



# Logical Domains (LDoms) 1.0.3 Administration Guide

---

Sun Microsystems, Inc.  
[www.sun.com](http://www.sun.com)

Part No. 820-4894-10  
May 2008, Revision A

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

Copyright © 2008 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JumpStart, OpenBoot, Sun Fire, Netra, SunSolve, Sun BluePrints, Sun Blade, Sun Ultra, and Sun VTS are service marks, trademarks, or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

The Adobe PostScript logo is a trademark of Adobe Systems, Incorporated.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright © 2008 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JumpStart, OpenBoot, Sun Fire, Netra, SunSolve, Sun BluePrints, Sun Blade, Sun Ultra, et Sun VTS sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Le logo Adobe PostScript est une marque déposée de Adobe Systems, Incorporated.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE



Adobe PostScript

# Contents

---

## **Preface   xvii**

## **1. Overview of the Logical Domains Software   1**

Hypervisor and Logical Domains   1

Logical Domains Manager   3

    Roles for Logical Domains   4

    Command-Line Interface   4

    Virtual Input/Output   5

        Virtual Network   5

        Virtual Storage   6

        Virtual Console   6

    Dynamic Reconfiguration   6

    Delayed Reconfiguration   6

    Persistent Configurations   7

## **2. Security   9**

Security Considerations   9

Solaris Security Toolkit and the Logical Domains Manager   10

    Hardening   11

    Minimizing Logical Domains   12

Authorization 12

Auditing 13

Compliance 14

### **3. Installing and Enabling Software 15**

Upgrading the Solaris OS 15

Saving and Restoring the Logical Domains Constraints Database File 15

Using Live Upgrade on the Control Domain 16

Upgrading to LDOMs 1.0.3 Software 16

▼ To Upgrade From LDOMs 1.0 to LDOMs 1.0.3 Software 16

Freshly Installing Software on the Control Domain 18

▼ To Install the Solaris 10 OS 18

▼ To Upgrade System Firmware 19

▼ To Upgrade System Firmware Without an FTP Server 20

▼ To Downgrade System Firmware 21

Downloading Logical Domains Manager and Solaris Security Toolkit 21

▼ To Download the Logical Domains Manager, Solaris Security Toolkit, and Logical Domains MIB 21

Installing Logical Domains Manager and Solaris Security Toolkit 22

Using the Installation Script to Install the Logical Domains Manager 1.0.3 and Solaris Security Toolkit 4.2 Software 23

▼ To Install Using the `install-ldm` Script With No Options 24

▼ To Install Using the `install-ldm` Script With the `-d` Option 27

▼ To Install Using the `install-ldm` Script With the `-d none` Option 28

▼ To Install Using the `install-ldm` Script With the `-p` Option 29

Using JumpStart to Install the Logical Domains Manager 1.0.3 and Solaris Security Toolkit 4.2 Software 29

▼ To Set Up a JumpStart Server 30

▼ To Install Using JumpStart Software 30

Installing Logical Domains Manager and Solaris Security Toolkit Software  
Manually 32

- ▼ To Install the Logical Domains Manager (LDoms) 1.0.3 Software  
Manually 32
- ▼ (*Optional*) To Install the Solaris Security Toolkit 4.2 Software  
Manually 33
- ▼ (*Optional*) To Harden the Control Domain Manually 33
- ▼ To Validate Hardening 34
- ▼ To Undo Hardening 34

Enabling the Logical Domains Manager Daemon 35

- ▼ To Enable the Logical Domains Manager Daemon 35

Creating Authorization and Profiles and Assigning Roles for User Accounts 35

Managing User Authorizations 36

- ▼ To Add an Authorization for a User 36
- ▼ To Delete All Authorizations for a User 37

Managing User Profiles 37

- ▼ To Add a Profile for a User 37
- ▼ To Delete All Profiles for a User 37

Assigning Roles to Users 37

- ▼ To Create a Role and Assign the Role to a User 38

**4. Setting Up Services and Logical Domains 39**

Output Messages 39

Sun UltraSPARC T1 Processors 39

Sun UltraSPARC T2 Processors 40

Creating Default Services 40

- ▼ To Create Default Services 40

Initial Configuration of the Control Domain 42

- ▼ To Set Up the Control Domain 42

Rebooting to Use Logical Domains 44

▼ To Reboot	44
Enabling Networking Between the Control/Service Domain and Other Domains	45
▼ To Configure the Virtual Switch as the Primary Interface	45
Enabling the Virtual Network Terminal Server Daemon	46
▼ To Enable the Virtual Network Terminal Server Daemon	47
Creating and Starting a Guest Domain	47
▼ To Create and Start a Guest Domain	48
Jump-Starting a Guest Domain	51

## 5. Using Virtual Disks With Logical Domains 53

Introduction to Virtual Disks	53
Managing Virtual Disks	54
▼ To Add a Virtual Disk	54
▼ To Export a Virtual Disk Backend Multiple Times	55
▼ To Change Virtual Disk Options	56
▼ To Change the Timeout Option	56
▼ To Remove a Virtual Disk	56
Virtual Disk Appearance	56
Full Disk	56
Single Slice Disk	57
Virtual Disk Backend Options	57
Read-only (ro) Option	57
Exclusive (excl) Option	57
Slice (slice) Option	58
Virtual Disk Backend	59
Physical Disk or Disk LUN	59
▼ To Export a Physical Disk as a Virtual Disk	59
Physical Disk Slice	60

▼ To Export a Physical Disk Slice as a Virtual Disk	60
▼ To Export Slice 2	61
File and Volume	61
File or Volume Exported as a Full Disk	61
▼ To Export a File as a Full Disk	62
File or Volume Exported as a Single Slice Disk	62
▼ To Export a ZFS Volume as a Single Slice Disk	63
Exporting Volumes and Backward Compatibility	63
Summary of How Different Types of Backends Are Exported	64
Guidelines	64
CD, DVD and ISO Images	65
▼ To Export a CD or DVD From the Service Domain to the Guest Domain	66
Virtual Disk Timeout	67
Virtual Disk and SCSI	68
Virtual Disk and the <code>format(1M)</code> Command	68
Using ZFS With Virtual Disks	69
Creating a Virtual Disk on Top of a ZFS Volume	69
▼ To Create a Virtual Disk on Top of a ZFS Volume	69
Using ZFS Over a Virtual Disk	70
▼ To Use ZFS Over a Virtual Disk	71
Using ZFS for Boot Disks	72
▼ To Use ZFS for Boot Disks	72
Using Volume Managers in a Logical Domains Environment	74
Using Virtual Disks on Top of Volume Managers	74
Using Virtual Disks on Top of SVM	75
Using Virtual Disks When VxVM Is Installed	76
Using Volume Managers on Top of Virtual Disks	77
Using ZFS on Top of Virtual Disks	77

Using SVM on Top of Virtual Disks	77
Using VxVM on Top of Virtual Disks	78

## 6. Other Information and Tasks 79

Restrictions on Entering Names in the CLI	79
---	----

File Names ( <i>file</i> ) and Variable Names ( <i>var_name</i> )	79
---	----

Virtual Disk Server <i>backend</i> and Virtual Switch device Names	79
--	----

Configuration Name ( <i>config_name</i> )	79
---	----

All Other Names	80
-----------------	----

Using <code>ldm list</code> Subcommands	80
---	----

Machine-Readable Output	80
-------------------------	----

▼ To Show Syntax Usage for <code>ldm</code> Subcommands	80
---	----

Flag Definitions	83
------------------	----

Utilization Statistic Definition	84
----------------------------------	----

Examples of Various Lists	84
---------------------------	----

▼ To Show Software Versions ( <code>-v</code> )	84
---	----

▼ To Generate a Short List	84
----------------------------	----

▼ To Generate a Long List ( <code>-l</code> )	85
---	----

▼ To Generate an Extended List ( <code>-e</code> )	86
--	----

▼ To Generate a Parseable, Machine-Readable List ( <code>-p</code> )	88
--	----

▼ To Show the Status of a Domain	88
----------------------------------	----

▼ To List a Variable	89
----------------------	----

▼ To List Bindings	89
--------------------	----

▼ To List Configurations	90
--------------------------	----

▼ To List Devices	90
-------------------	----

▼ To List Services	92
--------------------	----

Listing Constraints	92
---------------------	----

▼ To List Constraints for One Domain	92
--------------------------------------	----

▼ To List Constraints in XML Format	93
-------------------------------------	----



▼ To List Constraints in a Machine-Readable Format	94
The <code>ldm stop-domain</code> Command Can Time Out If the Domain Is Heavily Loaded	95
Determining the Solaris Network Interface Name Corresponding to a Virtual Network Device	96
▼ To Find Solaris OS Network Interface Name	96
Assigning MAC Addresses Automatically or Manually	97
Range of MAC Addresses Assigned to Logical Domains Software	98
Automatic Assignment Algorithm	98
Duplicate MAC Address Detection	99
Freed MAC Addresses	99
CPU and Memory Address Mapping	100
CPU Mapping	100
▼ To Determine the CPU Number	100
Memory Mapping	101
▼ To Determine the Real Memory Address	101
Examples of CPU and Memory Mapping	101
Configuring Split PCI Express Bus to Use Multiple Logical Domains	103
▼ To Create a Split PCI Configuration	104
Enabling the I/O MMU Bypass Mode on a PCI Bus	106
Using Console Groups	107
▼ To Combine Multiple Consoles Into One Group	107
Moving a Logical Domain From One Server to Another	108
▼ To Set Up Domains to Move	108
▼ To Move the Domain	108
Removing Logical Domains	109
▼ To Remove All Guest Logical Domains	109
Operating the Solaris OS With Logical Domains	110
OpenBoot Firmware Not Available After Solaris OS Has Started If Domaining Is Enabled	110

Power-Cycling a Server	110
▼ To Save Your Current Logical Domain Configurations to the SC	111
Result of an OpenBoot <code>power-off</code> Command	111
Result of Solaris OS Breaks	111
Results from Halting or Rebooting the Control Domain	111
Using LDoms With ALOM CMT	113
▼ To Reset the Logical Domain Configuration to the Default or Another Configuration	113
Enabling and Using BSM Auditing	114
▼ To Use the <code>enable-bsm.fin</code> Finish Script	114
▼ To Use the Solaris OS <code>bsmconv(1M)</code> Command	115
▼ To Verify that BSM Auditing is Enabled	115
▼ To Disable Auditing	116
▼ To Print Audit Output	116
▼ To Rotate Audit Logs	116
Supported Network Adapters	116
▼ To Determine If a Network Adapter Is GLDv3-Compliant	117
Configuring Virtual Switch and Service Domain for NAT and Routing	117
▼ To Set Up the Virtual Switch to Provide External Connectivity to Domains	118
Configuring IPMP in a Logical Domains Environment	118
Configuring Virtual Network Devices into an IPMP Group in a Logical Domain	119
Configuring and Using IPMP in the Service Domain	120
<b>Glossary</b>	<b>123</b>

# Figures

---

FIGURE 1-1	Hypervisor Supporting Two Logical Domains	2
FIGURE 5-1	Virtual Disks With Logical Domains	54
FIGURE 6-1	Two Virtual Networks Connected to Separate Virtual Switch Instances	119
FIGURE 6-2	Each Virtual Network Device Connected to Different Service Domains	120
FIGURE 6-3	Two Network Interfaces Configured as Part of IPMP Group	121



# Tables

---

TABLE 1-1	Logical Domain Roles	4
TABLE 2-1	The <code>ldm</code> Subcommands and User Authorizations	13
TABLE 6-1	Expected Behavior of Halting or Rebooting the Control ( <code>primary</code> ) Domain	112



# Code Examples

---

CODE EXAMPLE 3-1	Directory Structure for Downloaded Logical Domains 1.0.3 Software	22
CODE EXAMPLE 3-2	Output From Hardened Solaris Configuration for LDoms	25
CODE EXAMPLE 3-3	Output From Choosing Customized Configuration Profile	26
CODE EXAMPLE 3-4	Output From Successful Run of the install-ldm -d Script	27
CODE EXAMPLE 3-5	Output From Successful Run of the install-ldm -d none Script	28
CODE EXAMPLE 6-1	Syntax Usage for All ldm Subcommands	80
CODE EXAMPLE 6-2	Software Versions Installed	84
CODE EXAMPLE 6-3	Short List for All Domains	84
CODE EXAMPLE 6-4	Long List for All Domains	85
CODE EXAMPLE 6-5	Extended List for all Domains	86
CODE EXAMPLE 6-6	Machine-Readable List	88
CODE EXAMPLE 6-7	Domain Status	88
CODE EXAMPLE 6-8	Variable List for a Domain	89
CODE EXAMPLE 6-9	Bindings List for a Domain	89
CODE EXAMPLE 6-10	Configurations List	90
CODE EXAMPLE 6-11	List of All Server Resources	90
CODE EXAMPLE 6-12	Services List	92
CODE EXAMPLE 6-13	Constraints List for One Domain	92
CODE EXAMPLE 6-14	Constraints for a Domain in XML Format	93
CODE EXAMPLE 6-15	Constraints for All Domains in a Machine-Readable Format	94





# Preface

---

The *Logical Domains (LDDoms) 1.0.3 Administration Guide* provides detailed information and procedures that describe the overview, security considerations, installation, configuration, modification, and execution of common tasks for the Logical Domains Manager 1.0.3 software on supported servers, blades, and server modules. Refer to “Supported Platforms” in the *Logical Domains (LDDoms) 1.0.3 Release Notes* for a list. This guide is intended for the system administrators on these servers who have a working knowledge of UNIX<sup>®</sup> systems and the Solaris<sup>™</sup> Operating System (Solaris OS).

---

## Before You Read This Document

If you do not have a working knowledge of UNIX commands and procedures and your Solaris Operating System, read the Solaris OS user and system administrator documentation provided with your system hardware, and consider UNIX system administration training.

---

## How This Book Is Organized

[Chapter 1](#) provides an overview of the Logical Domains software.

[Chapter 2](#) discusses the Solaris Security Toolkit, and how it can provide security for the Solaris OS in logical domains.

[Chapter 3](#) provides detailed procedures for upgrading or installing, and enabling Logical Domains Manager software.

[Chapter 4](#) provides detailed procedures for setting up services and logical domains.

[Chapter 5](#) describes how to use virtual disks with Logical Domains software.

[Chapter 6](#) provides other information and procedures for executing common tasks in using the Logical Domains software to manage logical domains.

[Glossary](#) is a list of LDoms-specific abbreviations, acronyms, and terms and their definitions.

---

## 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	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
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 <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
<b>AaBbCc123</b>	What you type, when contrasted with on-screen computer output	% <b>su</b> Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. To delete a file, type <b>rm</b> <i>filename</i> .

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

---

# Related Documentation

The *Logical Domains (LDoms) 1.0.3 Administration Guide* and *Release Notes* are available at:

<http://docs.sun.com>

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>

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

<http://docs.sun.com>

Type the name of the server, software, or Solaris OS in the Search box to find the documents you need.

Application	Title	Part Number	Format	Location
Release notes for LDoms	<i>Logical Domains (LDoms) 1.0.3 Release Notes</i>	820-4895-10	HTML PDF	Online
Solaris man pages for LDoms	Solaris 10 Reference Manual Collection: • <i>drd(1M)</i> man page • <i>vntsd(1M)</i> man page	N/A	HTML	Online
LDoms man page	<i>ldm(1M)</i> man page	N/A	SGML	Online
Basics for Logical Domains software	<i>Logical Domains (LDoms) 1.0.1 Manager Man Page Guide</i>	820-4896-10	PDF	Online
Basics for Logical Domains software	<i>Beginners Guide to LDoms: Understanding and Deploying Logical Domains Software</i>	820-0832-20	PDF	Online
Administration for LDoms MIB	<i>Logical Domains (LDoms) MIB 1.0.1 Administration Guide</i>	820-2319-10	HTML PDF	Online
Release notes for LDoms MIB	<i>Logical Domains (LDoms) MIB 1.0.1 Release Notes</i>	820-2320-10	HTML PDF	Online
Solaris OS including installation, using JumpStart™, and using the SMF	Solaris 10 Collection	N/A	HTML PDF	Online
Security	<i>Solaris Security Toolkit 4.2 Administration Guide</i>	819-1402-10	HTML PDF	Online
Security	<i>Solaris Security Toolkit 4.2 Reference Manual</i>	819-1503-10	HTML PDF	Online
Security	<i>Solaris Security Toolkit 4.2 Release Notes</i>	819-1504-10	HTML PDF	Online
Security	<i>Solaris Security Toolkit 4.2 Man Page Guide</i>	819-1505-10	HTML PDF	Online

---

# Documentation, Support, and Training

Sun Function	URL
Documentation	<a href="http://docs.sun.com">http://docs.sun.com</a>
Support	<a href="http://www.sun.com/support">http://www.sun.com/support</a>
Training	<a href="http://www.sun.com/training">http://www.sun.com/training</a>

---

## 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 with 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 (LDDoms) 1.0.3 Administration Guide*, part number 820-4894-10.



# Overview of the Logical Domains Software

---

This chapter contains a brief overview of the Logical Domains software. All of the Solaris OS functionality necessary to use Sun's Logical Domains technology is in the Solaris 10 11/06 release (at a minimum) with the addition of necessary patches. However, system firmware and the Logical Domains Manager are also required to use logical domains. Refer to "Required and Recommended Software" in the *Logical Domains (LDom)s 1.0.3 Release Notes* for specific details.

