# Logical Domains (LDoms) MIB 1.0.1 Administration Guide

For the Control Domain

Sun Microsystems, Inc.
www.sun.com

Part No. 820-2319-10
September 2007, Revision A

Submit comments about this document at: `http://www.sun.com/hwdocs/feedback`

# Contents

# Figures

# Tables

# Preface

This guide provides security, installation, configuration, usage, and removal information for system administrators who use the Logical Domains (LDoms) Management Information Base (MIB) 1.0.1 software on the control domain to monitor, receive notification changes, and start and stop logical domains on Sun UltraSPARC® T1– and T2–based platforms.

## Before You Read This Book

To fully use the information in this document, you must have thorough knowledge of the operation of Logical Domains software and the topics discussed in these books:

- *Beginners Guide to LDoms: Understanding and Deploying Logical Domains*
- *Logical Domains (LDoms) 1.0.1 Release Notes*
- *Logical Domains (LDoms) 1.0.1 Administration Guide*

You must know how to perform system administration operations in the Solaris 10 OS. You also must know how to use the Simple Network Management Protocol (SNMP) and must have an understanding of the SNMP Management Information Base (MIB) and the System Management Agent (SMA). You must know the following areas:

- SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2c), and SNMP version 3 (SNMPv3) protocols
- Structure of Management Information (SMI) version 1 and version 2
- Management Information Base (MIB) structure
- Abstract Syntax Notation (ASN.1)

# How This Book Is Organized

Chapter 1 provides an overview of the Logical Domains Management Information Base software.

Chapter 2 provides instructions for creating new SNMP version 3 users to provide secure access to SMA.

Chapter 3 provides instructions for installing, loading, and removing LDoms MIB software.

Chapter 4 describes how to monitor logical domains by querying the LDoms MIB using tables and scalar MIB variables.

Chapter 5 describes using the traps that the LDoms MIB generates and the active management operations the LDoms MIB provides.

Glossary is a list of abbreviations, acronyms, terms and their definitions specific to the LDoms and LDoms MIB software.

# Using UNIX Commands

This document might not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices. Refer to the following for this information:

- Software documentation that you received with your system
- Solaris™ Operating System documentation, which is at

  `http://docs.sun.com`

# Shell Prompts

| Shell | Prompt |
|---|---|
| C shell | *machine-name*`%` |
| C shell superuser | *machine-name*`#` |
| Bourne shell and Korn shell | `$` |
| Bourne shell and Korn shell superuser | `#` |

# Typographic Conventions

| Typeface* | Meaning | Examples |
|---|---|---|
| `AaBbCc123` | The names of commands, files, and directories; on-screen computer output | Edit your `.login` file.<br>Use `ls -a` to list all files.<br>`% You have mail.` |
| **`AaBbCc123`** | What you type, when contrasted with on-screen computer output | `%` **`su`**<br>`Password:` |
| *AaBbCc123* | Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values. | Read Chapter 6 in the *User's Guide.*<br>These are called *class* options.<br>To delete a file, type **`rm`** *filename.* |

* **The settings on your browser might differ from these settings.**

# Related Documentation

The *Beginners Guide to LDoms: Understanding and Deploying Logical Domains Software* can be found at the Sun BluePrints™ site at:

`http://www.sun.com/blueprints/0207/820-0832.html`

The other Logical Domains (LDoms) and Logical Domains Management Information Base (MIB) 1.0.1 documentation is available at:

`http://docs.sun.com`

You can find documents relating to your server or your Solaris OS at:

`http://www.sun.com/documentation/`

| Application | Title | Part Number | Format | Location |
|---|---|---|---|---|
| Release notes for LDoms MIB | *Logical Domains (LDoms) Management Information Base (MIB) 1.0.1 Release Notes* | 820-2320-10 | HTML<br>PDF | Online |
| Release notes for LDoms | *Logical Domains (LDoms) 1.0.1 Release Notes* | 819-6429-12 | HTML<br>PDF | Online |
| Basics for Logical Domains software | *Beginners Guide to LDoms: Understanding and Deploying Logical Domains Software* | 820-0832 | PDF | Online |
| Administration for LDoms | *Logical Domains (LDoms) 1.0.1 Administration Guide* | 819-6428-12 | HTML<br>PDF | Online |
| System Management Agent | *Solaris System Management Agent Administration Guide* | 819-6813-01 | HTML<br>PDF | Online |

# Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation`)
- Support (`http://www.sun.com/support`)
- Training (`http://www.sun.com/training`)

# Third-Party Web Sites

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection width the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

# Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

```
http://www.sun.com/hwdocs/feedback
```

Please include the title and part number of your document with your feedback:

*Logical Domains (LDoms) MIB 1.0.1 Administration Guide*, part number 820-2319-10.

CHAPTER 1

# Overview

The Logical Domains (LDoms) Management Information Base (MIB) enables third party system management applications to perform remote monitoring and starting and stopping logical domains using the Simple Network Management Protocol (SNMP).

The LDoms MIB software runs on the control domain only, and only one instance of the LDoms MIB can be run. You must use Solaris 10 11/06 at a minimum, and the Logical Domains 1.0.1 software at a minimum.

This chapter contains the following sections:

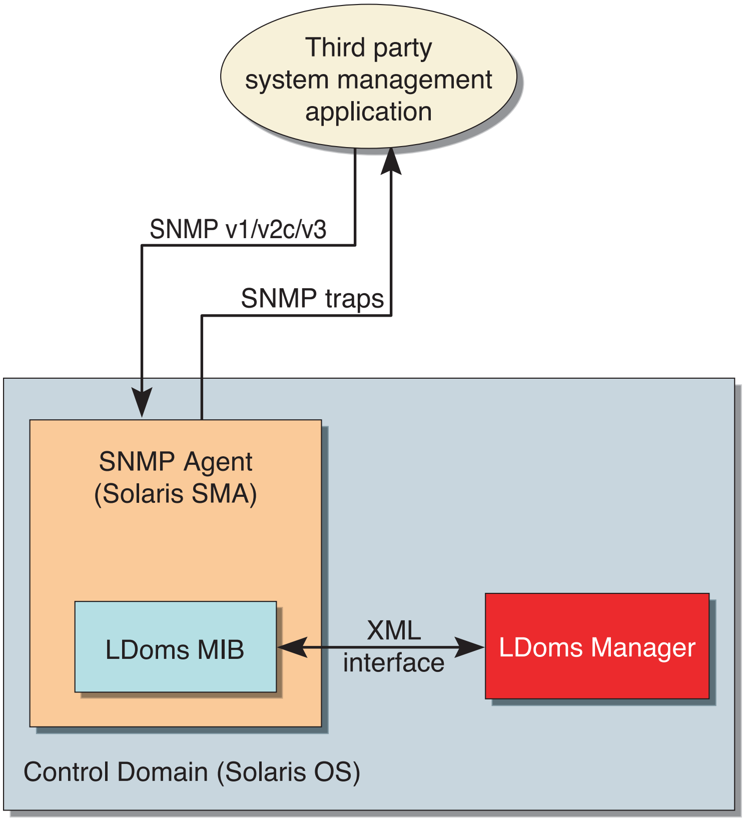
## Software Components

The LDoms MIB package (SUNWldmib.v) contains the following software components:

- SUN-LDOM-MIB.mib – An SNMP MIB in the form of a text file that defines the objects in the LDoms MIB.
- ldomMIB.so – An SMA extension module in the form of a shared library that enables the SMA to respond to requests for information specified in the LDoms MIB. This module also generates traps.

FIGURE 1-1 shows the interaction between the LDoms MIB, the SMA, the LDoms Manager, and a third party system management application.

**FIGURE 1-1** LDoms MIB Interaction With SMA, LDoms Manager, and Third Party System Management Application



# System Management Agent

The SNMP agent listens on the standard SNMP port (161) for requests from a third party system management application to get or set data offered by the LDoms MIB The SNMP agent can also emit traps to the configured system management application using the standard port for SNMP notifications (162). The LDoms MIB is exported by the Solaris OS default System Management Agent (SMA) on the control domain.

The SMA supports the get, set, and trap functions of SNMP versions `v1`, `v2c`, and `v3`. Most LDoms MIB objects are read-only for monitoring purposes, but starting and stopping a logical domain is supported using a write of the `ldomAdminState` property of the `ldomTable`. See TABLE 4-1 for more information about this property and table.

