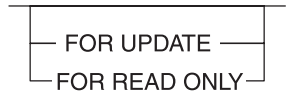
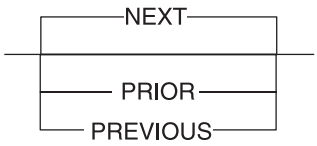
Component represented in PDF	Component represented in HTML	Meaning
	>>----------	Statement begins.
	---------->	Statement continues on next line.
	>----------	Statement continues from previous line.
	----------><	Statement ends.
	--------SELECT----------	Required item.
	--+----------+--- '------LOCAL------'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

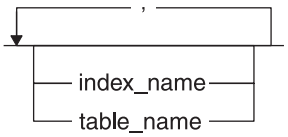
Table 1. Syntax Diagram Components (continued)

Component represented in PDF	Component represented in HTML	Meaning

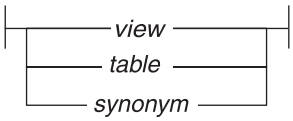
	```---+-----ALL-------+---
	  +--DISTINCT-----+
	  '---UNIQUE------'```
		Required item with choice. Only one item must be present.

	```---+------------------+---
	  +--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 is used by default.

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

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

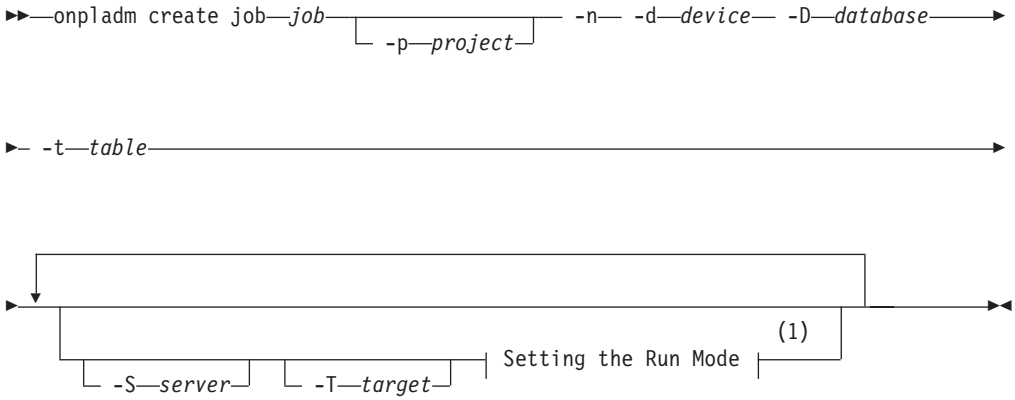
Table Reference
 
	``Table Reference
	|--+-----view--------+--|
	  +------table------+
	  '----synonym------'```
		Syntax segment.

How to read a command-line syntax diagram

Command-line syntax diagrams use similar elements to those of other syntax diagrams.

Some of the elements are listed in the table in Syntax Diagrams.

Creating a no-conversion job

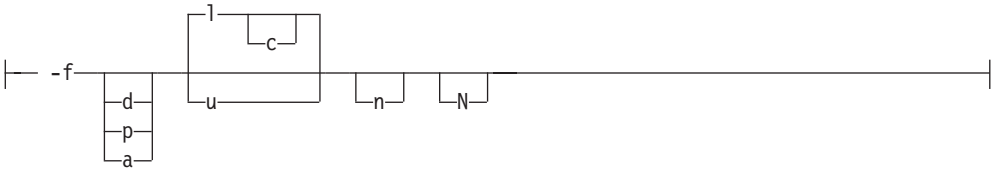


Notes:

1 See page Z-1

This diagram has a segment named "Setting the Run Mode," which according to the diagram footnote is on page Z-1. If this was an actual cross-reference, you would find this segment on the first page of Appendix Z. Instead, this segment is shown in the following segment diagram. Notice that the diagram uses segment start and end components.

Setting the run mode:



To see how to construct a command correctly, start at the upper left of the main diagram. Follow the diagram to the right, including the elements that you want. The elements in this diagram are case-sensitive because they illustrate utility syntax. Other types of syntax, such as SQL, are not 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.

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.

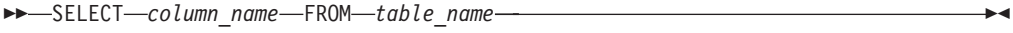
You must also use any punctuation in your statements and commands exactly as shown in the syntax diagrams.

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.



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.

How to provide documentation feedback

You are encouraged to send your comments about IBM Informix user documentation.

Use one of the following methods:

- Send email to docinf@us.ibm.com.
- In the Informix information center, which is available online at http://www.ibm.com/software/data/sw-library/, open the topic that you want to comment on. Click the feedback link at the bottom of the page, fill out the form, and submit your feedback.
- Add comments to topics directly in the information center and read comments that were added by other users. Share information about the product documentation, participate in discussions with other users, rate topics, and more!

Feedback from all methods is monitored by the team that maintains the user documentation. The feedback methods are reserved for reporting errors and omissions in the documentation. For immediate help with a technical problem, contact IBM Technical Support at http://www.ibm.com/planetwide/.

We appreciate your suggestions.

Chapter 1. SNMP concepts

This section provides a brief introduction to Simple Network Management Protocol (SNMP).

What is SNMP?

The 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.

Purpose of the SNMP

Although the original purpose of the Simple Network Management Protocol (SNMP) was to let network administrators remotely manage an Internet system, the design of SNMP lets network administrators manage applications and systems.

SNMP provides the following capabilities:

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

The SNMP architecture

The Simple Network Management Protocol (SNMP) architecture includes four layers.