---

## Hypervisor and Logical Domains

This section provides a brief overview of the SPARC® hypervisor and the logical domains it supports.

The SPARC hypervisor is a small firmware layer that provides a stable virtualized machine architecture to which an operating system can be written. Sun servers using the hypervisor provide hardware features to support the hypervisor's control over a logical operating system's activities.

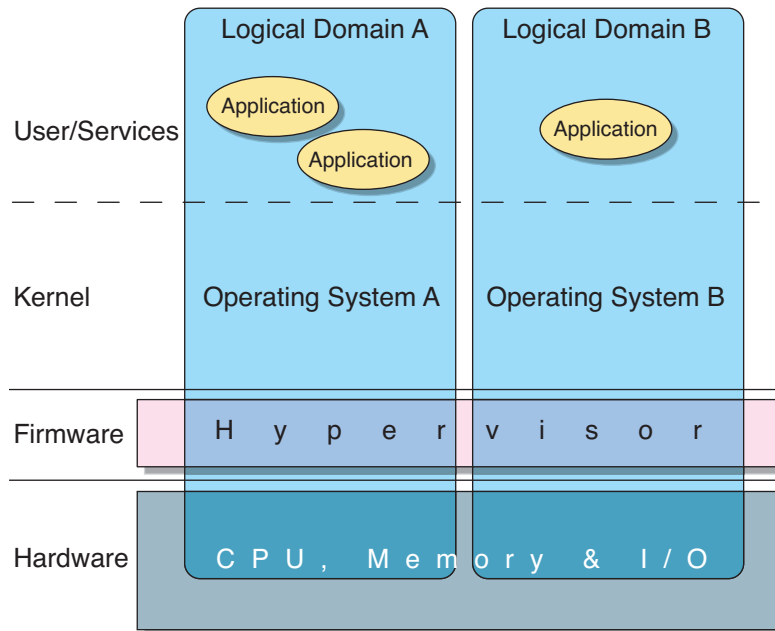
A logical domain is a discrete logical grouping with its own operating system, resources, and identity within a single computer system. Each logical domain can be created, destroyed, reconfigured, and rebooted independently, without requiring a power cycle of the server. You can run a variety of applications software in different logical domains and keep them independent for performance and security purposes.

Each logical domain is allowed to observe and interact with only those server resources made available to it by the hypervisor. Using the Logical Domains Manager, the system administrator specifies what the hypervisor should do through the control domain. Thus, the hypervisor enforces the partitioning of the resources of a server and provides limited subsets to multiple operating system environments.

This is the fundamental mechanism for creating logical domains. The following diagram shows the hypervisor supporting two logical domains. It also shows the layers that make up the Logical Domains functionality:

- Applications, or user/services
- Kernel, or operating systems
- Firmware, or hypervisor
- Hardware, including CPU, memory, and I/O

**FIGURE 1-1** Hypervisor Supporting Two Logical Domains



The number and capabilities of each logical domain that a specific SPARC hypervisor supports are server-dependent features. The hypervisor can allocate subsets of the overall CPU, memory, and I/O resources of a server to a given logical domain. This enables support of multiple operating systems simultaneously, each within its own logical domain. Resources can be rearranged between separate logical domains with an arbitrary granularity. For example, memory is assignable to a logical domain with an 8-kilobyte granularity.

Each virtual machine can be managed as an entirely independent machine with its own resources, such as:

- Kernel, patches, and tuning parameters
- User accounts and administrators
- Disks



- Network interfaces, MAC addresses, and IP addresses

Each virtual machine can be stopped, started, and rebooted independently of each other without requiring a power cycle of the server.

The hypervisor software is responsible for maintaining the separation between logical domains. The hypervisor software also provides logical domain channels (LDCs), so that logical domains can communicate with each other. Using logical domain channels, domains can provide services to each other, such as networking or disk services.

The system controller monitors and runs the physical machine, but it does not manage the virtual machines. The Logical Domains Manager runs the virtual machines.

---

## Logical Domains Manager

The Logical Domains Manager is used to create and manage logical domains. There can be only one Logical Domains Manager per server. The Logical Domains Manager maps logical domains to physical resources.

# Roles for Logical Domains

All logical domains are the same except for the roles that you specify for them. There are multiple roles that logical domains can perform.

**TABLE 1-1** Logical Domain Roles

Domain Role	Description
Control domain	Domain in which the Logical Domains Manager runs allowing you to create and manage other logical domains and allocate virtual resources to other domains. There can be only one control domain per server. The initial domain created when installing Logical Domains software is a control domain and is named <code>primary</code> .
Service domain	Domain that provides virtual device services to other domains, such as a virtual switch, a virtual console concentrator, and a virtual disk server.
I/O domain	Domain that has direct ownership of and direct access to physical I/O devices, such as a network card in a PCI Express controller. Shares the devices with other domains in the form of virtual devices when the I/O domain is also the control domain. The number of I/O domains you can have is dependent on your platform architecture. For example, if you are using a Sun UltraSPARC® T1 processor, you can have a maximum of two I/O domains, one of which also must be the control domain.
Guest domain	Domain that is managed by the control domain and uses services from the I/O and service domains.

If you have an existing system and already have an operating system and other software running on your server, that will be your control domain once you install the Logical Domains Manager. You might want to remove some of your applications from the control domain once it is set up, and balance the load of your applications throughout your domains to make the most efficient use of your system.

## Command-Line Interface

The Logical Domains Manager provides a command-line interface (CLI) for the system administrator to create and configure logical domains. The CLI is a single command, `ldm(1M)`, with multiple subcommands.

To use the Logical Domains Manager CLI, you must have the Logical Domains Manager daemon, `ldmd`, running. The `ldm(1M)` command and its subcommands are described in detail in the `ldm(1M)` man page and the *Logical Domains (LDoms) Manager Man Page Guide*. The `ldm(1M)` man page is part of the `SUNWldm` package and is installed when the `SUNWldm` package is installed.

To execute the `ldm` command, you must have the `/opt/SUNWldm/bin` directory in your UNIX `$PATH` variable. To access the `ldm(1M)` man page, add the directory path `/opt/SUNWldm/man` to the variable `$MANPATH`. Both are shown as follows:

```
$ PATH=$PATH:/opt/SUNWldm/bin; export PATH (for Bourne or K shell)
$ MANPATH=$MANPATH:/opt/SUNWldm/man; export MANPATH
% set PATH=($PATH /opt/SUNWldm/bin) (for C shell)
% set MANPATH=($MANPATH /opt/SUNWldm/man)
```

## Virtual Input/Output

In a Logical Domains environment, an administrator can provision up to 32 domains on a Sun Fire™ or SPARC Enterprise T1000 or T2000 server. Though each domain can be assigned dedicated CPUs and memory, the limited number of I/O buses and physical I/O slots in these systems makes it impossible to provide all domains exclusive access to the disk and network devices. Though some physical devices can be shared by splitting the PCI Express® (PCI-E) bus into two (see [“Configuring Split PCI Express Bus to Use Multiple Logical Domains” on page 103](#)), it is not sufficient to provide all domains exclusive device access. This lack of direct physical I/O device access is addressed by implementing a virtualized I/O model.

All logical domains with no direct I/O access are configured with virtual I/O devices that communicate with a service domain, which runs a service to provide access to a physical device or its functions. In this client-server model, virtual I/O devices either communicate with each other or a service counterpart through interdomain communication channels called logical domain channels (LDCs). In Logical Domains 1.0.3 software, the virtualized I/O functionality comprises support for virtual networking, storage, and consoles.

## Virtual Network

The virtual network support is implemented using two components: the virtual network and virtual network switch device. The virtual network (`vnet`) device emulates an Ethernet device and communicates with other `vnet` devices in the system using a point-to-point channel. The virtual switch (`vsw`) device mainly functions as a multiplexor of all the virtual network's incoming and outgoing packets. The `vsw` device interfaces directly with a physical network adapter on a service domain, and sends and receives packets on a virtual network's behalf. The `vsw` device also functions as a simple layer-2 switch and switches packets between the `vnet` devices connected to it within the system.

## Virtual Storage

The virtual storage infrastructure enables logical domains to access block-level storage that is not directly assigned to them through a client-server model. It consists of two components: a virtual disk client (`vdcc`) that exports as a block device interface; and a virtual disk service (`vds`) that processes disk requests on behalf of the virtual disk client and submits them to the physical storage residing on the service domain. Although the virtual disks appear as regular disks on the client domain, all disk operations are forwarded to the physical disk through the virtual disk service.

## Virtual Console

In a Logical Domains environment, console I/O from all domains, except the primary domain, is redirected to a service domain running the virtual console concentrator (`vcc`) and virtual network terminal server, instead of the systems controller. The virtual console concentrator service functions as a concentrator for all domains' console traffic, and interfaces with the virtual network terminal server daemon (`vntsd`) and provides access to each console through a UNIX socket.

## Dynamic Reconfiguration

Dynamic reconfiguration (DR) is the ability to add or remove resources while the operating system is running. The Solaris 10 OS supports only the adding and removing of a virtual CPU (`vcpu`). Dynamic reconfiguration of memory and input/output is not supported in the Solaris 10 OS. To use the dynamic reconfiguration capability in the Logical Domains Manager CLI, you must have the Logical Domains dynamic reconfiguration daemon, `drd(1M)` running in the domain you want to change.

## Delayed Reconfiguration

In contrast to dynamic reconfiguration operations that take place immediately, delayed reconfiguration operations take effect after the next reboot of the OS or stop and start of the logical domain if no OS is running. Any add or remove operations on active logical domains, except `add-vcpu`, `set-vcpu`, and `remove-vcpu` subcommands, are considered delayed reconfiguration operations. In addition, the `set-vswitch` subcommand on an active logical domain is considered a delayed reconfiguration operation.

If you are using a Sun UltraSPARC T1 processor, when the Logical Domains Manager is first installed and enabled (or when the configuration is restored to factory-default), the LDoms Manager runs in the configuration mode. In this mode, reconfiguration requests are accepted and queued up, but are not acted upon. This allows a new configuration to be generated and stored to the SC without affecting the state of the running machine, and therefore, without being encumbered by any of the restrictions around things like delayed reconfiguration and reboot of I/O domains.

Once a delayed reconfiguration is in progress for a particular logical domain, any other reconfiguration requests for that logical domain are also deferred until the domain is rebooted or stopped and started. Also, when there is a delayed reconfiguration outstanding for one logical domain, reconfiguration requests for other logical domains are severely restricted and will fail with an appropriate error message.

Even though attempts to remove virtual I/O devices on an active logical domain will be handled as a delayed reconfiguration operation, some configuration change does occur immediately. This means the device will in fact stop functioning as soon as the associated Logical Domains Manager CLI operation is invoked.

The Logical Domains Manager subcommand `remove-reconf` cancels delayed reconfiguration operations. You can list delayed reconfiguration operations by using the `ldm list-domain` command. Refer to the `ldm(1M)` man page or the *Logical Domains (LDoms) Manager Man Page Guide* for more information about how to use the delayed reconfiguration feature.

---

**Note** – You cannot use the `ldm remove-reconf` command if any other `ldm remove-*` commands have been issued on virtual I/O devices. The `ldm remove-reconf` command fails in these circumstances.

---

## Persistent Configurations

The current configuration of a logical domain can be stored on the system controller (SC) using the Logical Domains Manager CLI commands. You can add a configuration, specify a configuration to be used, remove a configuration, and list the configurations on the system controller. (Refer to the `ldm(1M)` man page or the *Logical Domains (LDoms) Manager Man Page Guide*.) In addition, there is an ALOM CMT Version 1.3 command that enables you to select a configuration to boot (see [“Using LDoms With ALOM CMT” on page 113](#)).



# Security

---

This chapter describes the Solaris Security Toolkit software and how you can use it to secure the Solaris OS in your logical domains.

---

## Security Considerations

The Solaris Security Toolkit software, informally known as the JumpStart™ Architecture and Security Scripts (JASS) toolkit, provides an automated, extensible, and scalable mechanism to build and maintain secure Solaris OS systems. The Solaris Security Toolkit provides security for devices critical to the management of your server, including the control domain in the Logical Domains Manager.

The Solaris Security Toolkit 4.2 software package, `SUNWjass`, provides the means to secure the Solaris Operating System on your control domain through the use of the `install-lbm` script by:

- Letting the Solaris Security Toolkit automatically harden your control domain by using the Logical Domains Manager install script (`install-lbm`) and the control driver specific to the Logical Domains Manager (`ldm_control-secure.driver`).
- Selecting an alternative driver when using the install script.
- Selecting no driver when using the install script and applying your own Solaris hardening.

The `SUNWjass` package is located with the Logical Domains (LDoms) Manager 1.0.3 software package, `SUNWldm`, at Sun's software download web site. You have the option to download and install the Solaris Security Toolkit 4.2 software package at the same time you download and install the Logical Domains Manager 1.0.3 software. The Solaris Security Toolkit 4.2 software package includes the required patches to enable the Solaris Security Toolkit software to work with the Logical

Domains Manager. Once the software is installed, you can harden your system with Solaris Security Toolkit 4.2 software. [Chapter 3](#) tells you how to install and configure the Solaris Security Toolkit, and harden your control domain.

Following are the security functions available to users of the Logical Domains Manager provided by the Solaris Security Toolkit:

- *Hardening* – Modifying Solaris OS configurations to improve a system’s security using the Solaris Security Toolkit 4.2 software with required patches to enable the Solaris Security Toolkit to work with the Logical Domains Manager.
- *Minimizing* – Installing the minimum number of core Solaris OS packages necessary for LDoms and LDoms Management Information Base (MIB) support.
- *Authorization* – Setting up authorization using the Solaris OS Role-Based Access Control (RBAC) adapted for the Logical Domains Manager.
- *Auditing* – Using the Solaris OS Basic Security module (BSM) adapted for the Logical Domains Manager to identify the source of security changes to the system to determine what was done, when it was done, by whom, and what was affected.
- *Compliance* – Determining if a system’s configuration is in compliance with a predefined security profile using the Solaris Security Toolkit’s auditing feature.

---

## Solaris Security Toolkit and the Logical Domains Manager

[Chapter 3](#) tells you how to install the Solaris Security Toolkit to make it work with the Logical Domains Manager. You would install the Solaris Security Toolkit on the control domain, which is where the Logical Domains Manager runs. You can also install the Solaris Security Toolkit on the other logical domains. The only difference would be that you would use the `ldm_control-secure.driver` to harden the control domain and you would use another driver, such as the `secure.driver`, to harden the other logical domains. This is because the `ldm_control-secure.driver` is specific to the control domain. The `ldm_control-secure.driver` is based on the `secure.driver` and has been customized and tested for use with the Logical Domains Manager. Refer to the *Solaris Security Toolkit 4.2 Reference Manual* for more information about the `secure.driver`.



# Hardening

The driver (`ldm_control-secure.driver`) that Solaris Security Toolkit uses to harden the Solaris OS on the control domain is specifically tailored so that the Logical Domains Manager can run with the OS. The `ldm_control-secure.driver` is analogous to the `secure.driver` described in the *Solaris Security Toolkit 4.2 Reference Manual*.

The `ldm_control-secure.driver` provides a baseline configuration for the control domain of a system running the Logical Domains Manager software. It is intended to provide fewer system services than typical for a Solaris OS domain, reserving the control domain for Logical Domains Manager operations, rather than general usage.

The `install-ldm` script installs the Logical Domains Manager software if it is not already installed, and enables the software.

Following is a short summary of the other notable changes from `secure.driver`.

- The Telnet server is disabled from running. You can use Secure Shell (`ssh`) instead. You also can still use the Telnet client to access virtual consoles started by the Logical Domains virtual network terminal server daemon (`vntsd`). For example, if a virtual console is running that is listening to TCP port 5001 on the local system, you can access it as follows.

```
# telnet localhost 5001
```

See “Enabling the Logical Domains Manager Daemon” on page 35 for instructions on enabling `vntsd`. It is not automatically enabled.

- The following finish scripts have been added. They enable the Logical Domains Manager to install and start. Some of these added scripts must be added to any customized drivers you make and some are optional. The scripts are marked as to whether they are required or optional.
  - `install-ldm.fin` – Installs the `SUNWldm` package. (*Required*)
  - `enable-ldmd.fin` – Enables the Logical Domains Manager daemon (`ldmd`) (*Required*)
  - `enable-ssh-root-login.fin` – Enables the superuser to directly log in through the Secure Shell (`ssh`). (*Optional*)
- The following files have changed. These changes are optional to make in any customized drivers you have and are marked as optional.
  - `/etc/ssh/sshd_config` – Root account access is allowed for the entire network. This file is not used in either driver. (*Optional*)
  - `/etc/ipf/ipf.conf` – UDP port 161 (SNMP) is opened. (*Optional*)
  - `/etc/host.allow` – The Secure Shell daemon (`sshd`) is open for the entire network, not just the local subnet. (*Optional*)

- The following finish scripts are disabled (commented out). You should comment out the `disable-rpc.fin` script in any customized driver you make. The other changes are optional. The scripts are marked as to whether they are required or optional.
  - `enable-ipfilter.fin` – IP Filter, a network packet filter, is not enabled. (*Optional*)
  - `disable-rpc.fin` – Leaves Remote Procedure Call (RPC) service enabled. The RPC service is used by many other system services, such as Network Information Service (NIS) and Network File System (NFS). (*Required*)
  - `disable-sma.fin` – Leaves the System Management Agent (NET-SNMP) enabled. (*Optional*)
  - `disable-ssh-root-login.fin` – ssh root login cannot be disabled.
  - `set-term-type.fin` – Unneeded legacy script. (*Optional*)

## Minimizing Logical Domains

The Solaris OS can be configured with different quantities of packages, depending on your needs. Minimization reduces this set of packages to the bare minimum required to run your desired applications. Minimization is important because it reduces the amount of software containing potential security vulnerabilities and also reduces the level of effort associated with keeping the installed software properly patched. The logical domain minimization activity provides JumpStart™ support for installing a minimized Solaris OS that still fully supports any domain.