# Logical Domains Manager and the LDoms MIB

A logical domain is a container that consists of a set of virtual resources for a guest operating system. The Logical Domains Manager provides the command-line interface (CLI) for creating, configuring, and managing the logical domains.

The Logical Domains Manager and the LDoms MIB support the following virtual resources:

- Virtual CPUs
- Memory
- Input/Output, including disk, network, and console
- Cryptographic units

## Parsing XML-Based Control Interface

The Logical Domains Manager exports an XML-based control interface to the LDoms MIB. The LDoms MIB parses the XML interface and populates the MIB. The LDoms MIB provides support only for the control domain.

## Providing Traps

The LDoms Manager 1.0.1 software does not provide event information when there is a change in a logical domain's state, such as add, remove, start, or stop. So the LDoms MIB polls the Logical Domains Manager periodically for updates or status change and issues SNMP traps to the system management applications.

## Providing Fault and Recovery Information

If the LDoms MIB can no longer allocate a needed resource, it returns a general error to the system management application through the SNMP agent. The SNMP delivery mechanism for traps does not provide confirmation. No specific state or checkpointing is implemented in the LDoms MIB. The SMA with the LDoms MIB is started and monitored by the `init` process and the Service Management Facility (SMF). When the SMA fails and exits, the SMF restarts the process automatically. Once restarted, the new process dynamically restarts the LDoms MIB module.

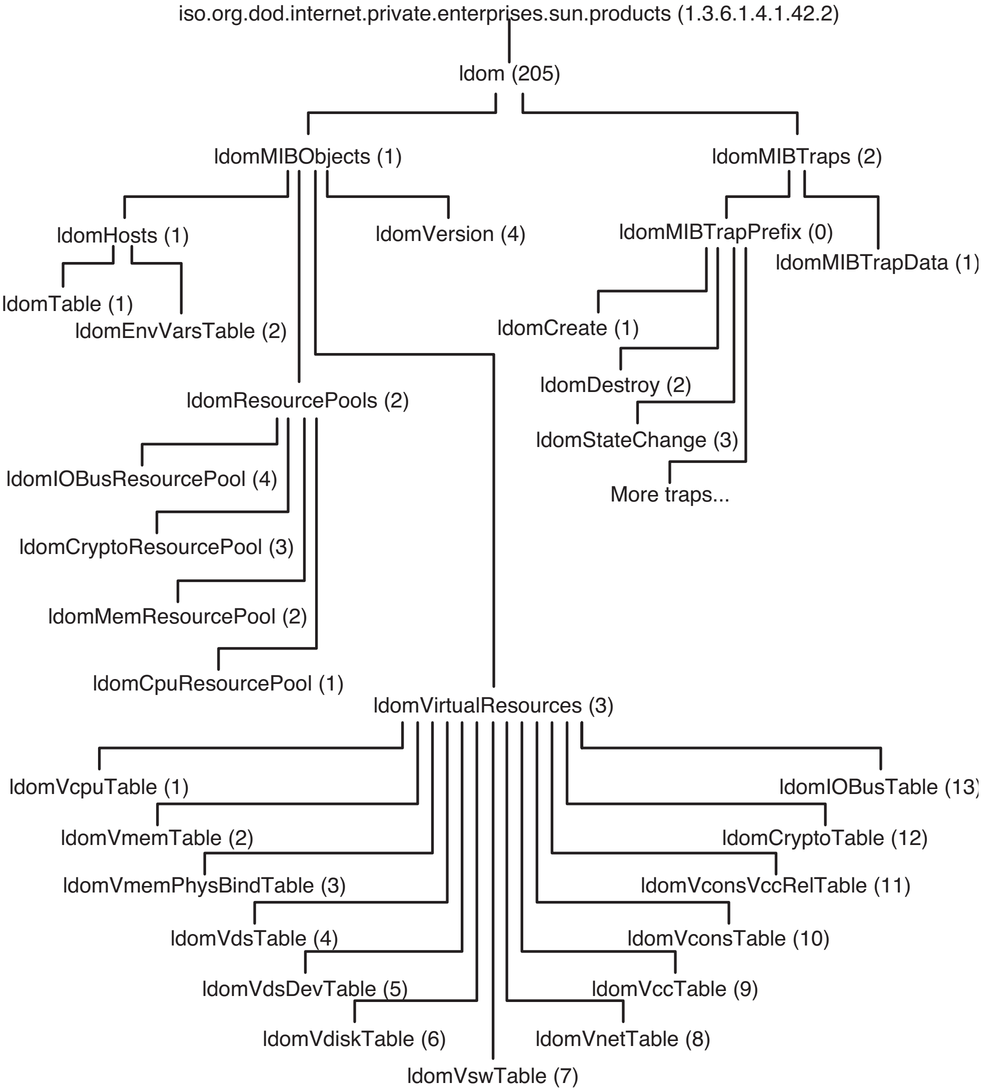
# LDoms MIB Object Tree

SNMP-managed objects are organized into a tree-like hierarchy. An object identifier (OID) consists of a series of integers based on the nodes in the tree, separated by dots. Each managed object has a numerical OID and an associated textual name. The LDoms MIB is registered as the `ldom (205)` branch under:

```
iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).
sun(42).products(2)
```

FIGURE 1-2 shows the major subtrees under the LDoms MIB.

**FIGURE 1-2** LDoms MIB Tree



CHAPTER 2

# Managing Security

This chapter provides information about how to create new Simple Network Management Protocol (SNMP) version 3 (v3) users to provide secure access to SMA. The access control mechanism is the *community_string* for SNMP version 1 (v1) and version 2 (v2c). Refer to Chapter 2 of the *Logical Domains (LDoms) 1.0.1 Administration Guide* for complete information about providing security for logical domains.

**Note –** Creating snmpv3 users is for using the System Management Agent (SMA) in SNMP with the LDoms MIB. This type of user in no way interacts with or conflicts with users you might have set up using the Solaris OS Role-Based Access control (RBAC) adapted for the Logical Domains Manager.

## Creating SNMP Version 3 (v3) Users

This section explains how to create the first new snmpv3 user. See "To Create the First New snmpv3 User" on page 8. Additional users are cloned from this initial user, so that the initial user's authentication and security types can be inherited. These types can be changed later. You set secret key data for the user when you clone from the initial user. You must know the passwords for the initial user and later users that you set up. You can only clone one user at a time from the initial user. For more information, refer to the *Solaris System Management Agent Administration Guide* for your version of the Solaris OS.

## ▼ To Create the First New snmpv3 User

This procedure describes using the `net-snmp-config` utility in SMA to create the first new user. The utility adds a line to the `/etc/sma/snmp/snmpd.conf` file, giving the initial user read and write access to the agent.

1. **Stop the System Management Agent.**

```
# svcadm disable -t svc:/application/management/sma:default
```

2. **Use the** `net-snmp-config` **utility to create the new user.**

```
# /usr/sfw/bin/net-snmp-config --create-snmpv3-user -a my_password newuser
```

   This command causes a new user to be created, named *newuser*, with a password you choose.

---

**Note –** Passwords must contain at least eight characters.

---

3. **Start the System Management Agent.**

```
# svcadm enable svc:/application/management/sma:default
```

4. **Check whether the new user (*newuser*) exists.**

```
# snmpget -v 3 -u newuser -l authNoPriv -a MD5 -A my_password localhost sysUpTime.0
```

## Creating Additional snmpv3 Users

The preferred method of creating additional new users in secure SNMP is to clone the initial user that you originally set up. Refer to Chapter 4 of the Solaris System Management Agent Administration Guide for your version of the Solaris OS. The procedure in that document “To Create Additional SNMPv3 Users with Security” copies the user you set up in “To Create the First New snmpv3 User” on page 8.

CHAPTER 3

# Installing, Loading, and Removing the LDoms MIB

This chapter describes how to perform the following procedures:

- "Installing the LDoms MIB Package" on page 9
- "Loading the LDoms MIB Module" on page 10
- "Removing the LDoms MIB Package" on page 11
- "Administering the SNMP System" on page 11

## Installing the LDoms MIB Package

The LDoms MIB software package (`SUNWldmib`.v) is bundled with the LDoms 1.0.1 software package, which can be found at the Logical Domains 1.0.1 software download site at:

```
http://www.sun.com/ldoms
```

After you have downloaded and installed the LDoms 1.0.1 software, you can install the LDoms MIB software package in the control domain, named `primary`. You can find the package under the directory `LDoms_Manager-1_0_1/Product/`. For more information about installing the LDoms 1.0.1 software, refer to the *Logical Domains (LDoms) 1.0.1 Administration Guide*.