As the following figure illustrates, the SNMP architecture includes the following layers:

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

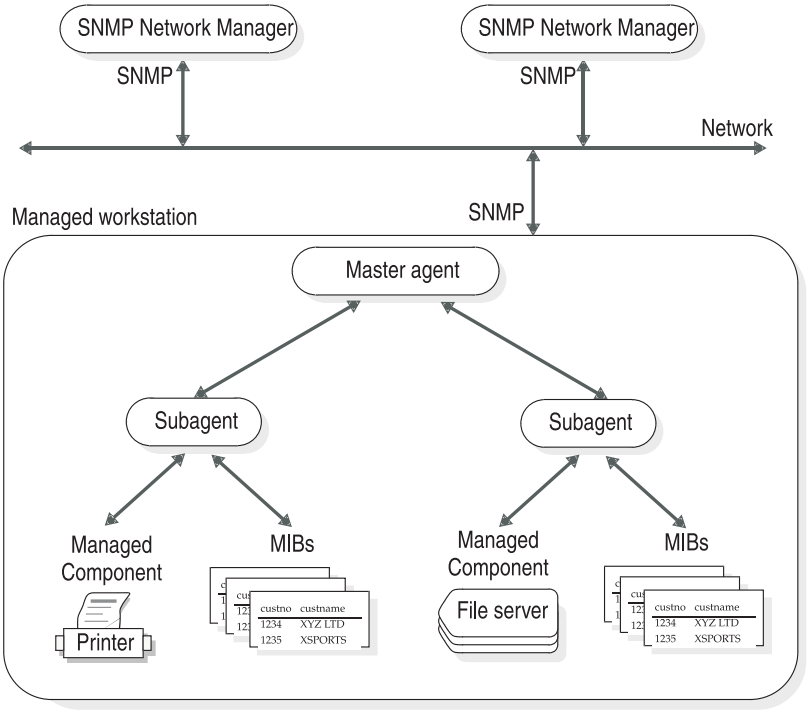


Figure 1-1. SNMP architecture

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. You can use most SNMP Network Managers to 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 the following figure illustrates. With this SNMP Network Manager, you select a node to monitor and then choose specific information from a menu.

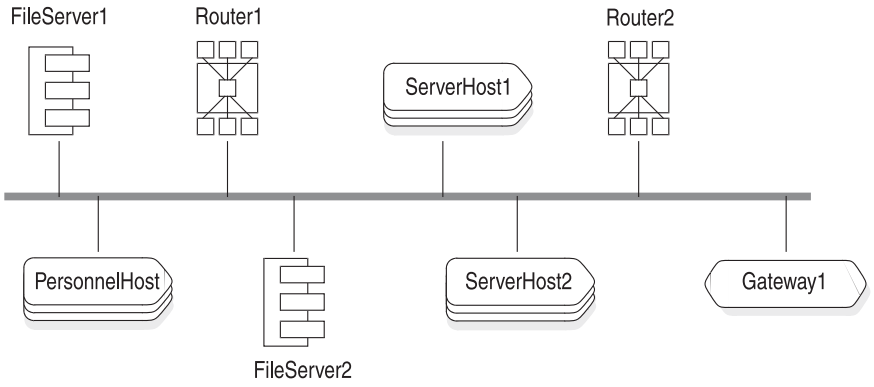


Figure 1-2. SNMP Network Manager example

The following code 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
```

The following code 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
```

In addition to text, an SNMP Network Manager might also display graphs or charts, as the following figure illustrates.

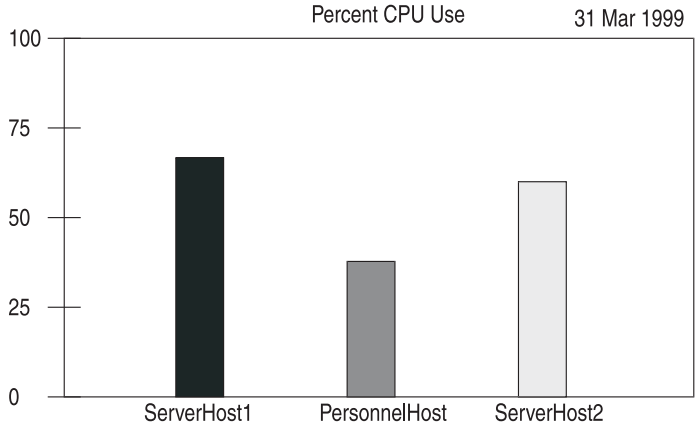


Figure 1-3. Example of monitoring information

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 need to 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 flash 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 need to generate a trap.
- Specify how the SNMP Network Manager notifies you of a trap.

  You might choose to have an icon flash 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.

Management Information Bases

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.

MIB table 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**.

The MIB 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 numeric object identifiers (OIDs). The following figure shows the MIB hierarchy.

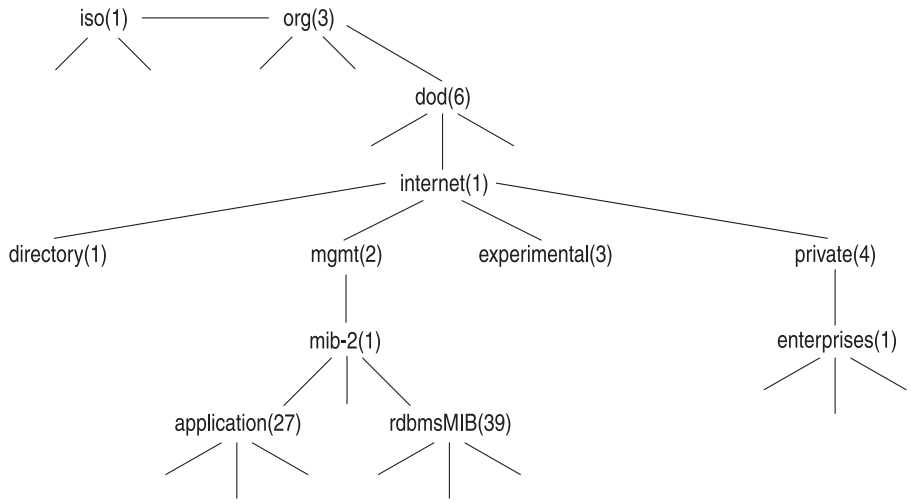


Figure 1-4. MIB hierarchy

Even though you rarely see the full path to a table, column, or value, the path is important because the SNMP components use the numeric 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 numeric 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.

Related concepts:

"MIB types and objects" on page 2-15

Chapter 2. Informix implementation of SNMP

The IBM Informix implementation of SNMP lets database administrators monitor Informix database servers and databases.