The Solaris Security Toolkit provides a JumpStart profile, `minimal-ldm_control.profile`, for minimizing a logical domain for LDoms, which installs all the Solaris OS packages necessary for LDoms and LDoms MIB support. If you want to use the LDoms MIB on the control domain, you need to add that package separately after you install the LDoms and Solaris Security Toolkit packages. It is not installed automatically with the other software. Refer to the *Logical Domains (LDoms) MIB 1.0.1 Administration Guide* for more information about installing and using the LDoms MIB.

## Authorization

Authorization for the Logical Domains Manager has two levels:

- Read – allows you to view, but not modify the configuration.
- Read and write – allows you to view and change the configuration.

The changes are not made to the Solaris OS, but are added to the authorization file by the package script `postinstall` when the Logical Domains Manager is installed. Similarly, the authorization entries are removed by the package script `preremove`.

The following table lists the `ldm` subcommands with the corresponding user authorization that is needed to perform the commands.

**TABLE 2-1** The `ldm` Subcommands and User Authorizations

<b>ldm Subcommand*</b>	<b>User Authorization</b>
<code>add-*</code>	<code>solaris.ldoms.write</code>
<code>bind-domain</code>	<code>solaris.ldoms.write</code>
<code>list</code>	<code>solaris.ldoms.read</code>
<code>list-*</code>	<code>solaris.ldoms.read</code>
<code>panic-domain</code>	<code>solaris.ldoms.write</code>
<code>remove-*</code>	<code>solaris.ldoms.write</code>
<code>set-*</code>	<code>solaris.ldoms.write</code>
<code>start-domain</code>	<code>solaris.ldoms.write</code>
<code>stop-domain</code>	<code>solaris.ldoms.write</code>
<code>unbind-domain</code>	<code>solaris.ldoms.write</code>

\* Refers to all the resources you can add, list, remove, or set.

## Auditing

Auditing the Logical Domains Manager CLI commands is done with Solaris OS Basic Security module (BSM) auditing. Refer to the Solaris 10 *System Administration Guide: Security Services* for detailed information about using Solaris OS BSM auditing.

BSM auditing is not enabled by default for the Logical Domains Manager; however, the infrastructure is provided. You can enable BSM auditing in one of two ways:

- Run the `enable-bsm.fin` finish script in the Solaris Security Toolkit.
- Use the Solaris OS `bsmconv(1M)` command.

For further details about enabling, verifying, disabling, printing output, and rotating logs using BSM auditing with the Logical Domains Manager, see [“Enabling and Using BSM Auditing” on page 114](#).

# Compliance

Solaris Security Toolkit does have its own auditing capabilities. The Solaris Security Toolkit software can automatically validate the security posture of any system running the Solaris OS by comparing it with a predefined security profile. Refer to “Auditing System Security” in the *Solaris Security Toolkit 4.2 Administration Guide* for more information about this compliance function.

## Installing and Enabling Software

---

This chapter describes how to install and enable Logical Domains Manager 1.0.3 software and other software on a control domain on the supported servers. Refer to “Supported Servers” in the *Logical Domains (LDom)s 1.0.3 Release Notes* for a list of supported servers.

You can use what you need from this chapter depending on your platform. If you are using Logical Domains software on a new Sun UltraSPARC T2 platform, all the software should come preinstalled from the factory.

---

## Upgrading the Solaris OS

This section contains information you need to know about saving and restoring the Logical Domains constraints database file or performing a live upgrade on the control domain.

### Saving and Restoring the Logical Domains Constraints Database File

Whenever you upgrade the operating system on the control domain, you must save and restore the Logical Domains constraints database file that can be found in `/var/opt/SUNWldm/ldom-db.xml`.

---

**Note** – You must also save and restore the `/var/opt/SUNWldm/ldom-db.xml` file when you perform any other operation that is destructive to the control domain’s file data, such as a disk swap.

---

## Using Live Upgrade on the Control Domain

If you are using live upgrade on the control domain, consider adding the following line to the `/etc/lu/synclist` file:

<code>/var/opt/SUNWldm/ldom-db.xml</code>	<code>OVERWRITE</code>
---	------------------------

This causes the database to be copied automatically from the active boot environment to the new boot environment when switching boot environments. For more information about `/etc/lu/synclist` and synchronizing files between boot environments, refer to “Synchronizing Files Between Boot Environments” in the *Solaris 10 8/07 Installation Guide: Solaris Live Upgrade and Upgrade Planning*.

---

## Upgrading to LDoms 1.0.3 Software

Existing LDoms 1.0.1 and 1.0.2 configurations work in LDoms 1.0.3 software, so you do not need to perform the following procedure if you are upgrading from LDoms 1.0.1 or 1.0.2 software to LDoms 1.0.3 software. However, you do need to use the following procedure if you want to use your existing LDoms 1.0 configurations with LDoms 1.0.3 software.

### ▼ To Upgrade From LDoms 1.0 to LDoms 1.0.3 Software

Existing LDoms 1.0 configurations do *not* work in LDoms 1.0.3 software. The following procedure describes a method for saving and rebuilding a configuration using XML constraints files and the `-i` option to the `ldm start-domain` command. This method does not preserve actual bindings, only the constraints used to create those bindings. This means that, after this procedure, the domains will have the same virtual resources, but will not necessarily be bound to the same physical resources.

The basic process is to save the constraints information for each domain into an XML file, which can then be re-issued to the Logical Domains Manager after the upgrade to rebuild a desired configuration. This procedure works for guest domains, not the control domain. Although you can save the control (primary) domain’s constraints to an XML file, you cannot feed it back into the `ldm start-domain -i` command.

1. **Update to the latest version of the Solaris OS.** For more information, see Step 2, [“To Install the Solaris 10 OS”](#) on page 18.

2. For each domain, create an XML file containing the domain's constraints.

```
# ldm ls-constraints -x ldom > ldom.xml
```

3. List all the logical domain configurations stored on the system controller.

```
# ldm ls-config
```

4. Remove each logical domain configuration stored on the system controller.

```
# ldm rm-config config_name
```

5. Disable the Logical Domains Manager daemon (ldmd).

```
# svcadm disable ldmd
```

6. Remove the Logical Domains Manager package (SUNWldm).

```
# pkgrm SUNWldm
```

7. Remove the Solaris Security Toolkit package (SUNWjass) if you are using that.

```
# pkgrm SUNWjass
```

8. Flash update the system firmware. For the entire procedure, see [“To Upgrade System Firmware”](#) on page 19 or [“To Upgrade System Firmware Without an FTP Server”](#) on page 20.

9. Download the LDomS 1.0.3 software package.

See [“To Download the Logical Domains Manager, Solaris Security Toolkit, and Logical Domains MIB”](#) on page 21 for procedures for downloading and installing the Logical domains Manager, the Solaris Security Toolkit, and the Logical Domains MIB.

10. Reconfigure the primary domain manually. For instructions, see [“To Set Up the Control Domain”](#) on page 42.

11. Run the following commands for each guest domain's XML file you created in Step 2.

```
# ldm create -i ldom.xml
# ldm bind-domain ldom
# ldm start-domain ldom
```

---

# Freshly Installing Software on the Control Domain

The first domain that is created when the Logical Domains Manager software is installed is the control domain. That first domain is named `primary`, and you cannot change the name. The following major components are installed on the control domain.

- Solaris 10 OS. Add any patches recommended in the *Logical Domains (LDoms) 1.0.3 Release Notes*, if necessary. See [“To Install the Solaris 10 OS” on page 18](#).
- System firmware version 6.5 for your Sun UltraSPARC T1 platform or system firmware version 7.0 for your Sun UltraSPARC T2 platform. See [“To Upgrade System Firmware” on page 19](#).
- Logical Domain Manager 1.0.3 software. See [“Installing Logical Domains Manager and Solaris Security Toolkit” on page 22](#).
- (Optional) Solaris Security Toolkit 4.2 software. See [“Installing Logical Domains Manager and Solaris Security Toolkit” on page 22](#).
- (Optional) Logical Domains (LDoms) Management Information Base (MIB) software package. Refer to the *Logical Domains (LDoms) Management Information Base (MIB) 1.0.1 Administration Guide* for more information about installing and using the LDoms MIB.

The Solaris OS and the system firmware must be installed on your server before you install the Logical Domains Manager. After the Solaris OS, the system firmware, and the Logical Domains Manager have been installed, the original domain becomes the control domain.

## ▼ To Install the Solaris 10 OS

Install the Solaris 10 OS if it has not already been installed. Refer to “Required and Recommended Software” in the *Logical Domains (LDoms) 1.0.3 Release Notes* to find the Solaris 10 OS that you should use for this version of the Logical Domains software. Refer to your Solaris 10 OS installation guide for complete instructions for installing the Solaris OS. You can tailor your installation to the needs of your system.

---

**Note** – For logical domains, you can install the Solaris OS only to an entire disk or a file exported as a block device.

---



### 1. Install the Solaris 10 OS.

Minimization is optional. The Solaris Security Toolkit has the following JumpStart minimization profile for Logical Domains software:

```
/opt/SUNWjass/Profiles/minimal-ldm_control.profile
```

### 2. Install the required patches if you are installing the Solaris 10 11/06 OS. Refer to “Required Solaris 10 11/06 OS Patches” in the *Logical Domains (LDDoms) 1.0.3 Release Notes* for the list of required patches.

---

**Note** – If you are installing an operating system in non-English languages in a guest domain, the terminal for the console must be in the locale required by the OS installer. For example, the Solaris OS installer requires EUC locales, while the Linux installer might need Unicode locales.

---

## ▼ To Upgrade System Firmware

You can find system firmware for your platform at the SunSolve site:

```
http://sunsolve.sun.com
```

Refer to “Required System Firmware Patches” in the *Logical Domains (LDDoms) 1.0.3 Release Notes* for required system firmware by supported servers.

This procedure describes how to upgrade system firmware using the `flashupdate(1M)` command on your system controller.

- If you do not have access to a local FTP server, see “To Upgrade System Firmware Without an FTP Server” on page 20.
- If you want to update the system firmware from the control domain, refer to your system firmware release notes.

Refer to the administration guides or product notes for the supported servers for more information about installing and updating system firmware for these servers.

### 1. Shut down and power off the host server from either management port connected to the system controller: serial or network.

```
# shutdown -i5 -g0 -y
```

2. Use the `flashupdate(1M)` command to upgrade the system firmware, depending on your server.

```
sc> flashupdate -s IP-address -f path/Sun_System_Firmware-  
x_x_x_build_nn-server-name.bin  
username: your-userid  
password: your-password
```

Where:

- *IP-address* is the IP address of your FTP server.
- *path* is the location in SunSolve<sup>sm</sup> or your own directory where you can obtain the system firmware image.
- *x\_x\_x* is the version number of the System Firmware.
- *nn* is the number of the build that applies to this release.
- *server-name* is the name of your server. For example, the *server-name* for the Sun Fire T2000 server is `Sun_Fire_T2000`.

3. Reset the system controller.

```
sc> resetsc -y
```

4. Power on and boot the host server.

```
sc> poweron -c  
ok boot disk
```

## ▼ To Upgrade System Firmware Without an FTP Server

If you do not have access to a local FTP server to upload firmware to the system controller, you can use the `sysfwdownload` utility, which is provided with your system firmware upgrade package on the SunSolve site:

<http://sunsolve.sun.com>

1. Run the following commands within the Solaris OS.

```
# cd firmware_location  
# sysfwdownload system_firmware_file
```

2. Shut down the Solaris OS instance.

```
# shutdown -i5 -g0 -y
```

3. Power off and update the firmware on the system controller.

```
SC> poweroff -fy
SC> flashupdate -s 127.0.0.1
```

4. Reset and power on the system controller.

```
SC> resetsc -y
SC> poweron
```

## ▼ To Downgrade System Firmware

Once you have upgraded the system firmware for use with Logical Domains software, you can downgrade the firmware to the original non-Logical Domains firmware.

- Run the `flashupdate(1M)` command and specify the path to the original non-Logical Domains firmware.

---

# Downloading Logical Domains Manager and Solaris Security Toolkit

## ▼ To Download the Logical Domains Manager, Solaris Security Toolkit, and Logical Domains MIB

1. Download the tar file (`LDoms_Manager-1_0_3-04.zip`) containing the Logical Domains Manager package (`SUNWldm`), the Solaris Security Toolkit (`SUNWjass`) and installation script (`install-ldm`), and the Logical Domains Management Information Base package (`SUNWldmib.v`) from the Sun Software Download site. You can find the software from this web site:

<http://www.sun.com/ldoms>

## 2. Unzip the zip file.

```
$ unzip LDoms_Manager-1_0_3-04.zip
```

The directory structure for the downloaded software is similar to the following:

### CODE EXAMPLE 3-1 Directory Structure for Downloaded Logical Domains 1.0.3 Software

```
LDoms_Manager-1_0_3/  
  Install/  
    install-ldm  
  Legal/  
    LDoms_1.0.3_Entitlement.txt  
    LDoms_1.0.3_SLA_Entitlement.txt  
  Product/  
    SUNWjass/  
    SUNWldm.v/  
    SUNWldmib.v  
  README
```

---

# Installing Logical Domains Manager and Solaris Security Toolkit

There are three methods of installing Logical Domains Manager and Solaris Security Toolkit software:

- Using the installation script to install the packages and patches. This automatically installs both the Logical Domains Manager and the Solaris Security Toolkit software. See [“Using the Installation Script to Install the Logical Domains Manager 1.0.3 and Solaris Security Toolkit 4.2 Software”](#) on page 23.
- Using JumpStart to install the packages. See [“Using JumpStart to Install the Logical Domains Manager 1.0.3 and Solaris Security Toolkit 4.2 Software”](#) on page 29.
- Installing each package manually. See [“Installing Logical Domains Manager and Solaris Security Toolkit Software Manually”](#) on page 32.

---

**Note** – Remember that you need to manually install the LDoms MIB software package after you install the LDoms and Solaris Security Toolkit packages. It is not automatically installed with the other packages. Refer to the *Logical Domains (LDoms) Management Information Base 1.0.1 Administration Guide* for more information about installing and using the LDoms MIB.

---

# Using the Installation Script to Install the Logical Domains Manager 1.0.3 and Solaris Security Toolkit 4.2 Software

If you use the `install-ldm` installation script, you have several choices to specify how you want the script to run. Each choice is described in the procedures that follow.

- **Using the `install-ldm` script with no options does the following automatically:**
  - Checks that the Solaris OS release is Solaris 10 11/06
  - Verifies that the package subdirectories `SUNWldm/` and `SUNWjass/` are present
  - Verifies that the prerequisite Solaris Logical Domains driver packages, `SUNWldomr` and `SUNWldomu`, are present
  - Verifies that the `SUNWldm` and `SUNWjass` packages have not been installed

---

**Note** – If the script does detect a previous version of `SUNWjass` during installation, you will need to remove it. You do *not* need to undo any previous hardening of your Solaris OS.

---

- Installs the Logical Domains Manager 1.0.3 software (`SUNWldm` package)
- Installs the Solaris Security Toolkit 4.2 software including required patches (`SUNWjass` package)
- Verifies that all packages are installed
- Enables the Logical Domains Manager daemon, `ldmd`
- Hardens the Solaris OS on the control domain with the Solaris Security Toolkit `ldm_control-secure.driver` or one of the other drivers ending in `-secure.driver` that you select.
- **Using the `install-ldm` script with option `-d`** allows you to specify a Solaris Security Toolkit driver other than a driver ending with `-secure.driver`. This option automatically performs all the functions listed in the preceding choice with the added option:
  - Hardens the Solaris OS on the control domain with the Solaris Security Toolkit customized driver that you specify; for example, the `server-secure-myname.driver`.
- **Using the `install-ldm` script with option `-d` and specifying `none`** specifies that you do *not* want to harden the Solaris OS running on your control domain by using the Solaris Security Toolkit. This option automatically performs all the functions except hardening listed in the preceding choices. Bypassing the use of the Solaris Security Toolkit is not suggested and should only be done when you intend to harden your control domain using an alternate process.

- **Using the `install-ldm` script with option `-p`** specifies that you only want to perform the post-installation actions of enabling the Logical Domains Manager daemon (`ldmd`) and running the Solaris Security Toolkit. For example, you would use this option if the `SUNWldm` and `SUNWjass` packages are preinstalled on your server. See [“To Install Using the `install-ldm` Script With the `-p` Option” on page 29](#)

## ▼ To Install Using the `install-ldm` Script With No Options

- **Run the installation script with no options.**

The installation script is part of the `SUNWldm` package and is in the `Install` subdirectory.

```
# Install/install-ldm
```

- a. **If one or more packages are previously installed, you receive this message.**

```
# Install/install-ldm
ERROR: One or more packages are already installed: SUNWldm SUNWjass.
If packages SUNWldm.v and SUNWjass are factory pre-installed, run
install-ldm -p to perform post-install actions. Otherwise remove the
package(s) and restart install-ldm.
```

If you want to perform post-installation actions only, go to [“To Install Using the `install-ldm` Script With the `-p` Option” on page 29](#).