## ▼ To Install the LDoms MIB Software Package

● **Use the** `pkgadd`**(1M) command to add** `SUNWldmib.v` **to the control domain.**

```
# pkgadd -d . SUNWldmib.v
```

This command installs the following files:

- `/opt/SUNWldmib/lib/mibs/SUN-LDOM-MIB.mib`
- `/opt/SUNWldmib/lib/ldomMIB.so`

# Loading the LDoms MIB Module

The LDoms MIB module (`ldomMIB.so`) must be loaded into the Solaris System Management Agent (SMA) to query the LDoms MIB. The LDoms MIB module is dynamically loaded; that is, the module is included within the SMA agent without having to recompile and relink the agent binary. The following procedure shows how to do this. Instructions for dynamically loading a module without restarting the SMA are provided in the *Solaris System Management Agent Developer's Guide*. For more information about SMA, refer to the *Solaris System Management Administration Guide*.

## ▼ To Load the LDoms MIB Module Into SMA

1. **Add the following line to the SMA SNMP configuration file (**`/etc/sma/snmp/snmpd.conf`**):**

```
dlmod ldomMIB /opt/SUNWldmib/lib/ldomMIB.so
```

2. **Restart the SMA.**

```
# svcadm restart svc:/application/management/sma:default
```

# Removing the LDoms MIB Package

## ▼ To Remove the LDoms MIB Software Package

1. **Stop the System Management Agent.**

```
# svcadm disable svc:/application/management/sma:default
```

2. **Use the** `pkgrm`**(1M) command to remove the LDoms MIB software package (**`SUNWldmib.v`**) from the control domain.**

```
# pkgrm SUNWldmib
```

3. **Remove the following line from the** `/etc/sma/snmp/snmpd.conf` **file (that was previously added):**

```
dlmod ldomMIB /opt/SUNWldmib/lib/ldomMIB.so
```

4. **Restart SMA.**

```
# svcadm restart svc:/application/management/sma:default
```

# Administering the SNMP System

For more information about SNMP system administration, refer to the `snmpd.conf`(1M) man page.

CHAPTER 4

# Monitoring Logical Domains

This chapter tells how to monitor logical domains by querying the LDoms MIB using tables and scalar MIB variables, and provides descriptions of the various types of output.

**Note –** Be sure that you have added the `ldom` prefix to table names, objects in the table, and scalar object names when using the SNMP commands to retrieve data from the LDoms MIB.

This chapter contains the following sections:

## Setting Environmental Variables

Before you can query the LDoms MIB, you must set environmental variables depending on the UNIX shell you use:

- C shell
- Bourne (`sh`) or Korn (`ksh`) shell

### ▼ To Set Environmental Variables for C Shell Users

- **Set the following environment variables:**

```
setenv PATH /usr/sfw/bin:$PATH
setenv MIBDIRS /opt/SUNWldmib/lib/mibs:/etc/sma/snmp/mibs
setenv MIBS +SUN-LDOM-MIB
```

### ▼ To Set Environmental Variables for Bourne or Korn Shell Users

- **Set the following environment variables:**

```
PATH=/usr/sfw/bin:$PATH; export PATH
MIBDIRS=/opt/SUNWldmib/lib/mibs:/etc/sma/snmp/mibs; export MIBDIRS
MIBS=+SUN-LDOM-MIB; export MIBS
```

## Querying the LDoms MIB

This section shows examples of how to query the LDoms MIB using SNMP commands.

### ▼ To Retrieve a Single MIB Object

- **Retrieve a single MIB object using the** `snmpget`**(1M) command:**

```
% snmpget -v version -c community_string host MIB_object
```

For example, the following `snmpget`(1M) command queries the value for the `ldomVersionMajor` object using `snmpv1` (`-v1`) and *community_string* `public` (`-c`) on the `localhost` and returns the following output.

```
% snmpget -v1 -c public localhost SUN-LDOM-MIB::ldomVersionMajor.0
SUN-LDOM-MIB::ldomVersionMajor.0 = INTEGER: 1
```

## ▼ To Retrieve an Array of MIB Objects

- **Use either of these Solaris 10 OS commands to retrieve an array of MIB objects:**
  - `snmpwalk`(1M)
  - `snmptable`(1M)

## Examples of Retrieving an Array of MIB Objects

Following are examples of using the `snmpwalk`(1M) and the `snmptable`(1M) commands to retrieve an array of MIB objects.

### Example Using the `snmpwalk`(1M) Command for SNMP Version 1 (`v1`)

For example, using the following `snmpwalk`(1M) command on the `ldomTable` returns the values for all objects in the table:

```
% snmpwalk -v1 -c public localhost SUN-LDOM-MIB::ldomTable
SUN-LDOM-MIB::ldomName.1 = STRING: primary
SUN-LDOM-MIB::ldomName.2 = STRING: LdomMibTest_1
SUN-LDOM-MIB::ldomAdminState.1 = INTEGER: 0
SUN-LDOM-MIB::ldomAdminState.2 = INTEGER: 0
SUN-LDOM-MIB::ldomOperState.1 = INTEGER: active(1)
SUN-LDOM-MIB::ldomOperState.2 = INTEGER: inactive(3)
SUN-LDOM-MIB::ldomNumVCpu.1 = INTEGER: 32
SUN-LDOM-MIB::ldomNumVCpu.2 = INTEGER: 2
SUN-LDOM-MIB::ldomMemSize.1 = INTEGER: 3968
SUN-LDOM-MIB::ldomMemSize.2 = INTEGER: 256
SUN-LDOM-MIB::ldomMemUnit.1 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomMemUnit.2 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomNumCrypto.1 = INTEGER: 8
SUN-LDOM-MIB::ldomNumCrypto.2 = INTEGER: 0
SUN-LDOM-MIB::ldomNumIOBus.1 = INTEGER: 2
SUN-LDOM-MIB::ldomNumIOBus.2 = INTEGER: 0
```

## Examples Using the `snmpwalk`(1M) Command for SNMP Version 2 (`v2c`) and Version 3 (`v3`)

The following `snmpwalk`(1M) commands are used to retrieve the contents of `ldomTable` using `snmpv2c` and `snmpv3`.

```
% snmpwalk -v2c -c public localhost SUN-LDOM-MIB::ldomTable

% snmpwalk -v 3 -u test -l authNoPriv -a MD5 -A testpassword localhost SUN-LDOM-
MIB::ldomTable
```

## Example Using the `snmptable`(1M) Command for SNMP Version 1 (`v1`)

The `snmptable`(1M) command displays the contents of a table in tabular form. For example:

```
% snmptable -v1 -c public localhost SUN-LDOM-MIB::ldomTable

       ldomName ldomAdminState ldomOperState ldomNumVCpu ldomMemSize
ldomMemUnit ldomNumCrypto ldomNumIOBus
       primary               0        active          32        3968
megabytes               8            2
 LdomMibTest_1               0      inactive           2         256
megabytes               0            0
```

## Example Using the `snmptable`(1M) Command for SNMP Version 2 (`v2c`)

The following `snmptable`(1M) command is used to display the contents of a table in tabular form using `snmpv2c`.

```
% snmptable -v2c -CB -c public localhost SUN-LDOM-MIB::ldomTable
```

**Note –** For the `v2c` or `v3` `snmptable` command, use the `-CB` option to use only `GETNEXT`, *not* `GETBULK`, requests to retrieve data.

# Retrieving LDoms MIB Information

This section shows the information you can retrieve from the Logical Domains MIB. The information can come in the form of tables or scalar objects. Both are shown and identified in this section.

## Logical Domains Table (`ldomTable`)

The following table summarizes each logical domain in the system, including the resource constraints for virtual CPUs, memory, cryptographic units, and I/O buses.