Components of the Informix implementation

The IBM Informix implementation consists of the following components:

- Master agent
  - On UNIX, a master agent is provided through licensing agreements with vendors.
  - On Windows, install the Microsoft SNMP Extendible master agent.
- Subagent

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

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

  OnSNMP uses several MIBs.

Related tasks:

"Windows master agent" on page 2-12

Related reference:

"UNIX master agents" on page 2-6

Purpose of Informix SNMP

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.

Table 2-1. Possible events and the corresponding MIB objects

Event	MIB object
A database server 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 IBM 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 flash to notify you of the problem. You can then send data requests to determine the cause of the failure.

Related tasks:

"Installing and configuring a master agent manually" on page 2-7

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. OnSNMP provides information that lets an SNMP Network Manager monitor all the IBM Informix databases that are on a network. The following figure illustrates this concept.

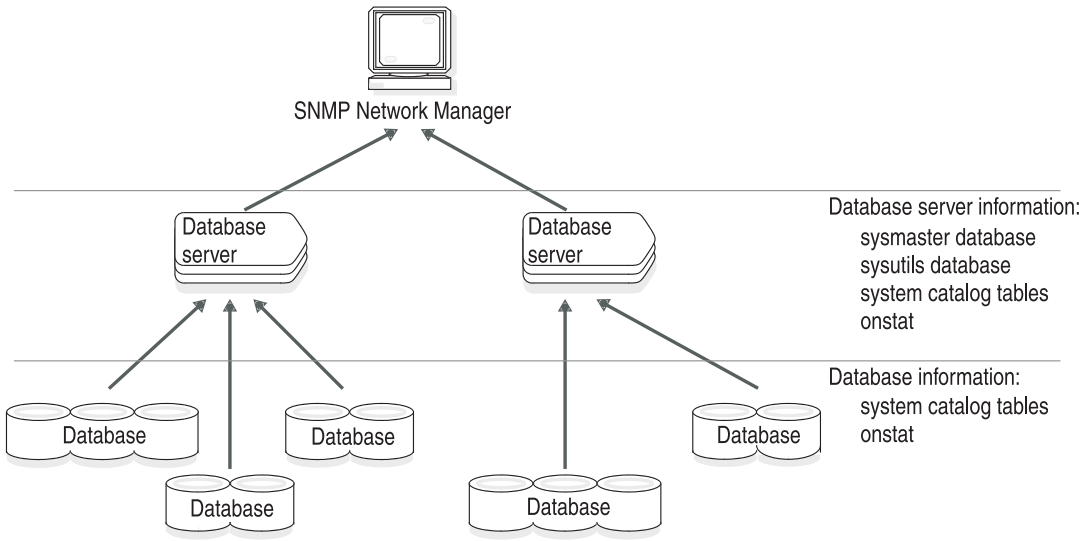


Figure 2-1. Monitoring Informix databases

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.

Table 2-2. Versions of the SNMP standard

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.

SNMP is incompatible on High-Availability Data Replication (HDR) secondary servers, remote standalone (RS) secondary servers, or shared disk (SD) secondary servers.

Informix SNMP architecture on UNIX

The following figure 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 has one OnSNMP process.

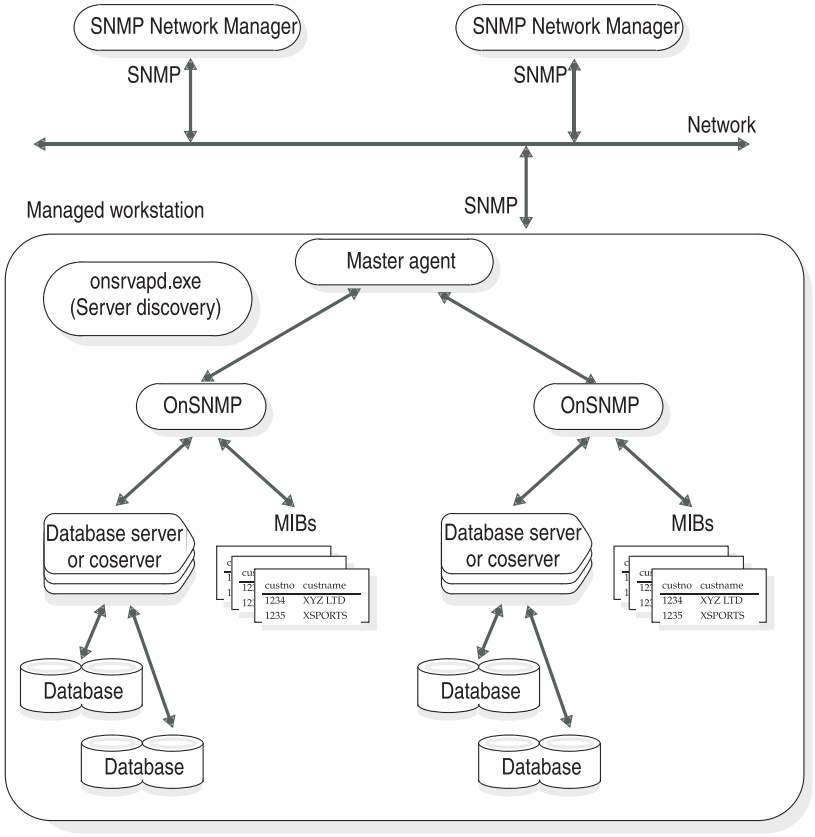


Figure 2-2. Informix SNMP architecture on UNIX

Informix SNMP architecture on Windows

The following figure 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 Informix that runs on the managed workstation. OnSNMP and the master agent do not need to use SNMP to communicate with each other.