- b. **If the process is successful, you receive messages similar to the following examples.**

- Code Example 3-2 shows a successful run of the `install-ldm` script if you choose the following default security profile:
  - a) Hardened Solaris configuration for LDomS (recommended)
- Code Example 3-3 shows a successful run of the `install-ldm` script if you choose the following security profile:
  - c) Your custom-defined Solaris security configuration profile

The drivers that are displayed for you to choose are drivers ending with `-secure.driver`. If you write a customized driver that does not end with `-secure.driver`, you must specify your customized driver with the `install-ldm -d` option. (See [“To Install Using the `install-ldm` Script With the `-d` Option” on page 27.](#))

**CODE EXAMPLE 3-2** Output From Hardened Solaris Configuration for LDoms

```
# Install/install-ldm
Welcome to the LDoms installer.

You are about to install the domain manager package that will enable
you to create, destroy and control other domains on your system. Given
the capabilities of the domain manager, you can now change the security
configuration of this Solaris instance using the Solaris Security
Toolkit.

Select a security profile from this list:

a) Hardened Solaris configuration for LDoms (recommended)
b) Standard Solaris configuration
c) Your custom-defined Solaris security configuration profile

Enter a, b, or c [a]: a
The changes made by selecting this option can be undone through the
Solaris Security Toolkit's undo feature. This can be done with the
'/opt/SUNWjass/bin/jass-execute -u' command.

Installing LDoms and Solaris Security Toolkit packages.
pkgadd -n -d "/var/tmp/install/Product/Logical_Domain_Manager" -a pkg_admin
SUNWldm.v
Copyright 2006 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Installation of <SUNWldm> was successful.
pkgadd -n -d "/var/tmp/install/Product/Solaris_Security_Toolkit" -a pkg_admin
SUNWjass
Copyright 2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Installation of <SUNWjass> was successful.

Verifying that all packages are fully installed. OK.
Enabling services: svc:/ldoms/ldmd:default
Running Solaris Security Toolkit 4.2.0 driver ldm_control-secure.driver.
Please wait. . .
/opt/SUNWjass/bin/jass-execute -q -d ldm_control-secure.driver
Executing driver, ldm_control-secure.driver
Solaris Security Toolkit hardening executed successfully; log file
```

**CODE EXAMPLE 3-2** Output From Hardened Solaris Configuration for LDomS (*Continued*)

```
/var/opt/SUNWjass/run/20070208142843/jass-install-log.txt. It will not
take effect until the next reboot. Before rebooting, make sure SSH or
the serial line is setup for use after the reboot.
```

**CODE EXAMPLE 3-3** Output From Choosing Customized Configuration Profile

```
# Install/install-ldm
Welcome to the LDomS installer.

You are about to install the domain manager package that will enable
you to create, destroy and control other domains on your system. Given
the capabilities of the domain manager, you can now change the security
configuration of this Solaris instance using the Solaris Security
Toolkit.

Select a security profile from this list:

a) Hardened Solaris configuration for LDomS (recommended)
b) Standard Solaris configuration
c) Your custom-defined Solaris security configuration profile

Enter a, b, or c [a]: c
Choose a Solaris Security Toolkit .driver configuration profile from
this list
1) ldm_control-secure.driver
2) secure.driver
3) server-secure.driver
4) suncluster3x-secure.driver
5) sunfire_15k_sc-secure.driver

Enter a number 1 to 5: 2
The driver you selected may not perform all the LDomS-specific
operations specified in the LDomS Administration Guide.
Is this OK (yes/no)? [no] y
The changes made by selecting this option can be undone through the
Solaris Security Toolkit's undo feature. This can be done with the
'/opt/SUNWjass/bin/jass-execute -u' command.

Installing LDomS and Solaris Security Toolkit packages.
pkgadd -n -d "/var/tmp/install/Product/Logical_Domain_Manager" -a pkg_admin
SUNWldm.v
Copyright 2006 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Installation of <SUNWldm> was successful.
pkgadd -n -d "/var/tmp/install/Product/Solaris_Security_Toolkit" -a pkg_admin
SUNWjass
Copyright 2005 Sun Microsystems, Inc. All rights reserved.
```



**CODE EXAMPLE 3-3** Output From Choosing Customized Configuration Profile (*Continued*)

```
Use is subject to license terms.

Installation of <SUNWjass> was successful.

Verifying that all packages are fully installed. OK.
Enabling services: svc:/ldoms/ldmd:default
Running Solaris Security Toolkit 4.2.0 driver secure.driver.
Please wait. . .
/opt/SUNWjass/bin/jass-execute -q -d secure.driver
Executing driver, secure.driver
Solaris Security Toolkit hardening executed successfully; log file
/var/opt/SUNWjass/run/20070102142843/jass-install-log.txt. It will not
take effect until the next reboot. Before rebooting, make sure SSH or
the serial line is setup for use after the reboot.
```

## ▼ To Install Using the `install-ldm` Script With the `-d` Option

- **Run the installation script with the `-d` option to specify a Solaris Security Toolkit customized hardening driver; for example, `server-secure-myname.driver`.**

The installation script is part of the `SUNWldm` package and is in the `Install` subdirectory.

```
# Install/install-ldm -d server-secure-myname.driver
```

If the process is successful, you receive messages similar to that in Code Example 3-4.

**CODE EXAMPLE 3-4** Output From Successful Run of the `install-ldm -d` Script

```
# Install/install-ldm -d server-secure.driver
The driver you selected may not perform all the LDoms-specific
operations specified in the LDoms Administration Guide.
Installing LDoms and Solaris Security Toolkit packages.
pkgadd -n -d "/var/tmp/install/Product/Logical_Domain_Manager" -a pkg_admin
SUNWldm.v
Copyright 2006 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Installation of <SUNWldm> was successful.
pkgadd -n -d "/var/tmp/install/Product/Solaris_Security_Toolkit" -a pkg_admin
SUNWjass
Copyright 2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
```

**CODE EXAMPLE 3-4** Output From Successful Run of the `install-ldm -d` Script (*Continued*)

```
Installation of <SUNWjass> was successful.

Verifying that all packages are fully installed.  OK.
Enabling services: svc:/ldoms/ldmd:default
Running Solaris Security Toolkit 4.2.0 driver server-secure-myname.driver.
Please wait. . .
/opt/SUNWjass/bin/jass-execute -q -d server-secure-myname.driver
Executing driver, server-secure-myname.driver
Solaris Security Toolkit hardening executed successfully; log file
/var/opt/SUNWjass/run/20061114143128/jass-install-log.txt.  It will not
take effect until the next reboot.  Before rebooting, make sure SSH or
the serial line is setup for use after the reboot.
```

## ▼ To Install Using the `install-ldm` Script With the `-d none` Option

- **Run the installation script with the `-d none` option to specify *not* to harden your system using a Solaris Security Toolkit driver.**

The installation script is part of the `SUNWldm` package and is in the `Install` subdirectory.

```
# Install/install-ldm -d none
```

If the process is successful, you receive messages similar to the example shown in Code Example 3-5.

**CODE EXAMPLE 3-5** Output From Successful Run of the `install-ldm -d none` Script

```
# Install/install-ldm -d none
Installing LDoms and Solaris Security Toolkit packages.
pkgadd -n -d "/var/tmp/install/Product/Logical_Domain_Manager" -a pkg_admin
SUNWldm.v
Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.

Installation of <SUNWldm> was successful.
pkgadd -n -d "/var/tmp/install/Product/Solaris_Security_Toolkit" -a pkg_admin
SUNWjass
Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.

Installation of <SUNWjass> was successful.

Verifying that all packages are fully installed.  OK.
```

**CODE EXAMPLE 3-5** Output From Successful Run of the `install-ldm -d none` Script (*Continued*)

```
Enabling services: svc:/ldoms/ldmd:default
Solaris Security Toolkit was not applied. Bypassing the use of the
Solaris Security Toolkit is not recommended and should only be
performed when alternative hardening steps are to be taken.
```

## ▼ To Install Using the `install-ldm` Script With the `-p` Option

You might use this option if the `SUNWldm` and `SUNWjass` packages are preinstalled on your server and you want to perform the post-installation actions of enabling the Logical Domains Manager daemon (`ldmd`) and running the Solaris Security Toolkit.

- **Run the installation script with the `-p` option to perform only the post-installation actions of enabling `ldmd` and running the Solaris Security Toolkit to harden your system.**

```
# Install/install-ldm -p
Verifying that all packages are fully installed.  OK.
Enabling services: svc:/ldoms/ldmd:default
Running Solaris Security Toolkit 4.2.0 driver ldm_control-secure.driver.
Please wait. . .
/opt/SUNWjass/bin/jass-execute -q -d ldm_control-secure.driver
Solaris Security Toolkit hardening executed successfully; log file
var/opt/SUNWjass/run/20070515140944/jass-install-log.txt.  It will not
take effect until the next reboot.  Before rebooting, make sure SSH or
the serial line is setup for use after the reboot.
```

## Using JumpStart to Install the Logical Domains Manager 1.0.3 and Solaris Security Toolkit 4.2 Software

Refer to *JumpStart Technology: Effective Use in the Solaris Operating Environment* for complete information about using JumpStart.



---

**Caution** – Do *not* disconnect from the virtual console during a network installation.

---

## ▼ To Set Up a JumpStart Server

- If you have already set up a JumpStart server, proceed to [“To Install Using JumpStart Software” on page 30](#) of this administration guide.

- If you have not already set up a JumpStart server, you must do so.

Refer to the *Solaris 10 11/06 Installation Guide: Custom JumpStart and Advanced Installation* for complete information about this procedure. You can find this installation guide at:

<http://docs.sun.com/app/docs/doc/819-6397>

1. Refer to Chapter 3 “Preparing Custom JumpStart Installations (Tasks)” in the *Solaris 10 11/06 Installation Guide: Custom JumpStart and Advanced Installation*, and perform the following steps.

- a. Read the task map in “Task Map: Preparing Custom JumpStart Installations.”

- b. Set up networked systems with the procedures in “Creating a Profile Server for Network Systems.”

- c. Create the `rules` file with the procedure in “Creating the `rules` File.”

2. Validate the `rules` file with the procedure in “Validating the `rules` File.”

The Solaris Security Toolkit provides profiles and finish scripts. Refer to the *Solaris Security Toolkit 4.2 Reference Manual* for more information about profiles and finish scripts.

## ▼ To Install Using JumpStart Software

1. Change to the directory where you have downloaded the Solaris Security Toolkit package (`SUNWjass`).

```
# cd /path-to-download
```

2. Install `SUNWjass` so that it creates the JumpStart (`jumpstart`) directory structure.

```
# pkgadd -R /jumpstart -d . SUNWjass
```

3. Use your text editor to modify the `/jumpstart/opt/SUNWjass/Sysidcfg/Solaris_10/sysidcfg` file to reflect your network environment.

4. Copy the `/jumpstart/opt/SUNWjass/Drivers/user.init.SAMPLE` file to the `/jumpstart/opt/SUNWjass/Drivers/user.init` file.

```
# cp user.init.SAMPLE user.init
```

5. Edit the `user.init` file to reflect your paths.
6. To install the Solaris Security Toolkit package (`SUNWjass`) onto the target system during a JumpStart install, you must place the package in the `JASS_PACKAGE_MOUNT` directory defined in your `user.init` file. For example:

```
# cp -r /path/to/LDoms_Manager-1_0_2/Product/SUNWjass  
/jumpstart/opt/SUNWjass/Packages
```

7. To install the Logical Domains Manager package (`SUNWldm.v`) onto the target system during a JumpStart install, you must place the package from the download area in the `JASS_PACKAGE_MOUNT` directory defined in your `user.init` file. For example:

```
# cp -r /path/to/LDoms_Manager-1_0_2/Product/SUNWldm.v  
/jumpstart/opt/SUNWjass/Packages
```

8. If you experience problems with a multihomed JumpStart server, modify the two entries in the `user.init` file for `JASS_PACKAGE_MOUNT` and `JASS_PATCH_MOUNT` to the correct path to the `JASS_HOME_DIR/Patches` and `JASS_HOME_DIR/Packages` directories. Refer to the comments in the `user.init.SAMPLE` file for more information.
9. Use the `ldm_control-secure.driver` as the basic driver for the Logical Domains Manager control domain.  
Refer to Chapter 4 in the *Solaris Security Toolkit 4.2 Reference Manual* for information about how to modify the driver for your use. The main driver in the Solaris Security Toolkit that is the counterpart to the `ldm_control-secure.driver` is the `secure.driver`.
10. After completing the modifications to the `ldm_control-secure.driver`, make the correct entry in the rules file.
  - If you want to minimize the LDoms control domain, specify the `minimal-ldm-control.profile` in your rules file similar to the following.

```
hostname imbulu - Profiles/minimal-ldm_control.profile  
Drivers/ldm_control-secure-abc.driver
```

---

**Note** – Remember that you need to manually install the LDoms MIB software package after you install the LDoms and Solaris Security Toolkit packages. It is not automatically installed with the other packages. Refer to the *Logical Domains (LDoms) Management Information Base 1.0.1 Administration Guide* for more information about installing and using the LDoms MIB.

---

- If you do not want to minimize the LDoms control domain, your entry should be similar to the following.

```
hostname imbulu - Profiles/oem.profile Drivers/ldm_control-secure-abc.driver
```

11. If you undo hardening during a JumpStart install, you must run the following SMF command to restart the Logical Domains Manager.

```
# svcadm enable svc:/ldoms/ldmd:default
```

## Installing Logical Domains Manager and Solaris Security Toolkit Software Manually

Perform the following procedures to install the Logical Domains Manager and Solaris Security Toolkit Software manually:

- [“To Install the Logical Domains Manager \(LDoms\) 1.0.3 Software Manually”](#) on page 32.
- [“\(Optional\) To Install the Solaris Security Toolkit 4.2 Software Manually”](#) on page 33.
- [“\(Optional\) To Harden the Control Domain Manually”](#) on page 33.

### ▼ To Install the Logical Domains Manager (LDoms) 1.0.3 Software Manually

Download the Logical Domains Manager 1.0.3 software, the `SUNWldm` package, from the Sun Software Download site. See [“To Download the Logical Domains Manager, Solaris Security Toolkit, and Logical Domains MIB”](#) on page 21 for specific instructions.

1. Use the `pkgadd(1M)` command to install the `SUNWldm.v` package. Use the `-G` option to install the package in the global zone only and the `-d` option to specify the path to the directory that contains the `SUNWldm.v` package.

```
# pkgadd -Gd . SUNWldm.v
```

2. Answer `y` for yes to all questions in the interactive prompts.

3. Use the `pkginfo(1)` command to verify that the `SUNWldm` package for Logical Domains Manager 1.0.3 software is installed.

The revision (REV) information shown below is an example.

```
# pkginfo -l SUNWldm | grep VERSION
VERSION=1.0.3,REV=2007.08.23.10.20
```

## ▼ (Optional) To Install the Solaris Security Toolkit 4.2 Software Manually

If you want to secure your system, download and install the `SUNWjass` package. The required patches (122608-03 and 125672-01) are included in the `SUNWjass` package. See [“To Download the Logical Domains Manager, Solaris Security Toolkit, and Logical Domains MIB” on page 21](#) for specific instructions about downloading the software.

See [Chapter 2](#) in this document for more information about security considerations when using Logical Domains Manager software. For further reference, you can find Solaris Security Toolkit 4.2 documentation at:

<http://docs.sun.com>

1. Use the `pkgadd(1M)` command to install the `SUNWjass` package.

```
# pkgadd -d . SUNWjass
```

2. Use the `pkginfo(1)` command to verify that the `SUNWjass` package for Solaris Security Toolkit 4.2 software is installed.

```
# pkginfo -l SUNWjass | grep VERSION
VERSION: 4.2.0
```

## ▼ (Optional) To Harden the Control Domain Manually

Perform this procedure only if you have installed the Solaris Security Toolkit 4.2 package.

---

**Note** – When you use the Solaris Security Toolkit to harden the control domain, you disable many system services and place certain restrictions on network access. Refer to [“Related Documentation” on page xix](#) in this book to find Solaris Security Toolkit 4.2 documentation for more information.

---

1. Harden using the `ldm_control-secure.driver`.

```
# /opt/SUNWjass/bin/jass-execute -d ldm_control-secure.driver
```

You can use other drivers to harden your system. You can also customize drivers to tune the security of your environment. Refer to the *Solaris Security Toolkit 4.2 Reference Manual* for more information about drivers and customizing them.

2. Answer `y` for yes to all questions in the interactive prompts.
3. Shut down and reboot your server for the hardening to take place.

```
# /usr/sbin/shutdown -y -g0 -i6
```

## ▼ To Validate Hardening

- Check whether the Logical Domains hardening driver (`ldm_control-secure.driver`) applied hardening correctly.

If you want to check on another driver, substitute that driver's name in this command example.

```
# /opt/SUNWjass/bin/jass-execute -a ldm_control-secure.driver
```

## ▼ To Undo Hardening

1. Undo the configuration changes applied by the Solaris Security Toolkit.

```
# /opt/SUNWjass/bin/jass-execute -u
```

The Solaris Security Toolkit asks you which hardening runs you want to undo.