**TABLE 4-1** Logical Domains (`ldomTable`) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| `ldomIndex` | Integer | Not accessible | Integer used as an index of the table |
| `ldomName` | Display string | Read-only | Name of the logical domain |
| `ldomAdminState` | Integer | Read/Write | **Write** -<br>1 (Active) starts the domain<br>2 (Stopping) stops the domain |
| `ldomOperState` | Integer | Read-only | Current actual state of the logical domain:<br>1=Active<br>2=Stopping<br>3=Inactive<br>4=Binding<br>5=Unbinding<br>6=Bound<br>7=Starting |
| `ldomNumVCPU` | Integer | Read-only | Number of virtual CPUs used. If the logical domain is in a inactive state, then this is the requested number of virtual CPUs. |
| `ldomMemSize` | Integer | Read-only | Amount of virtual memory used. If the logical domain is in a inactive state, then this is the requested memory size. |

**TABLE 4-1** Logical Domains (ldomTable) Table *(Continued)*

| Name | Data Type | Access | Description |
|---|---|---|---|
| ldomMemUnit | Integer | Read-only | Memory unit:<br>1=KB (kilobytes)<br>2=MB (megabytes)<br>3=GB (gigabytes)<br>4=(bytes)<br>Default is bytes if the memory unit is missing. |
| ldomNumCrypto | Integer | Read-only | Number of cryptographic (crypto) units used. If the logical domain is in a inactive state, then this is the requested number of crypto units. |
| ldomNumIOBus | Integer | Read-only | Number of physical I/O devices used |

# LDoms Resource Pool and Scalar Variables

The following resources can be assigned to logical domains:

- Virtual CPU (vcpu)
- Memory (mem)
- Cryptographic unit (mau)
- Virtual switch (vsw)
- Virtual network (vnet)
- Virtual disk server (vds)
- Virtual disk server device (vdsdev)
- Virtual disk (vdisk)
- Virtual console concentrator (vcc)
- Virtual console (vcons)
- Physical I/O device (io)

Following are the scalar MIB variables that are used to represent these resource pools and their properties.

**TABLE 4-2** Scalar Variables for CPU Resource Pool

| Name | Data Type | Access | Description |
|---|---|---|---|
| `CpuRpCapacity` | Integer | Read-only | Maximum reservation allowed by the pool, in `CpuRpCapacityUnit` |
| `CpuRpReserved` | Integer | Read-only | Quantity of resources from pool in `CpuRpReservedUnit`, which are currently reserved |
| `CpuRpCapacityUnit` and<br>`CpuRpReservedUnit` | Integer | Read-only | Various CPU allocation units:<br>1=MHz (megahertz)<br>2=GHz (gigahertz)<br>Set to 1(MHz) |

**TABLE 4-3** Scalar Variables for Memory Resource Pool

| Name | Data Type | Access | Description |
|---|---|---|---|
| `MemRpCapacity` | Integer | Read-only | Maximum reservation allowed by the pool, in `MemRpCapacityUnit` |
| `MemRpReserved` | Integer | Read-only | Quantity of resources from pool in `MemRpReservedUnit`, which are currently reserved |
| `MemRpCapacityUnit` and<br>`MemRpReservedUnit` | Integer | Read-only | Various memory allocation units:<br>1=KB (kilobytes)<br>2=MB (megabytes)<br>3=GB (gigabytes)<br>4= (bytes)<br>Default set to 4 (bytes) |

**TABLE 4-4** Scalar Variables for Crypto Resource Pool

| Name | Data Type | Access | Description |
|---|---|---|---|
| `CryptoRpCapacity` | Integer | Read-only | Maximum reservation allowed by the pool |
| `CryptoRpReserved` | Integer | Read-only | Quantity of resources from the pool, which are currently reserved |

**TABLE 4-5** Scalar Variables for I/O Bus Resource Pool

| Name | Data Type | Access | Description |
|---|---|---|---|
| `IOBusRpCapacity` | Integer | Read-only | Maximum reservation allowed by the pool |
| `IOBusRpReserved` | Integer | Read-only | Quantity of resources from the pool that are currently reserved |

## Virtual CPU Table (VcpuTable)

This table describes virtual CPUs belonging to all logical domains.

**TABLE 4-6** Virtual CPU (VcpuTable) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| VcpuLdomIndex | Integer | Read-only | An integer used to index to the ldomTable representing the logical domain containing the virtual CPU (vcpu) |
| VcpuIndex | Integer | Not accessible | An integer used to index the vcpu in this table |
| VcpuDeviceID | Display string | Read-only | Identifier of the virtual CPU |

**TABLE 4-6** Virtual CPU (`VcpuTable`) Table *(Continued)*

| Name | Data Type | Access | Description |
|---|---|---|---|
| `VcpuOperationalStatus` | Integer | Read-only | Various CPU statuses:<br>1=Unknown<br>2=Other<br>3=OK<br>4=Degraded<br>5=Stressed<br>6=Predictive Failure<br>7=Error<br>8=Nonrecoverable error<br>9=Starting<br>10=Stopping<br>11=Stopped<br>12=In service<br>13=No contact<br>14=Lost communication<br>15=Aborted<br>16=Dormant<br>17=Supporting entity in error<br>18=Completed<br>19=Power mode<br>Set to 1 (Unknown), because the Logical Domains Manager does not provide the CPU state. |
| `VcpuPhysBind` | Display string | Read-only | Physical binding. Contains identifier of strand (hardware thread) that is assigned to this virtual CPU. This ID also uniquely identifies the core and chip. |
| `VcpuPhysBindUsage` | Integer | Read-only | Indicates how many megahertz of the total capacity of the strand is used for this virtual CPU. For example, if a thread can run at a maximum of one gigahertz, and only half of that is allocated to this virtual CPU (that is, 50% of the strand), this field would have the value of 500. |

## Virtual Memory Tables

Memory space seen in a logical domain is referred to as *real* memory; that is, virtual memory. Host platform memory space seen by the hypervisor is referred to as *physical* memory. The hypervisor maps blocks of physical memory to form a block of real memory which is used by a logical domain. For example, a logical domain that asks for 521 megabytes of real memory can be assigned by two blocks of 256 megabytes in the host system as physical memory in the format {physical-address, real-address, size}:

{0x1000000, 0x1000000, 256}, {0x2000000, 0x2000000,256}

Because a logical domain can have up to 64 physical memory segments assigned to a guest, an auxiliary table is used to hold each memory segment instead of using the display string which has a 255-character limit.

## Virtual Memory Table (`VmemTable`)

This table describes properties of virtual memory belonging to logical domains.

**TABLE 4-7** Virtual Memory (`VmemTable`) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| `VmemLdomIndex` | Integer | Read-only | An integer used to index to the `ldomTable` representing the logical domain containing the virtual memory |
| `VmemIndex` | Integer | Not accessible | An integer used to index virtual memory in this table |
| `VmemNumberofBlocks` | Integer | Read-only | Number of blocks of virtual memory |

## Virtual Memory Physical Binding Table (`VmemPhysBind`)

This is an auxiliary table containing physical memory segments for all logical domains.

**TABLE 4-8** Virtual Memory Physical Binding (`VmemPhysBind`) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| `VmemPhysBindLdomIndex` | Integer | Read-only | An integer used to index to the `ldomTable` representing the logical domain containing physical memory segments |
| `VmemPhysBind` | Display string | Read-only | List of physical memory in the format {physical-address, real-address, size} that is mapped to this virtual memory block |

# Virtual Disk Tables

A virtual disk service (vds) and the physical device to which it maps (vdsdev) provides the virtual disk capability to the Logical Domains 1.0.1 technology. A virtual disk service exports a number of local volumes (physical disk or file system). When a virtual disk service is specified, included are the complete /dev path of the backing device (vdsdev) and a unique name (volume name) for the device being added to the service.

One or more disks (or disk slices) or file systems can be bound to a single disk service. Each disk has a unique name and volume name. This volume name is used when the disk is bound to the service. The Logical Domains Manager creates virtual disk clients (vdisk) from the virtual disk service and its logical volumes.

## Virtual Disk Service (Vds) Table

This table describes virtual disk services for all logical domains.