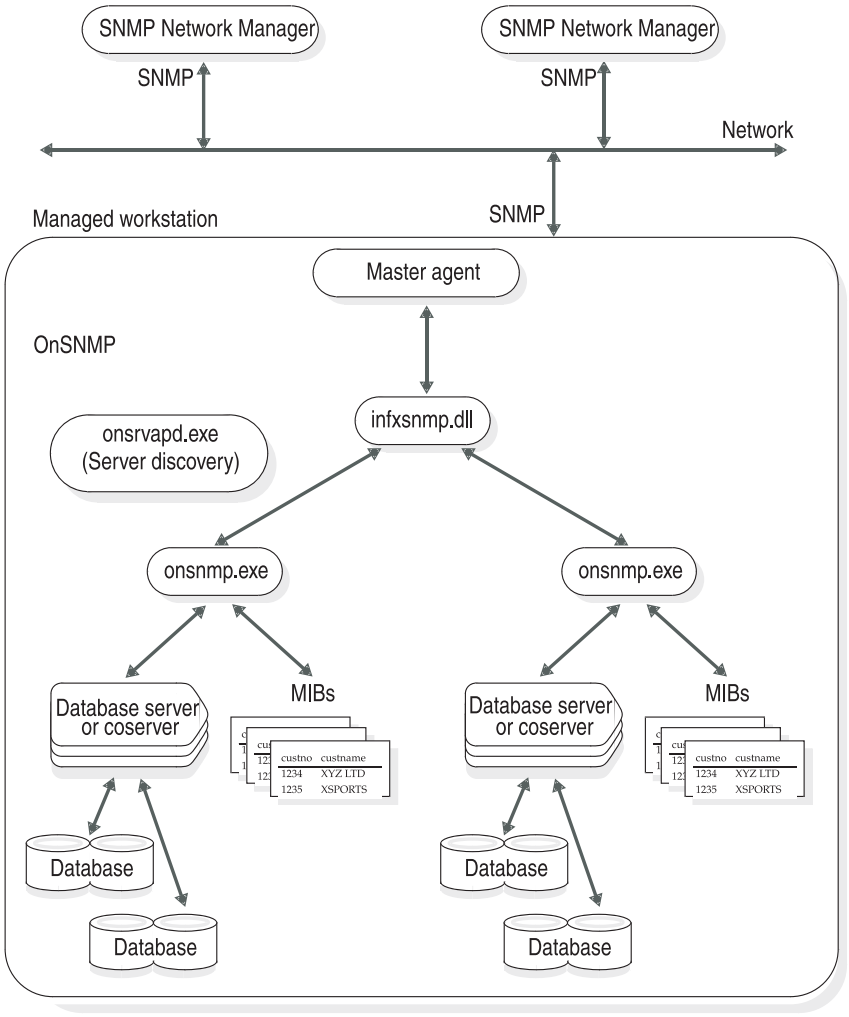


Figure 2-3. Informix SNMP architecture on Windows

Informix implementation of SNMP on UNIX or Linux

To use the IBM Informix implementation of SNMP on UNIX or Linux, 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 IBM Informix database server
- An IBM Informix database server

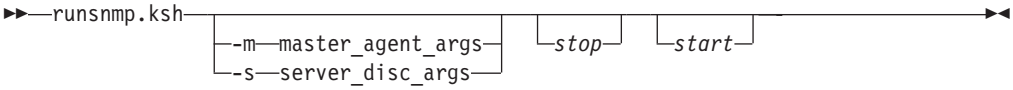
  When you install an Informix database server, the installation procedure installs the OnSNMP subagent and the server discovery process as well as the files needed for SNMP support.

  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 creates an instance of OnSNMP to monitor the database server.

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**.



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

Table 2-3. The runsnmp.ksh commands

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** and **onsrvapd** if they are not running. This option is the default.
stop	Stops **snmpdm** 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** 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**, **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** if it is not running, and then start **onsrvapd** with the `none` option, if it is not running.
  ```
  runsnmp.ksh -s "-rnone"
  ```

Related reference:

"UNIX subagent" on page 2-9

UNIX master agents

On UNIX, master agents are provided through licensing agreements.

The following table lists these master agents.

Master Agent	Company	Website
EMANATE, Version 14.2	SNMP Research	www.snmp.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.

Related reference:

"Components of the Informix implementation" on page 2-1

Assuring compatibility

The following guidelines assure master agent compatibility:

- Only one master agent is provided, usually EMANATE, 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.
- Only run 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 instances exist.
- IBM 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** is similar to the following lines:

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

   b. Make sure that UDP port 161 is available so that the master agent can be the owner of the port.
3. Add the following line to the snmp configuration file for the **snmpd** daemon to accept messages from **onsnmp**:

```
smuxpeer 0.0
```

If this line does not exist, and the **snmpd** daemon is log enabled, the following message is reported:

```
 snmpd log:
refused smux peer: oid SNMPv2-SMI::zeroDotZero, password , descr rdbms subagent
 onsrvapd log:
INFO :  onsrvapd pid 9045, poll 5 secs, linger 5 mts, logfile
/tmp/onsrvapd.42f0d7392355.log.
MAJOR:  signalCatcher - Caught SIGCHLD.
MAJOR:  childKilled - Subagent pid 9046 Status 65280.
 onsnmp log:
MAJOR:  SMUX subagent failed to instantiate managed row
```

Related concepts:

"Traps" on page 2-2

Starting and stopping a master agent

Start the master agent before you start an IBM Informix database server, and stop all IBM Informix database servers 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.

The `runsnmp.ksh` script automatically starts the EMANATE master agent at startup and stops it at shutdown.

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

Starting a master agent manually:

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 &`

Stopping a master agent manually:

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**

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

Table 2-4. The snmpdm command-line options

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

Making sure that a master agent is running correctly:
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 "Installing and configuring a master agent manually" on page 2-7 is set to a different directory.
2. Verify that the process is running:

   For EMANATE, **snmpdm**

UNIX subagent

When you install an IBM Informix database server 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.

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

Table 2-5. Additional files provided with the database server

Program	Description
onsrvapd daemon	When you start an Informix database server that is on this workstation, **onsrvapd** detects this event and starts OnSNMP for the database server. When the database server halts, **onsrvapd** stops OnSNMP for that database server.
`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.

Related concepts:

"UNIX server discovery process"

Related reference:

"The runsnmp.ksh script" on page 2-6

UNIX server discovery process

The `runsnmp.ksh` script automatically starts the UNIX server discovery process. 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 the principles for a master agent: start **onsrvapd** before you start an IBM Informix database server, and stop all IBM Informix database servers on a workstation before you stop **onsrvapd**.

Related reference:

"UNIX subagent" on page 2-9

Preparing onsrvapd manually

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

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.

Issue the onsrvapd command

You can specify the **onsrvapd** command-line options that the following syntax shows. Some of these options affect OnSNMP.