2. Select the hardening runs you want to undo.
3. Reboot the system so that the unhardened configuration takes place.

```
# /usr/sbin/shutdown -y -g0 -i6
```

---

**Note** – If you undo hardening that was performed during a JumpStart installation, you must run the following SMF commands to restart the Logical Domains Manager and the Virtual Network Terminal Server Daemon.

---

```
# svcadm enable svc:/ldoms/ldmd:default
```



---

# Enabling the Logical Domains Manager Daemon

The installation script `install-ldm` automatically enables the Logical Domains Manager Daemon (`ldmd`). If you have installed the Logical Domains Manager software manually, you must enable the Logical Domains Manager daemon, `ldmd`, which allows you to create, modify, and control the logical domains.

## ▼ To Enable the Logical Domains Manager Daemon

1. Use the `svcadm(1M)` command to enable the Logical Domains Manager daemon, `ldmd`.

```
# svcadm enable ldmd
```

2. Use the `ldm list` command to verify that the Logical Domains Manager is running.

You receive a message similar to the following, which is for the `factory-default` configuration. Note that the primary domain is active, which means that the Logical Domains Manager is running.

```
# /opt/SUNWldm/bin/ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	---c-	SP	32	3264M	0.3%	19d 9m

---

# Creating Authorization and Profiles and Assigning Roles for User Accounts

You set up authorization and profiles and assign roles for user accounts using the Solaris OS Role-Based Access Control (RBAC) adapted for the Logical Domains Manager. Refer to the Solaris 10 System Administrator Collection for more information about RBAC.

Authorization for the Logical Domains Manager has two levels:

- Read – allows you to view, but not modify the configuration.

- Read and write – allows you to view and change the configuration.

Following are the Logical Domains entries automatically added to the Solaris OS `/etc/security/auth_attr` file:

- `solaris.ldoms:::LDom administration::`
- `solaris.ldoms.grant:::Delegate LDom configuration::`
- `solaris.ldoms.read:::View LDom configuration::`
- `solaris.ldoms.write:::Manage LDom configuration::`

## Managing User Authorizations

### ▼ To Add an Authorization for a User

Use the following steps as necessary to add authorizations in the `/etc/security/auth_attr` file for Logical Domains Manager users. Because the superuser already has `solaris.*` authorization, the superuser already has permission for `solaris.ldoms.*` authorizations.

1. **Create a local user account for each user who needs authorization to use the `ldm(1M)` subcommands.**

---

**Note** – To add Logical Domains Manager authorization for a user, a local (non-LDAP) account must be created for that user. Refer to the Solaris 10 System Administrator Collection for details.

---

2. **Do one of the following depending on which `ldm(1M)` subcommands you want the user to be able to access.**

See [TABLE 2-1](#) for a list of `ldm(1M)` commands and their user authorizations.

- Add a read-only authorization for a user using the `usermod(1M)` command.

```
# usermod -A solaris.ldoms.read username
```

- Add a read and write authorization for a user using the `usermod(1M)` command.

```
# usermod -A solaris.ldoms.write username
```

## ▼ To Delete All Authorizations for a User

- Delete all authorizations for a local user account (the only possible option).

```
# usermod -A `` username
```

## Managing User Profiles

The SUNWldm package adds two system-defined RBAC profiles in the `/etc/security/prof_attr` file for use in authorizing access to the Logical Domains Manager by non-superusers. The two LDoms-specific profiles are:

- LDoms Review:::Review LDoms configuration:auths=solaris.ldoms.read
- LDoms Management:::Manage LDoms domains:auths=solaris.ldoms.\*

One of the preceding profiles can be assigned to a user account using the following procedure.

## ▼ To Add a Profile for a User

- Add an administrative profile for a local user account; for example, LDoms Management.

```
# usermod -P "LDoms Management" username
```

## ▼ To Delete All Profiles for a User

- Delete all profiles for a local user account (the only possible option).

```
# usermod -P `` username
```

## Assigning Roles to Users

The advantage of using this procedure is that only a user who has been assigned a specific role can assume the role. In assuming a role, a password is required if the role is given a password. This provides two layers of security. If a user has not been assigned a role, then the user cannot assume the role (by doing the `su role_name` command) even if the user has the correct password.

## ▼ To Create a Role and Assign the Role to a User

1. Create a role.

```
# roleadd -A solaris.ldoms.read ldm_read
```

2. Assign a password to the role.

```
# passwd ldm_read
```

3. Assign the role to a user; for example, user\_1.

```
# useradd -R ldm_read user_1
```

4. Assign a password to the user (user\_1).

```
# passwd user_1
```

5. Assign access only to the user\_1 account to become the ldm\_read account.

```
# su user_1
```

6. Type the user password when or if prompted.

7. Verify the user ID and access to the ldm\_read role.

```
$ id
uid=nn(user_1) gid=nn(<group name>)
$ roles
ldm_read
```

8. Provide access to the user for ldm subcommands that have read authorization.

```
# su ldm_read
```

9. Type the user password when or if prompted.

10. Type the id command to show the user.

```
$ id
uid=nn(ldm_read) gid=nn(<group name>)
```

# Setting Up Services and Logical Domains

---

This chapter describes the procedures necessary to set up default services, your control domain, and guest domains.

---

## Output Messages

You receive different output messages from the commands you use to create default services and to set up the control (`primary`) domain depending on your platform:

- Sun UltraSPARC T1 processors
- Sun UltraSPARC T2 processors

### Sun UltraSPARC T1 Processors

You receive the following notice after the setup commands for the `primary` domain if you are using a server with a Sun UltraSPARC T1 processor:

Notice: the LDom Manager is running in configuration mode. Any configuration changes made will only take effect after the machine configuration is downloaded to the system controller and the host is reset.

# Sun UltraSPARC T2 Processors

**First Operation** – You receive the following message after the first operation on any device or for any service on the `primary` domain if you are using a server with a Sun UltraSPARC T2 processor:

```
Initiating delayed reconfigure operation on LDom primary. All
configuration changes for other LDom's are disabled until the
LDom reboots, at which time the new configuration for LDom
primary will also take effect.
```

**Subsequent Operations Until Reboot** – You receive the following notice after every subsequent operation on the `primary` domain until reboot if you are using a server with a Sun UltraSPARC T2 processor:

```
Notice: LDom primary is in the process of a delayed
reconfiguration. Any changes made to this LDom will only take
effect after it reboots.
```

---

## Creating Default Services

You must create the following virtual default services initially to be able to use them later:

- `vdiskserver` – virtual disk server
- `vswitch` – virtual switch service
- `vconscon` – virtual console concentrator service

### ▼ To Create Default Services

1. **Create a virtual disk server (`vds`) to allow importing virtual disks into a logical domain.**

For example, the following command adds a virtual disk server (`primary-vds0`) to the control domain (`primary`).

```
primary$ ldm add-vds primary-vds0 primary
```

2. Create a virtual console concentrator service (vcc) for use by the virtual network terminal server daemon (vntsd) and as a concentrator for all logical domain consoles.

For example, the following command would add a virtual console concentrator service (primary-vcc0) with a port range from 5000 to 5100 to the control domain (primary).

```
primary$ ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

3. Create a virtual switch service (vsw) to enable networking between virtual network (vnet) devices in logical domains. Assign a GLDv3-compliant network adapter to the virtual switch if each of the logical domains needs to communicate outside the box through the virtual switch.

For example, the following command would add a virtual switch service (primary-vsw0) on network adapter driver e1000g0 to the control domain (primary).

```
primary$ ldm add-vsw net-dev=e1000g0 primary-vsw0 primary
```

This command automatically allocates a MAC address to the virtual switch. You can specify your own MAC address as an option to the `ldm add-vsw` command. However, in that case, it is your responsibility to ensure that the MAC address specified does not conflict with an already existing MAC address.

If the virtual switch being added replaces the underlying physical adapter as the primary network interface, it must be assigned the MAC address of the physical adapter, so that the Dynamic Host Configuration Protocol (DHCP) server assigns the domain the same IP address. See [“Enabling Networking Between the Control/Service Domain and Other Domains”](#) on page 45.

```
primary$ ldm add-vsw mac-addr=2:04:4f:fb:9f:0d net-dev=e1000g0 primary-vsw0
primary
```

---

**Note** – Be sure you do a reconfiguration reboot any time you add a virtual switch device as part of the domain reconfiguration. This is normally the case during the setting up of the control domain. See [“Rebooting to Use Logical Domains”](#) on page 44 for more information.

---

4. Verify the services have been created by using the `list-services` subcommand. Your output should look similar to the following.

```
primary$ ldm list-services primary
VDS
  NAME                VOLUME    OPTIONS    DEVICE
  primary-vds0
```

VCC				
NAME	PORT-RANGE			
primary-vcc0	5000-5100			
VSW				
NAME	MAC	NET-DEV	DEVICE	MODE
primary-vsw0	02:04:4f:fb:9f:0d	e1000g0	switch@0	prog,promisc

## Initial Configuration of the Control Domain

Initially, all system resources are allocated to the control domain. To allow the creation of other logical domains, you must release some of these resources.

---

**Note** – The notices that the LDoms Manager is running in configuration mode in the output in the following examples apply only to the Sun UltraSPARC T1 processors.

---

### ▼ To Set Up the Control Domain

---

**Note** – This procedure contains examples of resources to set for your control domain. These numbers are examples only, and the values used might not be appropriate for your control domain.

---

#### 1. Assign cryptographic resources to the control domain.

---

**Note** – If you have any cryptographic devices in the control domain, you cannot dynamically reconfigure CPUs. So if you are not using cryptographic devices, set-mau to 0.

---

The following example would assign one cryptographic resource to the control domain, `primary`. This leaves the remainder of the cryptographic resources available to a guest domain.

```
primary$ ldm set-mau 1 primary
```



## 2. Assign virtual CPUs to the control domain.

For example, the following command would assign 4 virtual CPUs to the control domain, `primary`. This leaves the remainder of the virtual CPUs available to a guest domain.

```
primary$ ldm set-vcpu 4 primary
```

## 3. Assign memory to the control domain.

For example, the following command would assign 1 gigabyte of memory to the control domain, `primary`. This leaves the remainder of the memory available to a guest domain.

```
primary$ ldm set-memory 1G primary
```

---

**Note** – If you are not using ZFS to deliver disk services, 1 gigabyte of memory should be adequate. If you are using ZFS to deliver disk services, assign a complete core of 4 virtual CPUs and at least 4 gigabyte of memory. You may need to assign additional complete cores for heavier I/O loads.

---

## 4. Add a logical domain machine configuration to the system controller (SC).

For example, the following command would add a configuration called `initial`.

```
primary$ ldm add-config initial
```

---

**Note** – Currently, there is a limit of 8 configurations that can be saved on the SC, not including the `factory-default` configuration.

---

## 5. Verify that the configuration is ready to be used at the next reboot.

```
primary$ ldm list-config
factory-default [current]
initial [next]
```

This list subcommand shows that the `factory-default` configuration set is currently being used and the `initial` configuration set will be used once you reboot.

---

# Rebooting to Use Logical Domains

You must reboot the control/service domain for the configuration changes to take effect and the resources to be released for other logical domains to use.

## ▼ To Reboot

1. Initiate a reconfiguration on the next boot.

```
primary# touch /reconfigure
```

---

**Note** – This reconfiguration step is necessary only if a virtual switch device was added as part of the domain reconfiguration. This is normally the case during the setting up of the control domain.

---

2. Shut down and reboot the `primary` domain, which is also the service domain in our examples.

```
primary# shutdown -y -g0 -i6
```

---

**Note** – While the reboot, using the command specified, allows the changes made to take effect, the `ldm list-config` command still shows the same output as before the reboot. Powering off and powering on are required for the `ldm list-config` command to update the displayed configuration.

---

---

# Enabling Networking Between the Control/Service Domain and Other Domains

By default, networking between the control/service domain and other domains in the system is disabled. To enable this, the virtual switch device should be configured as a network device. The virtual switch can either replace the underlying physical device (e1000g0 in this example) as the primary interface or be configured as an additional network interface in the domain.

---

**Note** – Perform the following configuration steps from the domain’s console, as the procedure could temporarily disrupt network connectivity to the domain.

---

## ▼ To Configure the Virtual Switch as the Primary Interface

1. Print out the addressing information for all interfaces.

```
primary# ifconfig -a
```

2. Plumb the virtual switch. In this example, vsw0 is the virtual switch being configured.

```
primary# ifconfig vsw0 plumb
```

3. (Optional) To obtain the list of all virtual switch instances in a domain, you can list them.

```
primary# /usr/sbin/dladm show-link | grep vsw
vsw0                type: non-vlan  mtu: 1500          device: vsw0
```

4. Unplumb the physical network device assigned to the virtual switch (net-dev), which is e1000g0 in this example.

```
primary# ifconfig e1000g0 down unplumb
```

5. To migrate properties of the physical network device (e1000g0) to the virtual switch (vsw0) device, do one of the following:

- If networking is configured using a static IP address, reuse the IP address and netmask of e1000g0 for vsw0.

```
primary# ifconfig vsw0 IP_of_e1000g0 netmask netmask_of_e1000g0 broadcast + up
```

- If networking is configured using DHCP, enable DHCP for vsw0.

```
primary# ifconfig vsw0 dhcp start
```

6. Make the required configuration file modifications to make this change permanent.

```
primary# mv /etc/hostname.e1000g0 /etc/hostname.vsw0
primary# mv /etc/dhcp.e1000g0 /etc/dhcp.vsw0
```

---

**Note** – If necessary, you can also configure the virtual switch as well as the physical network device. In this case, plumb the virtual switch as in Step 2, and do not unplumb the physical device (skip Step 4). The virtual switch must then be configured with either a static IP address or obtain a dynamic IP address from a DHCP server.

---

---

## Enabling the Virtual Network Terminal Server Daemon

You must enable the virtual network terminal server daemon (vntsd) to provide access to the virtual console of each logical domain. Refer to the Solaris 10 OS Reference Manual collection or the vntsd(1M) man page for information about how to use this daemon.

## ▼ To Enable the Virtual Network Terminal Server Daemon

---

**Note** – Be sure you have created the default service `vconscon` on the control domain before you enable `vntsd`. See “[Creating Default Services](#)” on page 40 for more information.

---

1. Use the `svcadm(1M)` command to enable the virtual network terminal server daemon, `vntsd(1M)`.

```
# svcadm enable vntsd
```

2. Use the `svcs(1)` command to verify that the `vntsd` is enabled.

```
# svcs -l vntsd
fmri          svc:/ldoms/vntsd:default
enabled       true
state         online
next_state    none
state_time    Sat Jan 27 03:14:17 2007
logfile       /var/svc/log/ldoms-vntsd:default.log
restarter     svc:/system/svc/restarter:default
contract_id   93
dependency    optional_all/error svc:/milestone/network (online)
dependency    optional_all/none svc:/system/system-log (online)
```

---

## Creating and Starting a Guest Domain

The guest domain must run an operating system that understands both the `sun4v` platform and the virtual devices presented by the hypervisor. Currently, this is the Solaris 10 11/06 OS at a minimum. Refer to the *Logical Domains (LDom)s 1.0.3 Release Notes* for any specific patches that might be necessary. Once you have created default services and reallocated resources from the control domain, you can create and start a guest domain.

## ▼ To Create and Start a Guest Domain

### 1. Create a logical domain.

For example, the following command would create a guest domain named `ldg1`.

```
primary$ ldm add-domain ldg1
```

### 2. Add CPUs to the guest domain.

For example, the following command would add four virtual CPUs to guest domain `ldg1`.

```
primary$ ldm add-vcpu 4 ldg1
```

### 3. Add memory to the guest domain.

For example, the following command would add 512 megabytes of memory to guest domain `ldg1`.

```
primary$ ldm add-memory 512m ldg1
```

### 4. Add a virtual network device to the guest domain.

For example, the following command would add a virtual network device with these specifics to the guest domain `ldg1`.

```
primary$ ldm add-vnet vnet1 primary-vsw0 ldg1
```

Where:

- `vnet1` is a unique interface name to the logical domain, assigned to this virtual network device instance for reference on subsequent `set-vnet` or `remove-vnet` subcommands.
  - `primary-vsw0` is the name of an existing network service (virtual switch) to which to connect.
- ### 5. Specify the device to be exported by the virtual disk server as a virtual disk to the guest domain.

You can export a physical disk, disk slice, volumes, or file as a block device. The following examples show a physical disk and a file.

- **Physical Disk Example.** The first example adds a physical disk with these specifics.

```
primary$ ldm add-vdsdev /dev/dsk/c0t0d0s2 vol1@primary-vds0
```

Where:

- `/dev/dsk/c0t0d0s2` is the path name of the actual physical device. When adding a device, the path name must be paired with the device name.
  - `vol1` is a unique name you must specify for the device being added to the virtual disk server. The volume name must be unique to this virtual disk server instance, because this name is exported by this virtual disk server to the clients for adding. When adding a device, the volume name must be paired with the path name of the actual device.
  - `primary-vds0` is the name of the virtual disk server to which to add this device.
- **File Example.** This second example is exporting a file as a block device.

```
primary$ ldm add-vdsdev backend vol1@primary-vds0
```

Where:

- `backend` is the path name of the actual file exported as a block device. When adding a device, the backend must be paired with the device name.
- `vol1` is a unique name you must specify for the device being added to the virtual disk server. The volume name must be unique to this virtual disk server instance, because this name is exported by this virtual disk server to the clients for adding. When adding a device, the volume name must be paired with the path name of the actual device.
- `primary-vds0` is the name of the virtual disk server to which to add this device.