**TABLE 4-9** Virtual Disk Service (Vds) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| VdsLdomIndex | Integer | Read-only | An integer used to index to the ldomTable representing the logical domain containing the virtual disk service |
| VdsIndex | Integer | Not accessible | An integer used to index vds in this table |
| VdsServiceName | Display string | Read-only | Service name for the virtual disk service. This is the *service_name* of the ldm add-vds command. |
| VdsNumofAvailVolume | Integer | Read-only | Number of logical volumes exported by this virtual disk service |
| VdsNumofUsedVolume | Integer | Read-only | Number of logical volumes used (bound) to this virtual disk service |

## Virtual Disk Service Device (VdsDev) Table

This table describes virtual disk service devices belonging to all virtual disk services.

**TABLE 4-10** Virtual Disk Service Device (VdsDev) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| VdsdevVdsIndex | Integer | Read-only | An integer used to index to the Virtual Disk Service Table representing the virtual disk service containing the virtual disk device |
| VdsdevIndex | Integer | Not accessible | An integer used to index the virtual disk service device in this table |
| VdsdevVolumeName | Display string | Read-only | Volume name for the virtual disk service device. This is a unique name for the device being added to the virtual disk service. This name is exported by this virtual disk service to the clients for adding. This is the volume_name property in the ldm add-vdsdev command. |
| VdsdevDevPath | Display string | Read-only | Path name of the disk device. This is the *device* property in the ldm add-vdsdev command. |

## Virtual Disk (Vdisk) Table

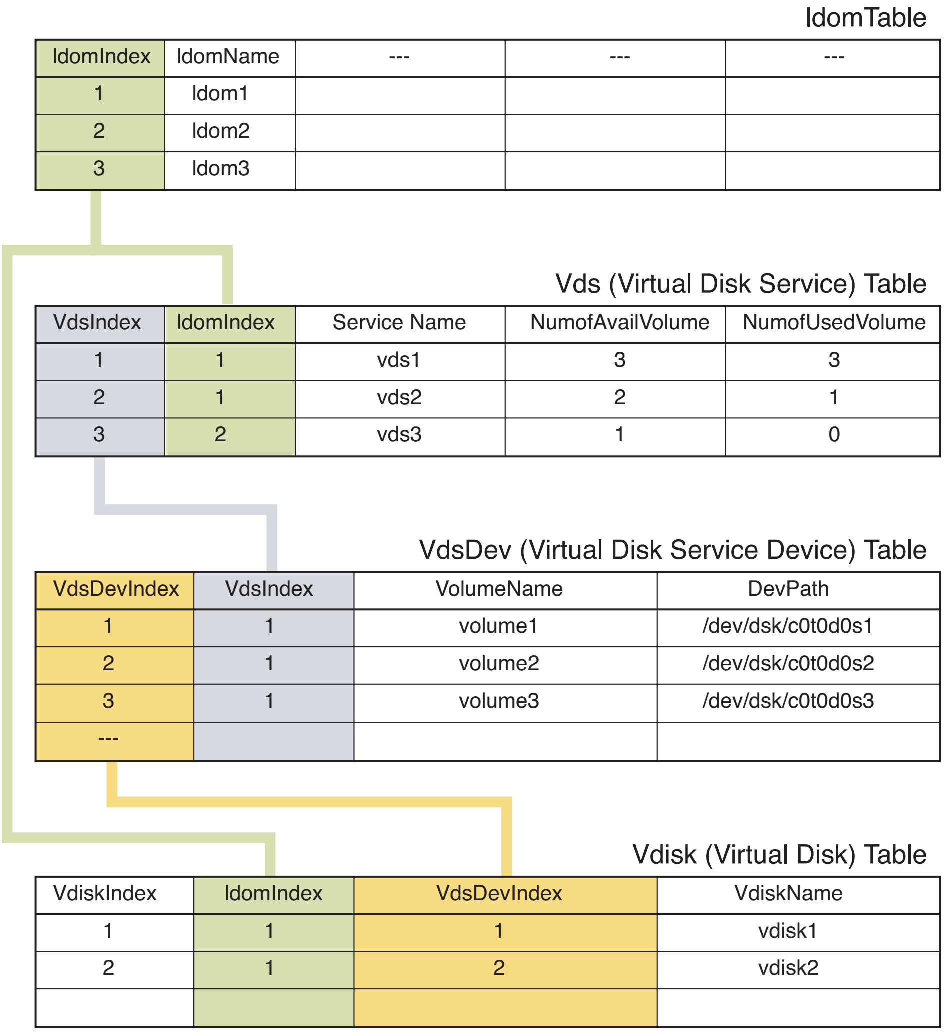
This table describes virtual disks for all logical domains.

**TABLE 4-11** Virtual Disk (Vdisk) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| VdiskLdomIndex | Integer | Read-only | An integer used to index to the ldomTable |
| VdiskVdsDevIndex | Integer | Read-only | An integer used to index to the Virtual Disk Service Device Table representing the virtual disk service device |
| VdiskIndex | Integer | Not accessible | An integer used to index the virtual disk in this table |
| VdiskName | Display string | Read-only | Name of the virtual disk. This is the *disk-name* property in the ldm add-vdisk command. |

The following diagram shows an example of the relationships among the virtual disk tables and the Logical Domains Table.

**FIGURE 4-1** Relationship Among Virtual Disk Tables and the Logical Domain Table



ldomTable

| ldomIndex | ldomName | --- | --- | --- |
|---|---|---|---|---|
| 1 | ldom1 | | | |
| 2 | ldom2 | | | |
| 3 | ldom3 | | | |

Vds (Virtual Disk Service) Table

| VdsIndex | ldomIndex | Service Name | NumofAvailVolume | NumofUsedVolume |
|---|---|---|---|---|
| 1 | 1 | vds1 | 3 | 3 |
| 2 | 1 | vds2 | 2 | 1 |
| 3 | 2 | vds3 | 1 | 0 |

VdsDev (Virtual Disk Service Device) Table

| VdsDevIndex | VdsIndex | VolumeName | DevPath |
|---|---|---|---|
| 1 | 1 | volume1 | /dev/dsk/c0t0d0s1 |
| 2 | 1 | volume2 | /dev/dsk/c0t0d0s2 |
| 3 | 1 | volume3 | /dev/dsk/c0t0d0s3 |
| --- | | | |

Vdisk (Virtual Disk) Table

| VdiskIndex | ldomIndex | VdsDevIndex | VdiskName |
|---|---|---|---|
| 1 | 1 | 1 | vdisk1 |
| 2 | 1 | 2 | vdisk2 |
| | | | |

## Virtual Network Tables

The virtual network support in Logical Domains software enables guest domains to communicate with each other and with external hosts through a physical Ethernet device. The virtual network contains two main components:

- Virtual switch (`vsw`)
- Virtual network device (`vnet`)

After you create a virtual switch on a service logical domain, you can bind a physical network device to the virtual switch. After that, you can create a virtual network device for a logical domain that uses the virtual switch service to communicate with other logical domains (connecting to the same virtual switch) and to the outside world (if a physical device is bound to the virtual switch).

## Virtual Switch Service (`Vsw`) Table

This table describes virtual switch services for all logical domains.

**TABLE 4-12** Virtual Switch Service (`Vsw`) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| `VswLdomIndex` | Integer | Read-only | An integer used to index to the `ldomTable` representing the logical domain containing the virtual switch service |
| `VswIndex` | Integer | Not accessible | An integer used to index `vsw` in this table |
| `VswServiceName` | Display string | Read-only | Virtual switch service name |
| `VswMacAddress` | Display string | Read-only | MAC address used by the virtual switch |
| `VswPhysDevPath` | Display string | Read-only | Physical device path for the virtual network switch. This is null when no physical device is bound to the virtual switch. |
| `VswMode` | Integer | Read-only | Mode of how the switch operates:<br>1=Switched (same as Programmed)<br>2=Promiscuous<br>3=SwitchedAndPromiscuous (same as Programmed/Promiscuous)<br>4=Routed |

## Virtual Network Device (Vnet) Table

This table describes the virtual network devices for all logical domains.