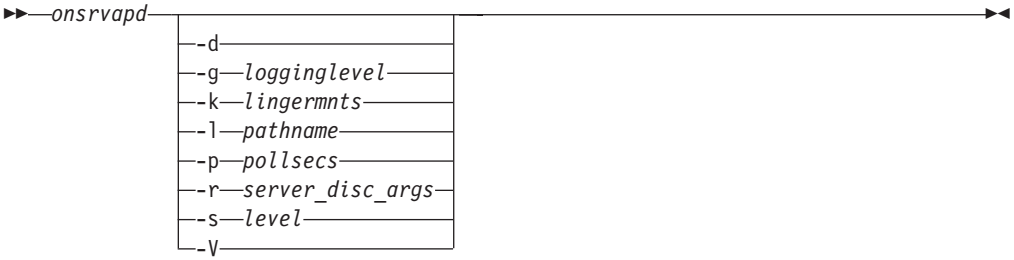


Table 2-6. The onsrvapd command-line options

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.

Table 2-6. The onsrvapd command-line options (continued)

Option	Description
-k *lingerings*	Number of minutes that **onsrvapd** waits after a database server goes down before **onsrvapd** kills the corresponding OnSNMP. If *lingermnts* is 0, **onsrvapd** waits indefinitely.
-l *pathname*	Directory for the error log files. The file name of the OnSNMP error log is `onsnmp.`*`servername`*`.log`. For example, if your server name is MyServer, the file name of the OnSNMP error log is `onsnmp.MyServer.log`. The file name of the **onsrvapd** error log is `onsrvapd.log`.
-p *pollsecs*	Frequency, in seconds, with which OnSNMP polls the database server. 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 2-18.
-V	Prints the OnSNMP version number.

Related reference:

"Refresh control value" on page 2-18

Starting onsrvapd manually:

To start **onsrvapd** manually:

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

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

Making sure that onsrvapd is running correctly:

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.

Choose an installation directory

When you have multiple IBM 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.

Informix implementation of SNMP on Windows

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

- Microsoft SNMP service on each workstation that includes an Informix database server
- An Informix database server

  When you install an Informix database server, 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.

Related reference:

"Components of the Informix implementation" on page 2-1

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.

Table 2-7. OnSNMP files and associated directories

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.	`%INFORMIXDIR%\bin`

`onsrvapd.exe`	Server discovery process, which starts `onsnmp.exe` for each database server 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
		32-bit platforms:
		`%Windows%\system32`
		
		64-bit platforms:
		`Windows\SysWOW64`

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

Start and stop 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.

Configure OnSNMP

The IBM Informix installation procedure creates a registry key, **OnSnmpSubagent**, under `HKEY_LOCAL_MACHINE\SOFTWARE\Informix`.

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

Table 2-8. OnSnmpSubagent arguments that can be changed

Argument	Value	Description
Environment\ LINGER_TIME	*lingermnts*	Number of minutes that the master agent waits after a database server 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 file name
Environment\ REFRESH_TIME	*pollsecs*	Frequency, in seconds, with which OnSNMP polls the database server
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 not change.

Table 2-9. OnSnmpSubagent arguments that do not get changed

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

The IBM Informix installation procedure also creates an 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 registry key for the OnSNMP logging level

On Windows, there is a registry entry to specify the logging level to which OnSNMP logs debugging information.

The logging levels that you can specify are:

- 6 (unrecoverable error conditions)
- 5 (major error conditions)
- 4 (warnings in the program)

- 3 (general information)
- 2 (debug information)
- 1 (dump all information)

Windows server discovery process

The IBM 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 creates an instance of OnSNMP to monitor the database server.

Start and stop 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 "Issue the onsrvapd command" on page 2-10.

Ensure that onsrvapd is running correctly, by checking the log file to verify that **onsrvapd** has not generated any errors. For location of the log files, see your release notes. Verify that **onsrvapd** is running.

Installing the Informix SNMP agent

If you install the Microsoft SNMP Extendible master agent after you install the IBM Informix database server, 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.

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 US English locale when it sends information to the master agent. If OnSNMP cannot convert the code set of the database to the US 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.

MIB types and objects

This section describes the types of MIBs and the types of MIB objects that the IBM Informix database server uses.

OnSNMP uses the following MIBs:

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

Application MIB

The Application MIB is a public MIB, which means that the Internet Engineering Task Force (IETF) specifies the structure of the MIB and the MIB tables. A public MIB is the same for all managed components on an SNMP network, not just for IBM Informix products.

OnSNMP uses only **applTable**, which is the portion of the Application MIB that the RDBMS MIB requires.Figure 1-4 on page 1-6 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 Relational Database Management System (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 1-4 on page 1-6 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 and databases. The following figure shows the MIB hierarchy for the Informix Private MIB.

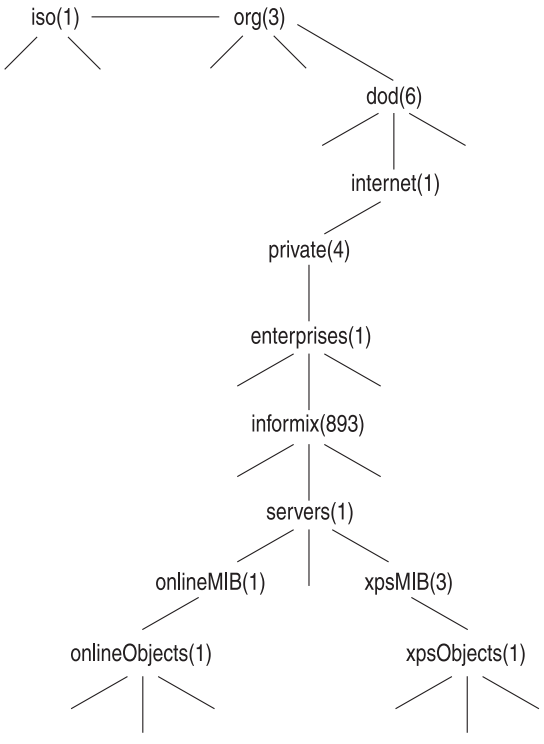


Figure 2-4. MIB hierarchy for the Informix Private MIB

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

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

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

1.3.6.1.4.1.893

Online MIB

The Online MIB is in the Informix Private MIB. The Online MIB contains information for all Informix database servers.