## 6. Add a virtual disk to the guest domain.

The following example adds a virtual disk to the guest domain `ldg1`.

```
primary$ ldm add-vdisk vdisk1 vol1@primary-vds0 ldg1
```

Where:

- `vdisk1` is the name of the virtual disk.
- `vol1` is the name of the existing volume to which to connect.
- `primary-vds0` is the name of the existing virtual disk server to which to connect.

---

**Note** – The virtual disks are generic block devices that are backed by different types of physical devices, volumes, or files. A virtual disk is not synonymous with a SCSI disk and, therefore, excludes the target ID in the disk label. Virtual disks in a logical domain have the following format: `cNdnSN`, where `cN` is the virtual controller, `dN` is the virtual disk number, and `sN` is the slice.

---

**7. Set auto-boot and boot-device variables for the guest domain.**

The first example command sets auto-boot\? to true for guest domain ldg1.

```
primary$ ldm set-var auto-boot\?=true ldg1
```

The second example command sets boot-device to vdisk for the guest domain ldg1.

```
primary$ ldm set-var boot-device=vdisk ldg1
```

**8. Bind resources to the guest domain ldg1 and then list the domain to verify that it is bound.**

```
primary$ ldm bind-domain ldg1
primary$ ldm list-domain ldg1
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg1	bound	-----	5001	4	512M		

**9. To find the console port of the guest domain, you can look at the output of the preceding list-domain subcommand.**

You can see under the heading Cons that logical domain guest 1 (ldg1) has its console output bound to port 5001.

**10. Start the guest domain ldg1.**

```
primary$ ldm start-domain ldg1
```

**11. Connect to the console of a guest domain. There are several ways you can do this.**

- You can log into the control domain and connect directly to the console port on the local host:

```
$ ssh admin@controldom.domain
$ telnet localhost 5001
```

- You can also connect to a guest console over a network if it is enabled in the vntsd(1M) SMF manifest. For example:

```
$ telnet host-name 5001
```

A Service Management Facility manifest is an XML file that describes a service. For more information about creating an SMF manifest, refer to the Solaris 10 System Administrator Collection.



---

**Note** – To access a non-English OS in a guest domain through the console, the terminal for the console must be in the locale required by the OS.

---

## Jump-Starting a Guest Domain

If you are jump-starting a guest domain, you would use a normal JumpStart procedure with the following profile syntax changes from a regular Solaris OS JumpStart procedure to a JumpStart procedure specific to LDomS as shown in the following two examples.

### Normal JumpStart Profile

```
filesys c1t1d0s0 free /
filesys c1t1d0s1 2048 swap
filesys c1t1d0s5 120 /spare1
filesys c1t1d0s6 120 /spare2
```

Virtual disk device names in a logical domain differ from physical disk device names in that they do not contain a target ID (*tN*) in the device name. Instead of the normal *cNtNdNsN* format, virtual disk device names are of the format *cNdNsN*, where *cN* is the virtual controller, *dN* is the virtual disk number, and *sN* is the slice. Modify your JumpStart profile to reflect this change as in the following profile example.

### Actual Profile Used for a Logical Domain

```
filesys c0d0s0 free /
filesys c0d0s1 2048 swap
filesys c0d0s5 120 /spare1
filesys c0d0s6 120 /spare2
```



# Using Virtual Disks With Logical Domains

---

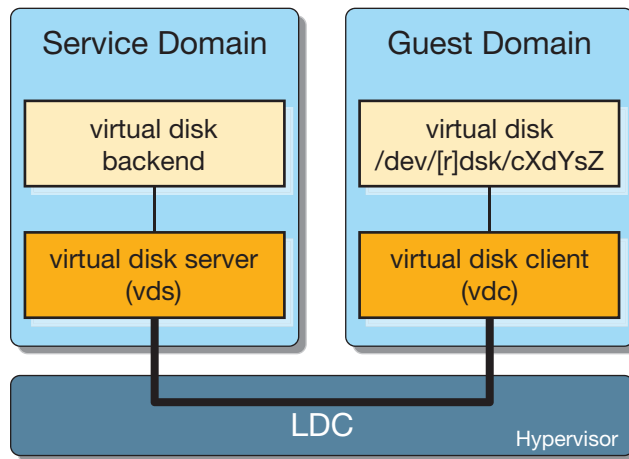
This chapter describes how to use virtual disks with Logical Domains software.

---

## Introduction to Virtual Disks

A virtual disk contains two components: the virtual disk itself as it appears in a domain guest, and the virtual disk backend, which is where data is stored and where virtual I/O ends up. The virtual disk backend is exported from a service domain by the virtual disk server (`vdS`) driver. The `vdS` driver communicates with the virtual disk client (`vdC`) driver in the guest domain through the hypervisor using a logical domain channel (LDC). Finally, a virtual disk appears as `/dev/[r]disk/cXdYsZ` devices in the guest domain.

**FIGURE 5-1** Virtual Disks With Logical Domains



The virtual disk backend can be a physical disk, a physical disk slice, a file, a volume from a volume management framework, such as the Zettabyte File System (ZFS), Solaris™ Volume Manager (SVM), Veritas Volume Manager (VxVM), or any disk pseudo device accessible from the service domain.

## Managing Virtual Disks

This section describes adding a virtual disk to a guest domain, changing virtual disk and timeout options, and removing a virtual disk from a guest domain. See [“Virtual Disk Backend Options” on page 57](#) for a description of virtual disk options. See [“Virtual Disk Timeout” on page 67](#) for a description of the virtual disk timeout.

### ▼ To Add a Virtual Disk

1. Export the virtual disk backend from a service domain by using the following command.

```
# ldm add-vdsdev [options={ro,slice,excl}] backend
volume_name@service_name
```

2. Assign the backend to a guest domain by using the following command.

```
# ldm add-vdisk [timeout=seconds] disk_name volume_name@service_name ldom
```

---

**Note** – A backend is actually exported from the service domain and assigned to the guest domain when the guest domain (*ldom*) is bound.

---

## ▼ To Export a Virtual Disk Backend Multiple Times

A virtual disk backend can be exported multiple times either through the same or different virtual disk servers. Each exported instance of the virtual disk backend can then be assigned to either the same or different guest domains.

When a virtual disk backend is exported multiple times, it should not be exported with the exclusive (*excl*) option. Specifying the *excl* option will only allow exporting the backend once. The backend can be safely exported multiple times as a read-only device with the *ro* option.



---

**Caution** – When a virtual disk backend is exported multiple times, applications running on guest domains and using that virtual disk are responsible for coordinating and synchronizing concurrent write access to ensure data coherency.

---

The following example describes how to add the same virtual disk to two different guest domains through the same virtual disk service.

1. Export the virtual disk backend two times from a service domain by using the following commands.

```
# ldm add-vdsdev [options={ro,slice}] backend volume1@service_name
# ldm add-vdsdev [options={ro,slice}] backend volume2@service_name
```

The *add-vdsdev* subcommand displays the following warning to indicate that the backend is being exported more than once.

```
Warning: "backend" is already in use by one or more servers in guest
"ldom"
```

2. Assign the exported backend to each guest domain by using the following commands.

The *disk\_name* can be different for *ldom1* and *ldom2*.

```
# ldm add-vdisk [timeout=seconds] disk_name volume1@service_name ldom1
# ldm add-vdisk [timeout=seconds] disk_name volume2@service_name ldom2
```

## ▼ To Change Virtual Disk Options

- After a backend is exported from the service domain, you can change the virtual disk options by using the following command.

```
# ldm set-vdsdev options=[{ro,slice,excl}] volume_name@service_name
```

## ▼ To Change the Timeout Option

- After a virtual disk is assigned to a guest domain, you can change the timeout of the virtual disk by using the following command.

```
# ldm set-vdisk timeout=seconds disk_name ldom
```

## ▼ To Remove a Virtual Disk

1. Remove a virtual disk from a guest domain by using the following command.

```
# ldm rm-vdisk disk_name ldom
```

2. Stop exporting the corresponding backend from the service domain by using the following command.

```
# ldm rm-vdsdev volume_name@service_name
```

---

# Virtual Disk Appearance

When a backend is exported as a virtual disk, it can appear in the guest domain either as a full disk or as a single slice disk. The way it appears depends on the type of the backend and on the options used to export it.

## Full Disk

When a backend is exported to a domain as a full disk, it appears in that domain as a regular disk with 8 slices (*s0* to *s7*). Such a disk is visible with the `format(1M)` command. The disk's partition table can be changed using either the `fmthard(1M)` or `format(1M)` command.

A full disk is also visible to the OS installation software and can be selected as a disk onto which the OS can be installed.

Any backend can be exported as a full disk except physical disk slices that can only be exported as single slice disks.

## Single Slice Disk

When a backend is exported to a domain as a single slice disk, it appears in that domain as a disk with a single partition (`s0`). Such a disk is not visible with the `format(1M)` command, and its partition table cannot be changed.

A single slice disk is not visible from the OS installation software and cannot be selected as a disk device onto which the OS can be installed.

Any backend can be exported as a single slice disk except physical disks that can only be exported as full disks.

---

## Virtual Disk Backend Options

Different options can be specified when exporting a virtual disk backend. These options are indicated in the `options=` argument of the `ldm add-vdsdev` command as a comma separated list. The valid options are: `ro`, `slice`, and `excl`.

### Read-only (`ro`) Option

The read-only (`ro`) option specifies that the backend is to be exported as a read-only device. In that case, the virtual disk assigned to the guest domain can only be accessed for read operations, and any write operation to the virtual disk will fail.

### Exclusive (`excl`) Option

The exclusive (`excl`) option specifies that the backend in the service domain has to be opened exclusively by the virtual disk server when it is exported as a virtual disk to another domain. When a backend is opened exclusively, it is not accessible by other applications in the service domain. This prevents the applications running in the service domain from inadvertently using a backend that is also being used by a guest domain.

---

**Note** – Some drivers do not honor the `excl` option and will disallow some virtual disk backends from being opened exclusively. The `excl` option is known to work with physical disks and slices, but the option does not work with files. It may or may not work with pseudo devices, such as disk volumes. If the driver of the backend does not honor the exclusive open, the backend `excl` option is ignored, and the backend is not opened exclusively.

---

Because the `excl` option prevents applications running in the service domain from accessing a backend exported to a guest domain, do not set the `excl` option in the following situations:

- When guest domains are running, if you want to be able to use commands such as `format(1M)` or `luxadm(1M)` to manage physical disks, then do not export these disks with the `excl` option.
- When you export an SVM volume, such as a RAID or a mirrored volume, do not set the `excl` option. Otherwise, this can prevent SVM from starting some recovery operation in case a component of the RAID or mirrored volume fails. See [“Using Virtual Disks on Top of SVM” on page 75](#) for more information.
- If the Veritas Volume Manager (VxVM) is installed in the service domain and Veritas Dynamic Multipathing (VxDMP) is enabled for physical disks, then physical disks have to be exported without the `excl` option. Otherwise, the export fails, because the virtual disk server (`vdcs`) is unable to open the physical disk device. See [“Using Virtual Disks When VxVM Is Installed” on page 76](#) for more information.
- If you are exporting the same virtual disk backend multiple times from the same virtual disk service, see [“To Export a Virtual Disk Backend Multiple Times” on page 55](#) for more information.

By default, the backend is opened non-exclusively. That way the backend still can be used by applications running in the service domain while it is exported to another domain. Note that this is a new behavior starting with the Solaris 10 5/08 OS release. Before the Solaris 10 5/08 OS release, disk backends were always opened exclusively, and it was not possible to have a backend opened non-exclusively.

## Slice (`slice`) Option

A backend is normally exported either as a full disk or as a single slice disk depending on its type. If the `slice` option is specified, then the backend is forcibly exported as a single slice disk.

This option is useful when you want to export the raw content of a backend. For example, if you have a ZFS or SVM volume where you have already stored data and you want your guest domain to access this data, then you should export the ZFS or SVM volume using the `slice` option.



For more information about this option, see "Virtual Disk Backend".

---

## Virtual Disk Backend

The virtual disk backend is the location where data of a virtual disk are stored. The backend can be a disk, a disk slice, a file, or a volume, such as ZFS, SVM, or VxVM. A backend appears in a guest domain either as a full disk or as single slice disk, depending on whether the `slice` option is set when the backend is exported from the service domain. By default, a virtual disk backend is exported non-exclusively as a readable-writable full disk.

## Physical Disk or Disk LUN

A physical disk or logical unit number (LUN) is always exported as a full disk. In that case, virtual disk drivers (`vds` and `vdcc`) forward I/O from the virtual disk and act as a pass-through to the physical disk or LUN.

A physical disk or LUN is exported from a service domain by exporting the device corresponding to the slice 2 (`s2`) of that disk without setting the `slice` option. If you export the slice 2 of a disk with the `slice` option, only this slice is exported and not the entire disk.

### ▼ To Export a Physical Disk as a Virtual Disk

1. For example, to export the physical disk `c1t48d0` as a virtual disk, you must export slice 2 of that disk (`c1t48d0s2`) from the service domain as follows.

```
service# ldm add-vdsdev /dev/dsk/c1t48d0s2 c1t48d0@primary-vds0
```

2. From the service domain, assign the disk (`pdisk`) to guest domain `ldg1`, for example.

```
service# ldm add-vdisk pdisk c1t48d0@primary-vds0 ldg1
```

3. After the guest domain is started and running the Solaris OS, you can list the disk (c0d1, for example) and see that the disk is accessible and is a full disk; that is, a regular disk with 8 slices.

```
ldg1# ls -l /dev/dsk/c0d1s*
/dev/dsk/c0d1s0
/dev/dsk/c0d1s1
/dev/dsk/c0d1s2
/dev/dsk/c0d1s3
/dev/dsk/c0d1s4
/dev/dsk/c0d1s5
/dev/dsk/c0d1s6
/dev/dsk/c0d1s7
```

## Physical Disk Slice

A physical disk slice is always exported as a single slice disk. In that case, virtual disk drivers (vds and vdc) forward I/O from the virtual disk and act as a pass-through to the physical disk slice.

A physical disk slice is exported from a service domain by exporting the corresponding slice device. If the device is different from slice 2 then it is automatically exported as a single slice disk whether or not you specify the `slice` option. If the device is the slice 2 of the disk, you must set the `slice` option to export only slice 2 as a single slice disk; otherwise, the entire disk is exported as full disk.

## ▼ To Export a Physical Disk Slice as a Virtual Disk

1. For example, to export slice 0 of the physical disk c1t57d0 as a virtual disk, you must export the device corresponding to that slice (c1t57d0s0) from the service domain as follows.

```
service# ldmd add-vdsdev /dev/dsk/c1t57d0s0 c1t57d0s0@primary-vds0
```

You do not need to specify the `slice` option, because a slice is always exported as a single slice disk.

2. From the service domain, assign the disk (pslice) to guest domain ldg1, for example.

```
service# ldmd add-vdisk pslice c1t57d0s0@primary-vds0 ldg1
```

3. After the guest domain is started and running the Solaris OS, you can list the disk (c0d1, for example) and see that the disk is accessible and is a single slice disk (s0).

```
ldg1# ls -l /dev/dsk/c0d13s*  
/dev/dsk/c0d13s0
```

## ▼ To Export Slice 2

- To export slice 2 (disk c1t57d0s2, for example) you must specify the `slice` option; otherwise, the entire disk is exported.

```
# ldm add-vdsdev options=slice /dev/dsk/c1t57d0s2 c1t57d0s2@primary-vds0
```

## File and Volume

A file or volume (for example from ZFS or SVM) is exported either as a full disk or as single slice disk depending on whether or not the `slice` option is set.

### File or Volume Exported as a Full Disk

If you do not set the `slice` option, a file or volume is exported as a full disk. In that case, virtual disk drivers (`vds` and `vdc`) forward I/O from the virtual disk and manage the partitioning of the virtual disk. The file or volume eventually becomes a disk image containing data from all slices of the virtual disk and the metadata used to manage the partitioning and disk structure.

When a blank file or volume is exported as full disk, it appears in the guest domain as an unformatted disk; that is, a disk with no partition. Then you need to run the `format(1M)` command in the guest domain to define usable partitions and to write a valid disk label. Any I/O to the virtual disk fails while the disk is unformatted.

---

**Note** – Before the Solaris 10 5/08 OS release, when a blank file was exported as a virtual disk, the system wrote a default disk label and created default partitioning. This is no longer the case with the Solaris 10 5/08 OS release, and you must run `format(1M)` in the guest domain to create partitions.

---

## ▼ To Export a File as a Full Disk

1. From the service domain, create a file (`fdisk0` for example) to use as the virtual disk.

```
service# mkfile 100m /ldoms/domain/test/fdisk0
```

The size of the file defines the size of the virtual disk. This example creates a 100-megabyte blank file to get a 100-megabyte virtual disk.

2. From the service domain, export the file as a virtual disk.

```
service# ldm add-vdsdev /ldoms/domain/test/fdisk0  
fdisk0@primary-vds0
```

In this example, the `slice` option is not set, so the file is exported as a full disk.

3. From the service domain, assign the disk (`fdisk`) to guest domain `ldg1`, for example.

```
service# ldm add-vdisk fdisk fdisk0@primary-vds0 ldg1
```

4. After the guest domain is started and running the Solaris OS, you can list the disk (`c0d5`, for example) and see that the disk is accessible and is a full disk; that is, a regular disk with 8 slices.