**TABLE 4-13** Virtual Network Device (Vnet) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| VnetLdomIndex | Integer | Read-only | An integer used to index to the ldomTable representing the logical domain containing the virtual network |
| VnetVswIndex | Integer | Read-only | An integer used to index to the virtual switch service table |
| VnetIndex | Integer | Not accessible | An integer used to index vnet in this table |
| VnetDevName | Display string | Read-only | Virtual network device name. This is the *net-dev* property in the ldm add-vnet command. |
| VnetDevMacAddress | Display string | Read-only | MAC address for this network device. This is the *mac-addr* property in the ldm add-vnet command. |

# Virtual Console Tables

The Logical Domains service domain provides a virtual network terminal service (vNTS). The virtual network terminal service provides a virtual console service, called a virtual console concentrator (vcc), with a range of port numbers. Each virtual console concentrator has multiple console groups (vcons), and each group is assigned a port number. Each group can contain multiple logical domains.

## Virtual Console Concentrator (Vcc) Table

This table describes virtual console concentrators for all logical domains.

**TABLE 4-14** Virtual Console Concentrator (Vcc) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| VccLdomIndex | Index | Read-only | An integer used to index to the ldomTable representing the logical domain containing the virtual console service |
| VccIndex | Integer | Not accessible | An integer used to index vcc in this table |

**TABLE 4-14** Virtual Console Concentrator (Vcc) Table *(Continued)*

| Name | Data Type | Access | Description |
|---|---|---|---|
| VccName | Display string | Read-only | Virtual console concentrator name. This is the *vcc_name* property in the ldm add-vcc command. |
| VccPortRangeLow | Integer | Read-only | Low number for the range of TCP ports to be used by the virtual console concentrator. This is the *port-range x* property in the ldm add-vcc command. |
| VccPortRangeHigh | Integer | Read-only | High number for the range of TCP ports to be used by the virtual console concentrator. This is the *port-range y* property in the ldm add-vcc command. |

## Virtual Console Group (Vcons) Table

This table describes virtual console groups for all virtual console services.

**TABLE 4-15** Virtual Console Group (Vcons) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| VconsIndex | Integer | Not accessible | An integer used to index a virtual group in this table |
| VconsGroupName | Display string | Read-only | Group name to which to attach the virtual console. This is the *group* property in the ldm set-vcons command. |
| VconsPortNumber | Integer | Read-only | Port number assigned to this group. This is the *port* property in the ldm set-vcons command. |

## Virtual Console Relationship (VconsVccRel) Table

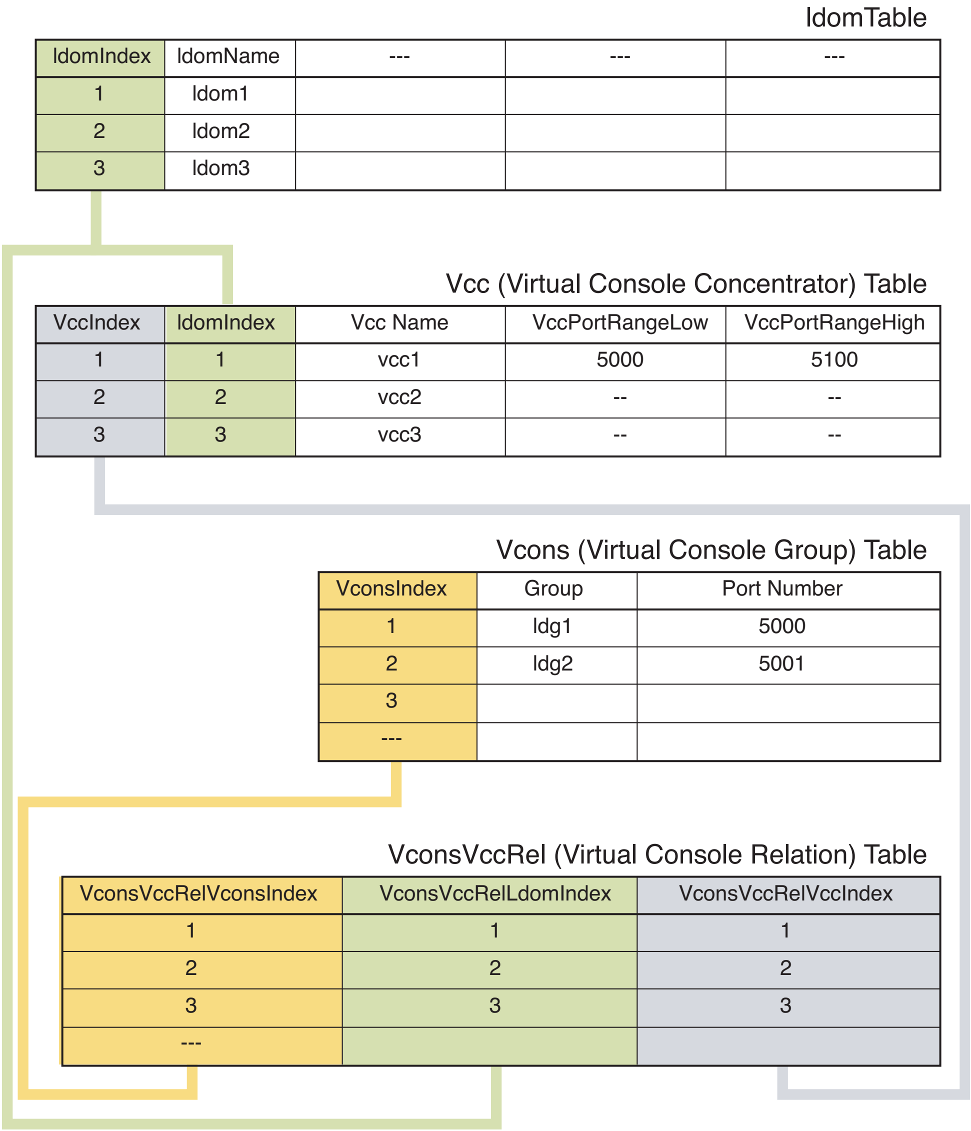
This table contains index values to show the intertable relationships between a logical domain, a virtual console concentrator, and console groups.

**TABLE 4-16** Virtual Console Relationship (VconsVccRel) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| VconsVccRelVconsIndex | Integer | Read-only | Value of the ldomVconsIndex in the ldomVconsTable |
| VconsVccRelLdomIndex | Integer | Read-only | Value of the ldomIndex in the ldomTable |
| VconsVccRelVccIndex | Integer | Read-only | Value of the ldomVccIndex in the ldomVccTable |

FIGURE 4-2 shows an example of the relationships among the virtual console tables and the Logical Domains Table.

**FIGURE 4-2** Relationship Among Virtual Console Tables and the Logical Domains Table



ldomTable

| ldomIndex | ldomName | --- | --- | --- |
|---|---|---|---|---|
| 1 | ldom1 | | | |
| 2 | ldom2 | | | |
| 3 | ldom3 | | | |

Vcc (Virtual Console Concentrator) Table

| VccIndex | ldomIndex | Vcc Name | VccPortRangeLow | VccPortRangeHigh |
|---|---|---|---|---|
| 1 | 1 | vcc1 | 5000 | 5100 |
| 2 | 2 | vcc2 | -- | -- |
| 3 | 3 | vcc3 | -- | -- |

Vcons (Virtual Console Group) Table

| VconsIndex | Group | Port Number |
|---|---|---|
| 1 | ldg1 | 5000 |
| 2 | ldg2 | 5001 |
| 3 | | |
| --- | | |

VconsVccRel (Virtual Console Relation) Table

| VconsVccRelVconsIndex | VconsVccRelLdomIndex | VconsVccRelVccIndex |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| --- | | |

# Other Scalar Variables and Tables

This section describes the following other scalar variables and tables provided by the LDoms MIB:

- Scalar Variables for Logical Domain Version Information
- Environment Variables Table
- Cryptographic Units Table
- I/O Bus Table

## Scalar Variables for Logical Domain Version Information

The Logical Domains Manager protocol supports logical domains versions. The version consists of both a major and minor number. The Logical Domain MIB has scalar variables to describe the logical domain version information.

**TABLE 4-17** Scalar Variables for Logical Domain Version Information

| Name | Data Type | Access | Description |
|---|---|---|---|
| `VersionMajor` | Integer | Read-only | Major version number |
| `VersionMinor` | Integer | Read-only | Minor version number |

## Environment Variables (`EnvVars`) Table

This table contains the OpenBoot™ PROM environment variables of all logical domains.