In the Online MIB, all tables are after 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

MIB objects

An MIB object is similar to a column in a table.

The 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 in "Refresh control value" on page 2-18) is `once` or `all`.
- Enterprise Replication objects are tables that exist only if a database server is configured to participate in Enterprise Replication.

Related concepts:

"Management Information Bases" on page 1-5

Table indexing

In the description of the MIBs in Chapter 3, "Management Information Base reference," on page 3-1, 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 3-5.

Table 2-10. Values to interpret the example

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

Alphabetical index values

When an index is an alphabetic string, such as the name of a configuration parameter, the OID for that index consists of the 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 3-7.

Table 2-11. Values to interpret the example

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.

Table 2-12. MIB tables affected by options

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.

Table 2-13. Possible values for 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.

Table 2-14. Default refresh control values

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.

Related concepts:

"Issue the onsrvapd command" on page 2-10

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 or Linux

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

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

Table 2-15. Files installed in $INFORMIXDIR/bin

File name	Description
`onsnmp`	OnSNMP executable file
`onsrvapd`	Server discovery process
`snmpdm`	EMANATE executable or a dummy file for UNIX platforms that EMANATE does not support

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

Table 2-16. Files installed in $INFORMIXDIR/snmp

File name	Description
./snmpr/snmpd.cnf	EMANATE configuration file or a dummy file for UNIX platforms that EMANATE does not support
.runsnmp.ksh	Script that starts the master agent and **onsrvapd**

OnSNMP uses the following log files by default.

Table 2-17. Default log files

File name	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 IBM Informix, the path is **onsnmp.servername.log**	Log file for **onsnmp**.

Files installed on Windows

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

Table 2-18. Files created in %Windows%\system32

File name	Description
infxsnmp.dll	DLL for OnSNMP
onsrvapd.exe	Server discovery process

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

Table 2-19. Files created in %INFORMIXDIR%\bin

File name	Description
onsnmp.exe	OnSNMP executable

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

Chapter 3. Management Information Base reference

An SNMP Network Manager hides most of the structures of the Management Information Base (MIB). However, an understanding of this structure can help you comprehend the information that an SNMP Network Manager displays.

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

Table 3-1. Directories for MIB files

Operating system	MIB directory
UNIX	`$INFORMIXDIR/snmp`
Windows	`%INFORMIXDIR%\etc`

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

This section 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 topics that contains more information.

Table 3-2. MIB tables that OnSNMP uses

MIB	Table	Description
Application	**applTable**	Attributes for each database server
RDBMS	**rdbmsDbInfoTable**	Information about databases
	rdbmsDbTable	Information about databases
	rdbmsRelTable	Information about the relationship between a database and the database server with which it is associated
	rdbmsSrvInfoTable	Information about the database server since it was started
	rdbmsSrvLimited-ResourceTable	Information about the limited resources for each database server
	rdbmsSrvParamTable	Information about the configuration parameters for each database server
	rdbmsSrvTable	Information about a database server
	rdbmsTraps	Information about the traps that OnSNMP can send to the SNMP Network Manager
Online	**onActiveBarTable**	Information about the current ON-Bar activity
	onActiveTableTable	Information about the open and active database tables
	onBarTable	Information about the backup and restore history
	onChunkTable	Information about the chunks that the database servers use
	onDatabaseTable	Information about active databases
	onDbspaceTable	Information about dbspaces
	onErQueueTable	Information about the Enterprise Replication queue
	onErSiteTable	Information about the Enterprise Replication site

Table 3-2. MIB tables that OnSNMP uses (continued)

MIB	Table	Description
	onFragmentTable	Information about the fragments that are in fragmented database tables
	onLockTable	Information about the active locks that database servers are using
	onLogicalLogTable	Information about logical logs
	onPhysicalLogTable	Information about physical logs
	onServerTable	Status and profile information about each active database server
	onSessionTable	Information about each session
	onSqlHostTable	Copy of the connection information
	onTableTable	Information about a database table

Application MIB

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

applTable

The following list summarizes this table:

Contents:
Attributes for each database server

Index: applIndex

Scope of a row:
One database server

The table has the following MIB objects.

Table 3-3. MIB objects for applTable

MIB object	Description
applIndex	Unique integer index that identifies each database server. This value is the sum of the following values: • IBM Informix Enterprise ID * 1,000,000 The IBM Informix Enterprise ID is 893. Therefore, Enterprise ID * 1,000,000 is 893,000,000. • SERVERNUM * 1000
applName	Name of the database server.
applDirectoryName	No OnSNMP support for this MIB object.
applVersion	Version of the database server.
applUptime	Time when the database server was last initialized. This time is the system time according to the master agent. If the database server was last initialized before OnSNMP was last initialized, this value is 0.

Table 3-3. MIB objects for applTable (continued)

MIB object	Description
applOperStatus	Operating status of the database server: • up (1) • down (2) • halted (3) • (4): OnSNMP does not use this value. • restarting (5)
applLastChange	Time when the database server entered its current state. This time is the system time according to the master agent. If the database server 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. 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 rejected an input connection due to administrative reasons or resource limitations.
applFailedOutboundAssociations	OnSNMP does not support this MIB object.

RDBMS MIB

The Relational Database Management System (RDBMS) MIB defines several tables that provide information about managed database servers and their databases.

OnSNMP does not support the tables **rdbmsDbLimitedResourceTable** and **rdbmsDbParamTable**.

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.

Table 3-4. MIB objects for rdbmsDbInfoTable

MIB object	Description
rdbmsDbIndex	See "rdbmsDbTable."
rdbmsDbInfoProductName	Name of the database product. For example, this value might be IBM Informix.
rdbmsDbInfoVersion	Version number of the database server 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).

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.

Table 3-5. MIB objects for rdbmsDbTable

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 IBM Informix Private MIB: 1.3.6.1.4.1.893

Table 3-5. MIB objects for rdbmsDbTable (continued)

MIB object	Description
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 with which it is associated

The table has the following MIB objects.