```
ldg1# ls -l /dev/dsk/c0d5*  
/dev/dsk/c0d5s0  
/dev/dsk/c0d5s1  
/dev/dsk/c0d5s2  
/dev/dsk/c0d5s3  
/dev/dsk/c0d5s4  
/dev/dsk/c0d5s5  
/dev/dsk/c0d5s6  
/dev/dsk/c0d5s7
```

## File or Volume Exported as a Single Slice Disk

If the `slice` option is set, then the file or volume is exported as a single slice disk. In that case, the virtual disk has only one partition (`s0`), which is directly mapped to the file or volume backend. The file or volume only contains data written to the virtual disk with no extra data like partitioning information or disk structure.

When a file or volume is exported as a single slice disk, the system simulates a fake disk partitioning which makes that file or volume appear as a disk slice. Because the disk partitioning is simulated, you do not create a partitioning for that disk.

## ▼ To Export a ZFS Volume as a Single Slice Disk

1. From the service domain, create a ZFS volume (zdisk0 for example) to use as a single slice disk.

```
service# zfs create -V 100m ldoms/domain/test/zdisk0
```

The size of the volume defines the size of the virtual disk. This example creates a 100-megabyte volume to get a 100-megabyte virtual disk.

2. From the service domain, export the device corresponding to that ZFS volume, and set the `slice` option so that the volume is exported as a single slice disk.

```
service# ldm add-vdsdev options=slice  
/dev/zvol/dsk/ldoms/domain/test/zdisk0 zdisk0@primary-vds0
```

3. From the service domain, assign the volume (zdisk0) to guest domain ldg1, for example.

```
service# ldm add-vdisk zdisk0 zdisk0@primary-vds0 ldg1
```

4. After the guest domain is started and running the Solaris OS, you can list the disk (c0d9, for example) and see that the disk is accessible and is a single slice disk (s0).

```
ldg1# ls -l /dev/dsk/c0d9s*  
/dev/dsk/c0d9s0
```

## Exporting Volumes and Backward Compatibility

Before the Solaris 10 5/08 OS release, the `slice` option did not exist, and volumes were exported as single slice disks. If you have a configuration exporting volumes as virtual disks and if you upgrade the system to the Solaris 10 5/08 OS, volumes are now exported as full disks instead of single slice disks. To preserve the old behavior and to have your volumes exported as single slice disks, you need to do either of the following:

- Use the `ldm set-vdsdev` command in LDomS 1.0.3 software, and set the `slice` option for all volumes you want to export as single slice disks. Refer to the `ldm` man page or the *Logical Domains (LDoms) Manager 1.0.3 Man Page Guide* for more information about this command.
- Add the following line to the `/etc/system` file on the service domain.

```
set vds:vd_volume_force_slice = 1
```

---

**Note** – Setting this tunable forces the export of all volumes as single slice disks, and you cannot export any volume as a full disk.

---

## Summary of How Different Types of Backends Are Exported

Backend	No Slice Option	Slice Option Set
Disk (disk slice 2)	Full disk <sup>*</sup>	Single slice disk <sup>d</sup>
Disk slice (not slice 2)	Single slice disk <sup>\</sup>	Single slice disk
File	Full disk	Single slice disk
Volume, including ZFS, SVM, or VxVM	Full disk	Single slice disk

<sup>\*</sup> Export the entire disk.

<sup>\</sup> A slice is always exported as a single slice disk.

<sup>d</sup> Export only slice 2

## Guidelines

### *Using the Loopback File (lofi) Driver*

It is possible to use the loopback file (lofi) driver to export a file as a virtual disk. However, doing this adds an extra driver layer and impacts performance of the virtual disk. Instead, you can directly export a file as a full disk or as a single slice disk. See [“File and Volume” on page 61](#).

### *Directly or Indirectly Exporting a Disk Slice*

To export a slice as a virtual disk either directly or indirectly (for example through a SVM volume), ensure that the slice does not start on the first block (block 0) of the physical disk by using the `prtvto(1M)` command.

If you directly or indirectly export a disk slice which starts on the first block of a physical disk, you might overwrite the partition table of the physical disk and make all partitions of that disk inaccessible.

---

## CD, DVD and ISO Images

You can export a compact disc (CD) or digital versatile disc (DVD) the same way you export any regular disk. To export a CD or DVD to a guest domain, export slice 2 of the CD or DVD device as a full disk; that is, without the `slice` option.

---

**Note** – You cannot export the CD or DVD drive itself; you only can export the CD or DVD that is inside the CD or DVD drive. Therefore, a CD or DVD must be present inside the drive before you can export it. Also, to be able to export a CD or DVD, that CD or DVD cannot be in use in the service domain. In particular, the Volume Management file system, `volfs(7FS)` service must not use the CD or DVD. See [“To Export a CD or DVD From the Service Domain to the Guest Domain” on page 66](#) for instructions on how to remove the device from use by `volfs`.

---

If you have an International Organization for Standardization (ISO) image of a CD or DVD stored in file or on a volume, and export that file or volume as a full disk then it appears as a CD or DVD in the guest domain.

When you export a CD, DVD, or an ISO image, it automatically appears as a read-only device in the guest domain. However, you cannot perform any CD control operations from the guest domain; that is, you cannot start, stop, or eject the CD from the guest domain. If the exported CD, DVD, or ISO image is bootable, the guest domain can be booted on the corresponding virtual disk.

For example, if you export a Solaris OS installation DVD, you can boot the guest domain on the virtual disk corresponding to that DVD and install the guest domain from that DVD. To do so, when the guest domain reaches the `ok` prompt, use the following command.

```
ok boot /virtual-devices@100/channel-devices@200/disk@n:f
```

Where *n* is the index of the virtual disk representing the exported DVD.

---

**Note** – If you export a Solaris OS installation DVD and boot a guest domain on the virtual disk corresponding to that DVD to install the guest domain, then you cannot change the DVD during the installation. So you might need to skip any step of the installation requesting a different CD/DVD, or you will need to provide an alternate path to access this requested media.

---

## ▼ To Export a CD or DVD From the Service Domain to the Guest Domain

1. Insert the CD or DVD in the CD or DVD drive.
2. From the service domain, check whether the volume management daemon, `vold(1M)`, is running and online.

```
service# svcs volfs
STATE          STIME          FMRI
online         12:28:12    svc:/system/filesystem/volfs:default
```

3. Do one of the following.

- If the volume management daemon is not running or online, go to step 5.
- If the volume management daemon is running and online, as in the example in Step 2, do the following:
  - a. Edit the `/etc/vold.conf` file and comment out the line starting with the following words.

```
use cdrom drive....
```

Refer to the `vold.conf(1M)` man page for more information.

- b. From the service domain, restart the volume management file system service.

```
service# svcadm refresh volfs
service# svcadm restart volfs
```

4. From the service domain, find the disk path for the CD-ROM device.

```
service# cdwr -l
Looking for CD devices...
Node                               Connected Device                               Device type
-----+-----+-----
/dev/rdisk/c1t0d0s2 | MATSHITA CD-RW  CW-8124  DZ13 | CD Reader/Writer
```

5. From the service domain, export the CD or DVD disk device as a full disk.

```
service# ldm add-vdsdev /dev/dsk/c1t0d0s2 cdrom@primary-vds0
```

6. From the service domain, assign the exported CD or DVD to the guest domain (`ldg1` in this example).

```
service# ldm add-vdisk cdrom cdrom@primary-vds0 ldg1
```



## Exporting a CD or DVD Multiple Times

A CD or DVD can be exported multiple times and assigned to different guest domain. See [“To Export a Virtual Disk Backend Multiple Times” on page 55](#) for more information.

---

# Virtual Disk Timeout

By default, if the service domain providing access to a virtual disk backend is down, all I/O from the guest domain to the corresponding virtual disk is blocked. The I/O automatically is resumed when the service domain is operational and is servicing I/O requests to the virtual disk backend.

However, there are some cases when file systems or applications might not want the I/O operation to block, but for it to fail and report an error if the service domain is down for too long. It is now possible to set a connection timeout period for each virtual disk, which can then be used to establish a connection between the virtual disk client on a guest domain and the virtual disk server on the service domain. When that timeout period is reached, any pending I/O and any new I/O will fail as long as the service domain is down and the connection between the virtual disk client and server is not reestablished.

This timeout can be set by doing one of the following:

- Using the `ldm add-vdisk` command.

```
ldm add-vdisk timeout=seconds disk_name volume_name@service_name ldom
```

- Using the `ldm set-vdisk` command.

```
ldm set-vdisk timeout=seconds disk_name ldom
```

Specify the timeout in seconds. If the timeout is set to 0, the timeout is disabled and I/O is blocked while the service domain is down (this is the default setting and behavior).

Alternatively, the timeout can be set by adding the following line to the `/etc/system` file on the guest domain.

```
set vdc:vdc_timeout = seconds
```

---

**Note** – If this tunable is set, it overwrites any timeout setting done using the `ldm` CLI. Also, the tunable sets the timeout for all virtual disks in the guest domain.

---

---

## Virtual Disk and SCSI

If a physical SCSI disk or LUN is exported as a full disk, the corresponding virtual disk supports the user SCSI command interface, `uscsi(7D)` and multihost disk control operations `mhd(7I)`. Other virtual disks, such as virtual disks having a file or a volume as a backend, do not support these interfaces.

As a consequence, applications or product features using SCSI commands (such as SVM metaset, or Solaris Cluster shared devices) can be used in guest domains only with virtual disks having a physical SCSI disk as a backend.

---

**Note** – SCSI operations are effectively executed by the service domain, which manages the physical SCSI disk or LUN used as a virtual disk backend. In particular, SCSI reservations are done by the service domain. Therefore, applications running in the service domain and in guest domains should not issue SCSI commands to the same physical SCSI disks; otherwise, this can lead to an unexpected disk state.

---

---

## Virtual Disk and the `format(1M)` Command

The `format(1M)` command works in a guest domain with virtual disks exported as full disk. Single slice disks are not seen by the `format(1M)` command, and it is not possible to change the partitioning of such disks.

Virtual disks whose backends are SCSI disks support all `format(1M)` subcommands. Virtual disks whose backends are not SCSI disks do not support some `format(1M)` subcommands, such as `repair` and `defect`. In that case, the behavior of `format(1M)` is similar to the behavior of integrated development environment (IDE) disks.

---

**Note** – The `format(1M)` command crashes when you select a virtual disk that has an extensible firmware interface (EFI) disk label. Refer to Bug ID 6363316 in the *Logical Domains (LDoms) 1.0.3 Release Notes*.

---

---

# Using ZFS With Virtual Disks

The following topics regarding using the Zettabyte File System (ZFS) with virtual disks on logical domains are described in this section:

- “Creating a Virtual Disk on Top of a ZFS Volume” on page 69
- “Using ZFS Over a Virtual Disk” on page 70
- “Using ZFS for Boot Disks” on page 72

## Creating a Virtual Disk on Top of a ZFS Volume

The following procedure describes how to create a ZFS volume in a service domain and make that volume available to other domains as a virtual disk. In this example, the service domain is the same as the control domain and is named `primary`. The guest domain is named `ldg1` as an example. The prompts in each step show in which domain to run the command.

### ▼ To Create a Virtual Disk on Top of a ZFS Volume

1. Create a ZFS storage pool (`zpool`).

```
primary# zpool create -f tank1 c2t42d1
```

2. Create a ZFS volume.

```
primary# zfs create -V 100m tank1/myvol
```

3. Verify that the `zpool` (`tank1` in this example) and ZFS volume (`tank/myvol` in this example) have been created.

```
primary# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
tank1	100M	43.0G	24.5K	/tank1
tank1/myvol	22.5K	43.1G	22.5K	-

4. Configure a service exporting `tank1/myvol` as a virtual disk.

```
primary# ldm add-vdsdev options=slice /dev/zvol/dsk/tank1/myvol  
zvol@primary-vds0
```

5. Add the exported disk to another domain (ldg1 in this example).

```
primary# ldm add-vdisk vzdisk zvol@primary-vds0 ldg1
```

6. On the other domain (ldg1 in this example), start the domain and ensure that the new virtual disk is visible (you might have to run the `devfsadm` command).

In this example, the new disk appears as `/dev/rdisk/c2d2s0`.

```
ldg1# newfs /dev/rdsk/c2d2s0
newfs: construct a new file system /dev/rdsk/c2d2s0: (y/n)? y
Warning: 4096 sector(s) in last cylinder unallocated
Warning: 4096 sector(s) in last cylinder unallocated
/dev/rdsk/c2d2s0: 204800 sectors in 34 cylinders of 48 tracks, 128 sectors
100.0MB in 3 cyl groups (14 c/g, 42.00MB/g, 20160 i/g) super-block backups
(for fsck -F ufs -o b=#) at: 32, 86176, 172320,

ldg1# mount /dev/dsk/c2d2s0 /mnt

ldg1# df -h /mnt
Filesystem                size      used    avail capacity  Mounted on
/dev/dsk/c2d2s0           93M       1.0M      82M        2%    /mnt
```

---

**Note** – In this example, the ZFS volume is exported as single slice disk. You also can export a ZFS volume as full disk. Export a ZFS volume as full disk to partition the virtual disk or to install the Solaris OS on the virtual disk.

---

## Using ZFS Over a Virtual Disk

The following procedure shows how to use ZFS directly from a domain on top of a virtual disk. You can create ZFS pools, file systems, and volumes over the top of virtual disks with the Solaris 10 OS `zpool(1M)` and `zfs(1M)` commands. Although the storage backend is different (virtual disks instead of physical disks), there is no change to the usage of ZFS.

Additionally, if you have an already existing ZFS file system, then you can export it from a service domain to use it in another domain.

In this example, the service domain is the same as the control domain and is named `primary`. The guest domain is named `ldg1` as an example. The prompts in each step show in which domain to run the command.

## ▼ To Use ZFS Over a Virtual Disk

1. Create a ZFS pool (tank in this example), and then verify that it has been created.

```
primary# zpool create -f tank c2t42d0
primary# zpool list
```

NAME	SIZE	USED	AVAIL	CAP	HEALTH	ALTROOT
tank	43.8G	108K	43.7G	0%	ONLINE	-

2. Create a ZFS file system (tank/test in this example), and then verify that it has been created.

In this example, the file system is created on top of disk c2t42d0 by running the following command on the service domain.

```
primary# zfs create tank/test
primary# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
tank	106K	43.1G	25.5K	/tank
tank/test	24.5K	43.1G	24.5K	/tank/test

3. Export the ZFS pool (tank in this example).

```
primary# zpool export tank
```

4. Configure a service exporting the physical disk c2t42d0s2 as a virtual disk.

```
primary# ldm add-vdsdev /dev/rdisk/c2t42d0s2 volz@primary-vds0
```

5. Add the exported disk to another domain (ldg1 in this example).

```
primary# ldm add-vdisk vdiskz volz@primary-vds0 ldg1
```

6. On the other domain (ldg1 in this example), start the domain and make sure the new virtual disk is visible (you might have to run the `devfsadm` command), and then import the ZFS pool.

```
ldg1# zpool import tank
ldg1# zpool list
```

NAME	SIZE	USED	AVAIL	CAP	HEALTH	ALTROOT
tank	43.8G	214K	43.7G	0%	ONLINE	-

```
ldg1# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
tank	106K	43.1G	25.5K	/tank
tank/test	24.5K	43.1G	24.5K	/tank/test

```
ldg1# df -hl -F zfs
```

Filesystem	size	used	avail	capacity	Mounted on
tank	43G	25K	43G	1%	/tank
tank/test	43G	24K	43G	1%	/tank/test

The ZFS pool (tank/test in this example) is now imported and usable from domain ldg1.

## Using ZFS for Boot Disks

You can use a ZFS file system with a large file as the virtual disks in logical domains.

---

**Note** – A ZFS file system requires more memory in the service domain. Take this into account when configuring the service domain.

---

ZFS enables:

- Cloning a file system quickly
- Using the clones to provision additional domains
- Net installing to disk on files and files within a ZFS file system

### ▼ To Use ZFS for Boot Disks

You can use the following procedure to create ZFS disks for logical domains, and also snapshot and clone them for other domains.

1. On the **primary domain**, reserve a entire disk or slice for use as the storage for the ZFS pool. Step 2 uses slice 5 of a disk.

2. Create a ZFS pool; for example, `ldomspool`.

```
# zpool create ldomspool /dev/dsk/c0t0d0s5
```

3. Create a ZFS file system for the first domain (`ldg1` in this example).

```
# zfs create ldomspool/ldg1
```

4. Create a file to be the disk for this domain.

```
# mkfile 1G /ldomspool/ldg1/bootdisk
```

5. Specify the file as the device to use when creating the domain.

```
primary# ldm add-vdsdev /ldomspool/ldg1/bootdisk vol1@primary-vds0
primary# ldm add-vdisk vdisk1 vol1@primary-vds0 ldg1
```

6. Boot domain `ldg1` and net install to `vdisk1`. This file functions as a full disk and can have partitions; that is, separate partitions for `root`, `usr`, `home`, `dump`, and `swap`.

7. Once the installation is complete, snapshot the file system.

```
# zfs snapshot ldomspool/ldg1@initial
```

---

**Note** – Doing the snapshot before the domain reboots does not save the domain state as part of the snapshot or any other clones created from the snapshot.