**TABLE 4-18** Environmental Variables (`EnvVars`) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| `EnvVarsLdomIndex` | Integer | Read-only | An integer used to index to the `ldomTable` representing the logical domain containing the environmental variables |
| `EnvVarsIndex` | Integer | Not accessible | Integer for an index of the table |
| `EnvVarsName` | Display string | Read-only | Name of the OpenBoot PROM variable |
| `EnvVarsValue` | Display string | Read-only | Value of the OpenBoot PROM variable |

## Cryptographic Units Table (CryptoTable)

This table describes the cryptographic (crypto) units belonging to all logical domains:

- MAU - Modular Arithmetic Unit; cryptographic unit for Sun UltraSPARC T1–based platforms
- CWQ - Control Word Queue; cryptographic unit for Sun UltraSPARC T2–based platforms

**TABLE 4-19** Cryptographic Units (CryptoTable) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| CryptoLdomIndex | Integer | Read-only | An integer used to index to the ldomTable representing the logical domain containing the cryptographic unit |
| CryptoIndex | Integer | Not accessible | An integer used to index the cryptographic unit in this table |
| CryptoCpuSet | Display string | Read-only | List of CPUs that is mapped to MAU-unit cpuset. For example, {0, 1, 2, 3} |

## Input/Output Bus Table (IOBusTable)

This table describes physical I/O devices, PCI buses, belonging to all logical domains.

**TABLE 4-20** Input/Output Bus (IOBusTable) Table

| Name | Data Type | Access | Description |
|---|---|---|---|
| IOBusLdomIndex | Integer | Read-only | An integer used to index to the ldomTable representing the logical domain containing the I/O bus |
| IOBusIndex | Integer | Not accessible | An integer used to index the I/O bus in this table |
| IOBusDevName | Display string | Read-only | Physical I/O device name |
| IOBusDevPath | Display string | Read-only | Physical I/O device path |

CHAPTER 5

# Using Traps and Starting and Stopping Logical Domains

This chapter contains the following sections:

- "Using LDoms MIB Module Traps" on page 33
- "Starting and Stopping a Logical Domain" on page 41

The first section describes how to set up your system to send and receive traps, the traps you can use to receive change notification for logical domains, and descriptions of the output.

The second section describes active management operations, which are controlled by using the ldomAdminState property of the Logical Domains Table (ldomTable). For more information about this property and this table, see TABLE 4-1.

## Using LDoms MIB Module Traps

This section describes:

- "Setting Up to Send and Receive Traps" on page 33
- "Descriptions of LDoms MIB Traps" on page 35

### Setting Up to Send and Receive Traps

The following procedures provide examples of how you can set up to send and receive traps.

## ▼ To Send Traps

- **Edit the /**`etc/sma/snmp/snmpd.conf` **file to add the directives to define the trap and inform version and destination.**

```
trapcommunity string --> define community string to be used when sending traps
trapsink host[community [port]] --> to send v1 traps
trap2sink host[community [port]] --> to send v2c traps
informsink host[community [port]] --> to send informs
```

For more details, refer to the Solaris 10 OS `snmpd.conf`(1M) man page.

For example, to send both v1 and v2c traps to the SNMP trap daemon running on the same host, add the following directives to the `/etc/sma/snmp/snmpd.conf` file:

```
trapcommunity public
trapsink localhost
trap2sink localhost
```

## ▼ To Receive Traps

- **Start the SNMP trap daemon utility,** `snmptrapd`**(1M).**

  Refer to the Solaris 10 OS `snmptrapd`(1M) man page for options to specify the format of the output.

  For example, the following input:

```
# /usr/sfw/sbin/snmptrapd -P -F “TRAP from %B on %m/%l/%y at %h:%j:%k
Enterprise=%N Type=%w SubType=%q \nwith Varbinds: %v \nSecurity info:%P\n\n"
localhost:162
```

displays the trap as follows:

```
TRAP from localhost on 5/18/2007 at 16:30:10 Enterprise=. Type=0 SubType=0
with Varbinds:  DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (47105)
0:07:51.05     SNMPv2-MIB::snmpTrapOID.0 = OID: SUN-LDOM-MIB::ldomCreate
SUN-LDOM-MIB::ldomIndexNotif = INTEGER: 3 SUN-LDOM-MIB::ldomName = STRING: ldg2
SUN-LDOM-MIB::ldomTrapDesc = STRING: Ldom Created
Security info:TRAP2, SNMP v2c, community public
```

## Descriptions of LDoms MIB Traps

This section describes the following LDoms MIB traps you can use:

- "Logical Domain Creation (ldomCreate)" on page 35
- "Logical Domain Destroy (ldomDestroy)" on page 36
- "Logical Domain State Change (ldomStateChange)" on page 36
- "Virtual CPU Change (ldomVCpuChange)" on page 36
- "Virtual Memory Change (ldomVMemChange)" on page 37
- "Virtual Disk Service Change (ldomVdsChange)" on page 37
- "Virtual Disk Change (ldomVdiskChange)" on page 38
- "Virtual Switch Change (ldomVswChange)" on page 38
- "Virtual Network Change (ldomVnetChange)" on page 39
- "Virtual Console Concentrator Change (ldomVccChange)" on page 40
- "Virtual Console Group Change (ldomVconsChange)" on page 40

---

**Note –** Be sure that you have added the `ldom` prefix to trap names when setting traps.

---

## Logical Domain Creation (ldomCreate)

Setting this trap notifies you of any logical domains that are created.

**TABLE 5-1** Logical Domain Creation (ldomCreate) Trap

| Name | Data Type | Description |
|---|---|---|
| `ldomIndexNotif` | Integer | Index into the `ldomTable` |
| `ldomName` | Display string | Name of the logical domain |
| `ldomTrapDesc` | Display string | Description of the trap |

## Logical Domain Destroy (ldomDestroy)

Setting this trap notifies you of any logical domains that are destroyed.

**TABLE 5-2** Logical Domain Destroy (ldomDestroy) Trap

| Name | Data Type | Description |
|---|---|---|
| ldomIndexNotif | Integer | Index into the ldomTable |
| ldomName | Display string | Name of the logical domain |
| ldomTrapDesc | Display string | Description of the trap |

## Logical Domain State Change (ldomStateChange)

Setting this trap notifies you of any operating state changes in a logical domain.

**TABLE 5-3** Logical Domain State Change (ldomStateChange) Trap

| Name | Data Type | Description |
|---|---|---|
| ldomIndexNotif | Integer | Index into the ldomTable |
| ldomName | Display string | Name of the logical domain |
| ldomOperState | Integer | New state of the logical domain |
| ldomStatePrev | Integer | Previous state of the logical domain |
| ldomTrapDesc | Display string | Description of the trap |

## Virtual CPU Change (ldomVCpuChange)

Setting this trap notifies you of any changes in the number of virtual CPUs for a logical domain.

**TABLE 5-4** Logical Domain Virtual CPU Change (ldomVCpuChange) Trap

| Name | Data Type | Description |
|---|---|---|
| ldomIndexNotif | Integer | Index into the ldomTable |
| ldomName | Display string | Name of the logical domain containing the virtual CPU |

**TABLE 5-4** Logical Domain Virtual CPU Change (ldomVCpuChange) Trap *(Continued)*

| Name | Data Type | Description |
|---|---|---|
| ldomNumVCPU | Integer | New number of virtual CPUs for the logical domain |
| ldomNumVCPUPrev | Integer | Previous number of virtual CPUs for this logical domain |
| ldomTrapDesc | Display string | Description of the trap |

## Virtual Memory Change (ldomVMemChange)

Setting this trap notifies you of any changes in the amount of virtual memory for a logical domain.

**TABLE 5-5** Logical Domain Virtual Memory Change (ldomVMemChange) Trap

| Name | Data Type | Description |
|---|---|---|
| ldomIndexNotif | Integer | Index into the ldomTable |
| ldomName | Display string | Name of the logical domain containing the virtual memory |
| ldomMemSize | Integer | Amount of virtual memory for the logical domain |
| ldomMemSizePrev | Integer | Previous amount of virtual memory for this logical domain |
| ldomMemUnit | Integer | Memory unit for virtual memory |
| ldomMemUnitPrev | Integer | Memory unit for previous virtual memory |
| ldomTrapDesc | Display string | Description of the trap |

## Virtual Disk Service Change (ldomVdsChange)

Setting this trap notifies you of any changes to a virtual disk service for a logical domain.