Table 3-6. MIB objects for rdbmsRelTable

MIB object	Description
rdbmsDbIndex	See "rdbmsDbTable" on page 3-4.
applIndex	See "applTable" on page 3-2.
rdbmsRelState	Access state between the database server and the database: • Other (1): The database server is online, but one of the dbspaces of the database is down. • Active (2): The database server is actively using the database. The database server is online, and a user opened the database. • Available (3): The database server could use the database if asked to do so. The database server is online, but the database is not open. • Restricted (4): The database is not available. The database server is online, and a user opened the database in exclusive mode. • Unavailable (5)
rdbmsRelActiveTime	Date and time that the database server 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 since it was started

Index: applIndex

Scope of a row:
One database server

The table has the following MIB objects.

Table 3-7. MIB objects for rdbmsSrvInfoTable

MIB Object	Description
applIndex	See "applTable" on page 3-2.
rdbmsSrvInfoStartupTime	Date and time when the database server 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 has been unable to obtain the desired disk space
rdbmsSrvInfoHandledRequests	Number of requests made to the database server on inbound associations
rdbmsSrvInfoRequestRecvs	Number of receive operations that the database server made while it was processing requests on inbound associations
rdbmsSrvInfoRequestSends	Number of send operations that the database server made while it was processing requests on inbound associations
rdbmsSrvInfoHighwaterInbound-Associations	Greatest number of inbound associations that have been open at the same time
rdbmsSrvInfoMaxInbound-Associations	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

Index: applIndex, rdbmsSrvLimitedResourceName

Scope of a row:
One limited resource

The table has the following MIB objects.

Table 3-8. MIB objects for rdbmsSrvLimitedResourceTable

MIB Object	Description
applIndex	See "applTable" on page 3-2.

Table 3-8. MIB objects for rdbmsSrvLimitedResourceTable (continued)

MIB Object	Description
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 IBM Informix Private MIB: 1.3.6.1.4.1.893 or informix
rdbmsSrvLimitedResourceLimit	Maximum value that this limited resource can attain
rdbmsSrvLimitedResourceCurrent	The 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 tried to exceed the maximum value for this limited resource since **applUptime** was reset. This value is 0 for DBSPACES and CHUNKS.
rdbmsSrvLimitedResourceDescription	Description of the limited resource. This description includes the units for the value for the limited resource.

rdbmsSrvParamTable

The following list summarizes this table:

Contents:
Information about the configuration parameters for each database server

Index: applIndex, rdbmsSrvParamName, rdbmsSrvParamSubIndex

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

The **ONCONFIG** environment variable specifies the file name 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*.

Table 3-9. Location of the configure files

Operating system	Location of configuration file
UNIX	`$INFORMIXDIR/etc/$ONCONFIG`

Table 3-9. Location of the configure files (continued)

Operating system	Location of configuration file
Windows	%INFORMIXDIR%\etc\%ONCONFIG%

The table has the following MIB objects.

Table 3-10. MIB objects for rdbmsSrvParamTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
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 IBM 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

Index: applIndex

Scope of a row:
One database server

The table has the following MIB objects.

Table 3-11. MIB objects for rdbmsSrvTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
rdbmsSrvPrivateMibOID	OID for the IBM 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 IBM Informix.
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 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 IBM Informix database server becomes unavailable, it generates one trap for each database.

Online MIB in the Informix Private MIB

The Online MIB defines several tables that provide information that is relevant for IBM Informix database servers 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.

Table 3-12. MIB objects for onActiveBarTable

MIB Object	Description
applIndex	See "applTable" on page 3-2.
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)

Table 3-12. MIB objects for onActiveBarTable (continued)

MIB Object	Description
onActiveBarElapsedTime	Length of time since the activity started, in hundredths of seconds
onActiveBarActivitySize	Total number of used pages to scan OnSNMP updates this value as the activity progresses.
onActiveBarActivityScanned	Number of used pages that the activity has scanned so far
onActiveBarActivityCompleted	Number of scanned pages that the activity has transferred for archiving so far
onActiveBarActivityStatus	Status of the activity

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.

Table 3-13. MIB objects for onActiveTableTable

MIB Object	Description
applIndex	See "applTable" on page 3-2.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-4.
onTableIndex	See "onDbspaceTable" on page 3-13.
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

Table 3-13. MIB objects for onActiveTableTable (continued)

MIB Object	Description
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.

Table 3-14. MIB objects for onBarTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
onBarActivityIndex	Index to the history
onBarObjectIndex	Index to the object
onBarName	Name of the object
onBarType	Type of object: • blobspace (1) (Only IBM Informix 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 use

Index: applIndex, onDbspaceIndex, onChunkIndex

Scope of a row:
One chunk

The table has the following MIB objects.

Table 3-15. MIB objects for onChunkTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
onDbspaceIndex	See "rdbmsDbInfoTable" on page 3-3.
onChunkIndex	Unique integer index for this chunk The database server generates this value.
onChunkFileName	Path name 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	Path name 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).

Table 3-15. MIB objects for onChunkTable (continued)

MIB object	Description
onChunkMirrorStatus	Mirroring status: • offline (1) • online (2) • recovering (3) • inconsistent (4) • dropped (5)

If the chunk is not mirrored, this value is noSuchInstance (SNMPv2) or noSuchName (SNMPv1).

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 a database is active and whether one of its dbspaces is down.)

The table has the following MIB objects.

Table 3-16. MIB objects for onDatabaseTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-4.
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.

Table 3-17. MIB objects for onDbspaceTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
onDbspaceIndex	Unique integer index for this dbspace. The database server 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).

Table 3-17. MIB objects for onDbspaceTable (continued)

MIB object	Description
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.

Table 3-18. MIB objects for onErQueueTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
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.

Table 3-19. MIB objects for onErSiteTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
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

Table 3-19. MIB objects for onErSiteTable (continued)

MIB object	Description
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.

Table 3-20. MIB objects for onFragmentTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-4.
onTableIndex	See "onDbspaceTable" on page 3-13.
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

Table 3-20. MIB objects for onFragmentTable (continued)

MIB object	Description
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 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.

Table 3-21. MIB objects for onLockTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
onSessionIndex	See "onServerTable" on page 3-21.
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

Table 3-21. MIB objects for onLockTable (continued)

MIB object	Description
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.