**TABLE 5-6** Logical Domain Virtual Disk Service Change (ldomVdsChange) Trap

| Name | Data Type | Description |
|---|---|---|
| ldomIndexNotif | Integer | Index into the ldomTable |
| ldomName | Display string | Name of the logical domain containing the virtual disk service |

**TABLE 5-6** Logical Domain Virtual Disk Service Change (ldomVdsChange) Trap *(Continued)*

| Name | Data Type | Description |
|---|---|---|
| ldomVdsServiceName | Display string | Name of the virtual disk service that has changed |
| ldomChangeFlag | Integer | Indicates the change that occurred to this virtual disk service:<br>1=Added<br>2=Modified<br>3=Removed |
| ldomTrapDesc | Display string | Description of the trap |

## Virtual Disk Change (ldomVdiskChange)

Setting this trap notifies you of any changes to a virtual disk for a logical domain.

**TABLE 5-7** Virtual Disk Change (ldomVdiskChange) Trap

| Name | Data Type | Description |
|---|---|---|
| ldomIndexNotif | Integer | Index into the ldomTable |
| ldomName | Display string | Name of the logical domain containing the virtual disk |
| ldomVdiskName | Display string | Name of the virtual disk that has changed |
| ldomChangeFlag | Integer | Indicates the change that occurred to this virtual disk service:<br>1=Added<br>2=Modified<br>3=Removed |
| ldomTrapDesc | Display string | Description of the trap |

## Virtual Switch Change (ldomVswChange)

Setting this trap notifies you of any changes to a virtual switch for a logical domain.

**TABLE 5-8** Virtual Switch Change (ldomVswChange) Trap

| Name | Data Type | Description |
|---|---|---|
| ldomIndexNotif | Integer | Index into the ldomTable |
| ldomName | Display string | Name of the logical domain containing the virtual switch service |

**TABLE 5-8** Virtual Switch Change (`ldomVswChange`) Trap *(Continued)*

| Name | Data Type | Description |
|---|---|---|
| `ldomVswServiceName` | Display string | Name of the virtual switch services that has changes |
| `ldomChangeFlag` | Integer | Indicates the change that occurred to this virtual switch service:<br>1=Added<br>2=Modified<br>3=Removed |
| `ldomTrapDesc` | Display string | Description of the trap |

## Virtual Network Change (`ldomVnetChange`)

Setting this trap notifies you of any changes to a virtual network for a logical domain.

**TABLE 5-9** Virtual Network Change (`ldomVnetChange`) Trap

| Name | Data Type | Description |
|---|---|---|
| `ldomIndexNotif` | Integer | Index into the `ldomTable` |
| `ldomName` | Display string | Name of the logical domain containing the virtual network |
| `ldomVnetDevName` | Display string | Name of the virtual network for the logical domain |
| `ldomChangeFlag` | Integer | Indicates the change that occurred to this virtual network:<br>1=Added<br>2=Modified<br>3=Removed |
| `ldomTrapDesc` | Display string | Description of the trap |

# Virtual Console Concentrator Change (ldomVccChange)

Setting this trap notifies you of any changes to a virtual console concentrator for a logical domain.

**TABLE 5-10** Virtual Console Concentrator Change (ldomVccChange) Trap

| Name | Data Type | Description |
|---|---|---|
| ldomIndexNotif | Integer | Index into the ldomTable |
| ldomName | Display string | Name of the logical domain containing the virtual memory |
| ldomVccName | Display string | Name of the virtual console concentrator service that has changes |
| ldomChangeFlag | Integer | Indicates the change that occurred to this virtual console concentrator:<br>1=Added<br>2=Modified<br>3=Removed |
| ldomTrapDesc | Display string | Description of the trap |

# Virtual Console Group Change (ldomVconsChange)

Setting this trap notifies you of any changes to a virtual console group for a logical domain.

**TABLE 5-11** Virtual Console Group Change (ldomVconsChange) Trap

| Name | Data Type | Description |
|---|---|---|
| ldomIndexNotif | Integer | Index into the ldomTable |
| ldomName | Display string | Name of the logical domain containing the virtual memory |
| ldomVconsGroupName | Display string | Name of the virtual console group that has changes |
| ldomChangeFlag | Integer | Indicates the change that occurred to this virtual console group:<br>1=Added<br>2=Modified<br>3=Removed |
| ldomTrapDesc | Display string | Description of the trap |

## Starting and Stopping a Logical Domain

You can use the snmpset(1M) command to start or stop any logical domain:

- The snmpset command with a write of 1 (active) to the ldomAdminState property starts the logical domain.
- The snmpset command with a write of 2 (stopping) to the ldomAdminState property stops the logical domain.

### ▼ To Start a Logical Domain

The start logical domain operation starts an existing bound logical domain. If a logical domain with the given domain name does not exist or is not already bound, then this operation fails.

1. **Check to see that the control (primary) domain exists and is bound.**

```
# ldm list primary
Name          State     Flags    Cons      VCPU  Memory    Util    Uptime
primary       bound     ---cv              4     1G
```

2. **Start the primary domain.**

```
% snmpset -v1 -c private localhost SUN-LDOM-MIB::ldomTable.1.ldomAdminState.1 =
1
```

3. **Check to see that the primary domain is now active using the ldm list command from the Logical Domains Manager.**

```
# ldm list primary
Name          State     Flags    Cons      VCPU  Memory    Util    Uptime
primary       active    -t-cv              4     1G        0.0%    0s
```

4. **You can also retrieve the logical domain state using the SNMP tools.**

```
% snmpget -v 1 -c public localhost SUN-LDOM-MIB::ldomOperState.1
SUN-LDOM-MIB::ldomOperState.1 = INTEGER: active(1)
```

## ▼ To Stop a Logical Domain

The stop operation stops a started logical domain. Any instances of an operating system that are hosted by the domain are stopped.

1. **Stop the primary domain.**

```
% snmpset -v1 -c private localhost SUN-LDOM-MIB::ldomTable.1.ldomAdminState.1 =
2
```

2. **Check to see that the `primary` domain is now bound (stopped) using the `ldm list` command from the Logical Domains Manager.**

```
# ldm list primary
Name          State     Flags   Cons     VCPU  Memory   Util   Uptime
primary       bound     ---cv            4     1G
```

3. **You can also retrieve the logical domain state using the SNMP tools.**

```
% snmpget -v 1 -c public localhost SUN-LDOM-MIB::ldomOperState.1
SUN-LDOM-MIB::ldomOperState.1 = INTEGER: bound(6)
```

# Glossary

This list defines terminology, abbreviations, and acronyms in the Logical Domains Management Information Base documentation.

## A

**ASN** Abstract Syntax Notation

## C

**control domain** A privileged domain that runs the Logical Domains Manager

**CPU** central processing unit

**CWQ** Control Word Queue; cryptographic unit for Sun UltraSPARC T2–based platforms

## I

**I/O** Input/Output

# L

**LDoms** Logical Domains software

**logical domains** Virtual machines consisting of a set of virtual resources, such as CPUs, memory, disks, and networks, capable of hosting an OS.

# M

**MAC** media access control address

**MAU** Modular Arithmetic Unit; cryptographic unit for Sun UltraSPARC T1–based platforms

**MIB** Management Information Base; data model for a specific set of objects being managed by SNMP

# O

**OID** Object Identifier; a sequence of numbers that uniquely identifies each object in a MIB

**OS** operating system

# P

**PROM** programmable read-only memory

## S

| | |
|---|---|
| **SMA** | System Management Agent; default Solaris OS SNMP agent |
| **SMF** | Service Management Facility; primary infrastructure for service management within Solaris 10 OS |
| **SMI** | Structure of Management Information |
| **SNMP** | Simple Network Management Protocol; a widely used protocol for monitoring the health and welfare of network equipment, computer equipment, and devices |

## T

| | |
|---|---|
| **TCP** | Transmission Control Protocol |

## V

| | |
|---|---|
| `vcc` | virtual console concentrator |
| `vcons` | virtual console |
| `vcpu` | virtual CPU |
| `vdisk` | virtual disk |
| `vds` | virtual disk server |
| `vdsdev` | virtual disk server device |
| `vnet` | virtual network |
| **vNTS** | virtual network terminal service |
| `vsw` | virtual switch |

## X

| | |
|---|---|
| **XML** | Extensible Markup Language |