Table 3-22. MIB objects for onLogicalLogTable

MIB Object	Description
applIndex	See "applTable" on page 3-2.
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

Table 3-22. MIB objects for onLogicalLogTable (continued)

MIB Object	Description
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
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.

Table 3-23. MIB objects for onPhysicalLogTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
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

Table 3-23. MIB objects for onPhysicalLogTable (continued)

MIB object	Description
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

Index: applIndex

Scope of a row:
One database server

The table has the following MIB objects.

Table 3-24. MIB objects for onServerTable

MIB Object	Description
applIndex	See "applTable" on page 3-2
onServerMode	Mode of the database server: • 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

Table 3-24. MIB objects for onServerTable (continued)

MIB Object	Description
onServerLockTimeouts	Number of deadlock time outs
onServerLogicalLogRecords	Number of logical-log records
onServerLogicalLogPageWrites	Number of logical-log page writes
onServerLogicalLogWrites	Number of logical-log writes
onServerBufferFlushes	Number of buffer flushes
onServerForegroundWrites	Number of foreground writes
onServerLRUWrites	Number of LRU writes
onServerChunkWrites	Number of chunk writes
onServerReadAheadPages	Number of read-ahead pages This value includes data and index read-ahead pages.
onServerReadAheadPagesUsed	Number of read-ahead pages used
onServerSequentialScans	Number of sequential scans
onServerMemorySorts	Number of memory sorts
onServerDiskSorts	Number of disk sorts
onServerMaxSortSpace	Maximum disk space that a sort uses, in pages
onServerNetworkReads	Number of network reads
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.

Table 3-25. MIB objects for onSessionTable

MIB Object	Description
applIndex	See "applTable" on page 3-2.
onSessionIndex	Unique integer index for the session
onSessionUserName	Name of the user, in the form name@host(tty)

Table 3-25. MIB objects for onSessionTable (continued)

MIB Object	Description
onSessionUserProgramVersion	Version of the database server
onSessionUserProcessId	Process ID for the session
onSessionUserTime	Length of time that the user has been connected to the database server, in hundredths of seconds
onSessionState	State of the session: • idle (1) • active (2) • waitingOnMutex (3) • waitingOnCondition (4) • waitingOnLock (5) • waitingOnBuffer (6) • waitingOnCheckPointing (7) • waitingOnLogicalLogWrite (8) • waitingOnTransaction (9)
onSessionDatabase	Connected database
onSessionCurrentMemory	Memory usage, in bytes
onSessionThreads	Number of active threads
onSessionCurrentStack	Average size of the stack for all threads
onSessionHighwaterStack	Maximum amount of memory that any thread has used so far
onSessionLockRequests	Number of lock requests
onSessionLocksHeld	Number of locks held
onSessionLockWaits	Number of lock waits
onSessionLockTimeouts	Number of timeouts 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)

Table 3-25. MIB objects for onSessionTable (continued)

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 selects 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).
onSessionOriginatingSessionId	Local session ID of the global session on the server 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.

Table 3-26. Location of connection information

Operating system	Location of connectivity information
UNIX	The **INFORMIXSQLHOSTS** environment variable specifies the full path name and file name of the connection information. The default location is `$INFORMIXDIR/etc/sqlhosts`. For information about **INFORMIXSQLHOSTS**, see the *IBM Informix Guide to SQL: Reference*.

Table 3-26. Location of connection information (continued)

Operating system	Location of connectivity information
Windows	The connectivity information is in a key in the Windows registry called `HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SQLHOSTS`.

For details about the connection information, see your *IBM Informix Administrator's Guide*.

The table has the following MIB objects.

Table 3-27. MIB objects for onSqlHostTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
onSqlHostIndex	Index to the entry in the connectivity information
onSqlHostName	Host name of the database server
onSqlHostNetType	Connection type
onSqlHostServerName	Name of the database server 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.

Table 3-28. MIB objects for onTableTable

MIB object	Description
applIndex	See "applTable" on page 3-2.
rdbmsDbIndex	See "rdbmsDbTable" on page 3-4.
onTableIndex	Table number This value is the same as **tabid** in the system catalog table **systables**
onTableName	Table name
onTableOwner	Table owner

Table 3-28. MIB objects for onTableTable (continued)

MIB object	Description
onTableType	Type of table: • table (1) • view (2) • privateSynonyn (3) • synonym (4)
onTableLockLevel	Locking level of the table: • page (1) • row (2)
onTableCreated	Creation date, in string format
onTableFirstDbspace	Name of the first (or only) dbspace for the table
onTableRowSize	Length of a row
onTableRows	Number of rows
onTableColumns	Number of columns
onTableIndices	Number of indexes
onTableExtents	Number of extents in use
onTablePagesAllocated	Total (extent) size allocated to the table, in pages
onTablePagesUsed	Number of pages in use
onTableFragments	Number of fragments
onTableFragmentStrategy	Fragmentation strategy: • roundRobin (1) • byExpression (2) • tableBased (3)

If the table is not fragmented, this value is noSuchInstance (SNMPv2) or noSuchName (SNMPv1).

onTableActiveFragments

Number of active fragments

If the table is not fragmented, this value is noSuchInstance (SNMPv2) or noSuchName (SNMPv1).

Appendix. Accessibility

IBM strives to provide products with usable access for everyone, regardless of age or ability.

Accessibility features for IBM Informix products

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in IBM Informix products. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers.
- The attachment of alternative input and output devices.

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

Related accessibility information

IBM is committed to making our documentation accessible to persons with disabilities. Our publications are available in HTML format so that they can be accessed with assistive technology such as screen reader software.

IBM and accessibility

See the *IBM Accessibility Center* at http://www.ibm.com/able for more information about the IBM commitment to accessibility.

Dotted decimal syntax diagrams

The syntax diagrams in our publications are available in dotted decimal format, which is an accessible format that is available only if you are using a screen reader.

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, that element 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 refers 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 can write `HOST STATE`, but you cannot 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 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.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other 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 grant 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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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 document 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 is for planning purposes only. The information herein is subject to change before the products described become available.

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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Index

M

O

P

R

S

T

U

V

Printed in USA

SC27-4533-00