---

8. Create additional clones from the snapshot and use it as the boot disk for other domains (`ldg2` and `ldg3` in this example).

```
# zfs clone ldomspool/ldg1@initial ldomspool/ldg2
# zfs clone ldomspool/ldg1@initial ldomspool/ldg3
```

9. Verify that everything was created successfully.

```
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPPOINT
ldomspool	1.07G	2.84G	28.5K	/ldomspool
ldomspool/ldg1	1.03G	2.84G	1.00G	/ldomspool/ldg1
ldomspool/ldg1@initial	23.0M	-	1.00G	-
ldomspool/ldg2	23.2M	2.84G	1.00G	/ldomspool/ldg2
ldomspool/ldg3	21.0M	2.84G	1.00G	/ldomspool/ldg3

---

**Note** – Ensure that the ZFS pool has enough space for the clones that are being created. ZFS uses copy-on-write and uses space from the pool only when the blocks in the clone are modified. Even after booting the domain, the clones only use a small percentage needed for the disk (since most of the OS binaries are the same as those in the initial snapshot).

---

---

## Using Volume Managers in a Logical Domains Environment

The following topics are described in this section:

- [“Using Virtual Disks on Top of Volume Managers” on page 74](#)
- [“Using Volume Managers on Top of Virtual Disks” on page 77](#)

### Using Virtual Disks on Top of Volume Managers

Any Zettabyte File System (ZFS), Solaris Volume Manager (SVM), or Veritas Volume Manager (VxVM) volume can be exported from a service domain to a guest domain as a virtual disk. A volume can be exported either as a single slice disk (if the `slice` option is specified with the `ldm add-vdsdev` command) or as a full disk.

---

**Note** – The remainder of this section uses an SVM volume as an example. However, the discussion also applies to ZFS and VxVM volumes.

---

The following example shows how to export a volume as a single slice disk. For example, if a service domain exports the SVM volume `/dev/md/dsk/d0` to `domain1` as a single slice disk, and `domain1` sees that virtual disk as `/dev/dsk/c0d2*`, then `domain1` only has an `s0` device; that is, `/dev/dsk/c0d2s0`.

The virtual disk in the guest domain (for example, `/dev/dsk/c0d2s0`) is directly mapped to the associated volume (for example, `/dev/md/dsk/d0`), and data stored onto the virtual disk from the guest domain are directly stored onto the associated volume with no extra metadata. So data stored on the virtual disk from the guest domain can also be directly accessed from the service domain through the associated volume.



Examples:

- If the SVM volume `d0` is exported from the primary domain to `domain1`, then the configuration of `domain1` requires some extra steps.

```
primary# metainit d0 3 1 c2t70d0s6 1 c2t80d0s6 1 c2t90d0s6
primary# ldm add-vdsdev options=slice /dev/md/dsk/d0
vol3@primary-vds0
primary# ldm add-vdisk vdisk3 vol3@primary-vds0 domain1
```

- After `domain1` has been bound and started, the exported volume appears as `/dev/dsk/c0d2s0`, for example, and you can use it.

```
domain1# newfs /dev/rdisk/c0d2s0
domain1# mount /dev/dsk/c0d2s0 /mnt
domain1# echo test-domain1 > /mnt/file
```

- After `domain1` has been stopped and unbound, data stored on the virtual disk from `domain1` can be directly accessed from the primary domain through SVM volume `d0`.

```
primary# mount /dev/md/dsk/d0 /mnt
primary# cat /mnt/file
test-domain1
```

---

**Note** – A single slice disk cannot be seen by the `format(1M)` command, cannot be partitioned, and cannot be used as an installation disk for the Solaris OS. See [“Virtual Disk Appearance” on page 56](#) for more information about this topic.

---

## Using Virtual Disks on Top of SVM

When a RAID or mirror SVM volume is used as a virtual disk by another domain, then it has to be exported without setting the exclusive (`excl`) option. Otherwise, if there is a failure on one of the components of the SVM volume, then the recovery of the SVM volume using the `metareplace` command or using a hot spare does not start. The `metastat` command sees the volume as resynchronizing, but the resynchronization does not progress.

For example, `/dev/md/dsk/d0` is a RAID SVM volume exported as a virtual disk with the `excl` option to another domain, and `d0` is configured with some hot-spare devices. If a component of `d0` fails, SVM replaces the failing component with a hot

spare and resynchronizes the SVM volume. However, the resynchronization does not start. The volume is reported as resynchronizing, but the resynchronization does not progress.

```
# metastat d0
d0: RAID
    State: Resyncing
    Hot spare pool: hsp000
    Interlace: 32 blocks
    Size: 20097600 blocks (9.6 GB)
Original device:
    Size: 20100992 blocks (9.6 GB)
Device                               Start Block  Dbase   State Reloc
c2t2d0s1                             330         No      Okay   Yes
c4t12d0s1                             330         No      Okay   Yes
/dev/dsk/c10t600C0FF0000000000015153295A4B100d0s1 330         No      Resyncing   Yes
```

In such a situation, the domain using the SVM volume as a virtual disk has to be stopped and unbound to complete the resynchronization. Then the SVM volume can be resynchronized using the `metasync` command.

```
# metasync d0
```

## Using Virtual Disks When VxVM Is Installed

When the Veritas Volume Manager (VxVM) is installed on your system, and if Veritas Dynamic Multipathing (DMP) is enabled on a physical disk or partition you want to export as virtual disk, then you have to export that disk or partition without setting the `excl` option. Otherwise, you receive an error in `/var/adm/messages` while binding a domain that uses such a disk.

```
vd_setup_vd(): ldi_open_by_name(/dev/dsk/c4t12d0s2) = errno 16
vds_add_vd(): Failed to add vdisk ID 0
```

You can check if Veritas DMP is enabled by checking multipathing information in the output of the command `vxdisk list`; for example:

```
# vxdisk list Disk_3
Device:      Disk_3
devicetag:   Disk_3
type:        auto
info:        format=none
flags:       online ready private autoconfig invalid
pubpaths:    block=/dev/vx/dmp/Disk_3s2 char=/dev/vx/rdmp/Disk_3s2
guid:        -
udid:        SEAGATE%5FST336753LSUN36G%5FDISKS%5F3032333948303144304E0000
```

```
site: -
Multipathing information:
numpaths: 1
c4t12d0s2 state=enabled
```

Alternatively, if Veritas DMP is enabled on a disk or a slice that you want to export as a virtual disk with the `excl` option set, then you can disable DMP using the `vxdmpadm` command. For example:

```
# vxdmpadm -f disable path=/dev/dsk/c4t12d0s2
```

## Using Volume Managers on Top of Virtual Disks

This section describes the following situations in the Logical Domains environment:

- [“Using ZFS on Top of Virtual Disks” on page 77](#)
- [“Using SVM on Top of Virtual Disks” on page 77](#)
- [“Using VxVM on Top of Virtual Disks” on page 78](#)

## Using ZFS on Top of Virtual Disks

Any virtual disk can be used with ZFS. A ZFS storage pool (`zpool`) can be imported in any domain that sees all the storage devices that are part of this `zpool`, regardless of whether the domain sees all these devices as virtual devices or real devices.

## Using SVM on Top of Virtual Disks

Any virtual disk can be used in the SVM local disk set. For example, a virtual disk can be used for storing the SVM metadvice state database, `metadb(1M)`, of the local disk set or for creating SVM volumes in the local disk set.

Any virtual disk whose backend is a SCSI disk can be used in a SVM shared disk set, `metaset(1M)`. Virtual disks whose backends are not SCSI disks cannot be added into a SVM share disk set. Trying to add a virtual disk whose backend is not a SCSI disk into a SVM shared disk set fails with an error similar to the following.

```
# metaset -s test -a c2d2
metaset: domain1: test: failed to reserve any drives
```

## Using VxVM on Top of Virtual Disks

For VxVM support in guest domains, refer to the VxVM documentation from Symantec.

## Other Information and Tasks

---

This chapter contains information and tasks about using the Logical Domains software that are not described in the preceding chapters.

---

### Restrictions on Entering Names in the CLI

The following sections describe the restrictions on entering names in the Logical Domains Manager CLI.

#### File Names (*file*) and Variable Names (*var\_name*)

- First character must be a letter, a number, or a forward slash (/).
- Subsequent letters must be letters, numbers, or punctuation.

#### Virtual Disk Server *backend* and Virtual Switch device Names

- Must contain letters, numbers, or punctuation.

#### Configuration Name (*config\_name*)

The logical domain configuration name (*config\_name*) that you assign to a configuration stored on the system controller must have no more than 64 characters.

## All Other Names

The remainder of the names, such as the logical domain name (*ldom*), service names (*vswitch\_name*, *service\_name*, *vdpcs\_service\_name*, and *vcc\_name*), virtual network name (*if\_name*), and virtual disk name (*disk\_name*), must be in the following format:

- First character must be a letter or number.
- Subsequent characters must be letters, numbers, or any of the following characters: `'-_+#.::~~()'`

---

## Using `ldm list` Subcommands

This section shows the syntax usage for the `ldm` subcommands, defines some output terms, such as flags and utilization statistics, and provides examples of the output.

### Machine-Readable Output

If you are creating scripts that use `ldm list` command output, *always* use the `-p` option to produce the machine-readable form of the output. See [“To Generate a Parseable, Machine-Readable List \(-p\)”](#) on page 88 for more information.

### ▼ To Show Syntax Usage for `ldm` Subcommands

- To look at syntax usage for all `ldm` subcommands, do the following.

#### CODE EXAMPLE 6-1 Syntax Usage for All `ldm` Subcommands

```
primary# ldm --help

Usage:
  ldm [--help] command [options] [properties] operands

Command(s) for each resource (aliases in parens):

    bindings
        list-bindings [-e] [-p] [<ldom>...]

    services
        list-bindings [-e] [-p] [<ldom>...]

    constraints
```

**CODE EXAMPLE 6-1** Syntax Usage for All ldm Subcommands *(Continued)*

```
list-constraints ([-x] | [-e] [-p]) [<ldom>...]

devices
  list-devices [-a] [-p] [cpu] [crypto|mau] [memory] [io]

domain      ( dom )
  add-domain (-i <file> | mac-addr=<num> <ldom> | <ldom>...)
  remove-domain (-a | <ldom>...)
  list-domain [-e] [-l] [-p] [<ldom>...]
  start-domain start-domain (-a | -i <file> | <ldom>...)
  stop-domain stop-domain [-f] (-a | <ldom>...)
  bind-domain (-i <file> | <ldom>)
  unbind-domain <ldom>
  panic-domain <ldom>

io
  add-io [bypass=on] <bus> <ldom>
  remove-io <bus> <ldom>

crypto      ( mau )
  add-crypto <number> <ldom>
  set-crypto <number> <ldom>
  remove-crypto <number> <ldom>

memory      ( mem )
  add-memory <number>[GMK] <ldom>
  set-memory <number>[GMK] <ldom>
  remove-memory <number>[GMK] <ldom>

reconf
  remove-reconf <ldom>

spconfig    ( config )
  add-spconfig <config_name>
  set-spconfig <config_name>
  remove-spconfig <config_name>
  list-spconfig

variable    ( var )
  add-variable <var_name>=<value>... <ldom>
  set-variable <var_name>=<value>... <ldom>
  remove-variable <var_name>... <ldom>
  list-variable [<var_name>...] <ldom>

vconscon    ( vcc )
  add-vconscon port-range=<x>-<y> <vcc_name> <ldom>
  set-vconscon port-range=<x>-<y> <vcc_name>
```

**CODE EXAMPLE 6-1** Syntax Usage for All ldm Subcommands *(Continued)*

```
remove-vconscon [-f] <vcc_name>

vconsole      ( vcons )
    set-vcons [port=[<port-num>]] [group=<group>] [service=<vcc_server>]
<ldom>

vcpu
    add-vcpu <number> <ldom>
    set-vcpu <number> <ldom>
    remove-vcpu <number> <ldom>

vdisk
    add-vdisk [timeout=<seconds>] <disk_name>
<volume_name>@<service_name> <ldom>
    set-vdisk [timeout=<seconds>] [volume=<volume_name>@<service_name>]
<disk_name> <ldom>
    remove-vdisk [-f] <disk_name> <ldom>

vdiskserver ( vds )
    add-vdiskserver <service_name> <ldom>
    remove-vdiskserver [-f] <service_name>

vdpcc         ( ndpsldcc )
    add-vdpcc <vdpcc_name> <service_name> <ldom>
    remove-vdpcc [-f] <vdpcc_name> <ldom>

vdpcs         ( ndpsldcs )
    add-vdpcs <vdpcs_name> <ldom>
    remove-vdpcs [-f] <vdpcs_name>

vdiskserverdevice ( vdsdev )
    add-vdiskserverdevice [options={ro,slice,excl}] <backend>
<volume_name>@<service_name>
    set-vdiskserverdevice options=[{ro,slice,excl}]
<volume_name>@<service_name>
    remove-vdiskserverdevice [-f] <volume_name>@<service_name>

vnet
    add-vnet [mac-addr=<num>] <if_name> <vswitch_name> <ldom>
    set-vnet [mac-addr=<num>] [vswitch=<vswitch_name>] <if_name> <ldom>
    remove-vnet [-f] <if_name> <ldom>

vswitch       ( vsw )
    add-vswitch [mac-addr=<num>] [net-dev=<device>] [mode=<mode>]
<vswitch_name> <ldom>
    set-vswitch [mac-addr=<num>] [net-dev=<device>] [mode=<mode>]
<vswitch_name>
```



**CODE EXAMPLE 6-1**    Syntax Usage for All ldm Subcommands *(Continued)*

```
remove-vswitch [-f] <vswitch_name>

Verb aliases:
  Alias      Verb
  -----
  rm         remove
  ls         list

Command aliases:
  Alias      Command
  -----
  create     add-domain
  destroy    remove-domain
  cancel-reconf remove-reconf
  start      start-domain
  stop       stop-domain
  bind       bind-domain
  unbind     unbind-domain
  panic      panic-domain
```

## Flag Definitions

The following flags can be shown in the output for a domain.

- placeholder
- c       control domain
- d       delayed reconfiguration
- n       normal
- s       starting or stopping
- t       transition
- v       virtual I/O domain

If you use the long (-l) option for the command, the flags are spelled out. If not, you see the letter abbreviation.

The list flag values are position dependent. Following are the values that can appear in each of the five columns from left to right.

Column 1	Column 2	Column 3	Column 4	Column 5
s or -	n or t	d or -	c or -	v or -

## Utilization Statistic Definition

The per virtual CPU utilization statistic (UTIL) is shown on the long (-l) option of the `ldm list` command. The statistic is the percentage of time, since the last statistics display, that the virtual CPU spent executing on behalf of the guest operating system. A virtual CPU is considered to be executing on behalf of the guest operating system except when it has been yielded to the hypervisor. If the guest operating system does not yield virtual CPUs to the hypervisor, the utilization of CPUs in the guest operating system will always show as 100%.

The utilization statistic reported for a logical domain is the average of the virtual CPU utilizations for the virtual CPUs in the domain.

## Examples of Various Lists

### ▼ To Show Software Versions (-V)

- To view the current software versions installed, do the following and you receive a listing similar to the following.

**CODE EXAMPLE 6-2** Software Versions Installed

```
primary$ ldm -V

Logical Domain Manager (v 1.0.3)
  Hypervisor control protocol v 1.0

System PROM:
  Hypervisor      v. 1.5.2          @(#)Hypervisor 1.5.2 2007/09/25 08:39/015
  OpenBoot       v. 4.27.2        @(#)OBP 4.27.2 2007/09/24 16:28
```

### ▼ To Generate a Short List

- To generate a short list for all domains, do the following.

**CODE EXAMPLE 6-3** Short List for All Domains

```
primary$ ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-t-cv		4	1G	0.5%	3d 21h 7m
ldg1	active	-t---	5000	8	1G	23%	2m

## ▼ To Generate a Long List (-l)

- To generate a long list for all domains, do the following.

**CODE EXAMPLE 6-4** Long List for All Domains

```
primary$ ldm list -l
NAME                STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary             active   -t-cv                1      768M      0.0%    0s

VCPU
  VID    PID    UTIL  STRAND
   0      0    0.0%   100%

MEMORY
  RA              PA              SIZE
 0x4000000      0x4000000    768M

IO
  DEVICE          PSEUDONYM          OPTIONS
 pci@780          bus_a
 pci@7c0          bus_b          bypass=on

VCC
  NAME            PORT-RANGE
 vcc0             5000-5100

VSW
  NAME            MAC              NET-DEV    DEVICE    MODE
 vsw0             08:00:20:aa:bb:e0 e1000g0    switch@0   prog,promisc
 vsw1             08:00:20:aa:bb:e1                routed

VDS
  NAME            VOLUME          OPTIONS          DEVICE
 vds0             myvol-a         slice            /disk/a
                  myvol-b                             /disk/b
                  myvol-c         ro,slice,excl    /disk/c
 vds1             myvol-d                             /disk/d

VDPCS
  NAME
 vdpcs0
 vdpcs1

-----
NAME                STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
ldg1                bound    -----    5000     1      512M

VCPU
```

**CODE EXAMPLE 6-4** Long List for All Domains *(Continued)*

VID	PID	UTIL	STRAND		
0	1		100%		
MEMORY					
RA		PA		SIZE	
0x4000000		0x34000000		512M	
NETWORK					
NAME		SERVICE		DEVICE	MAC
mynet-b		vsw0@primary		network@0	08:00:20:ab:9a:12
mynet-a		vsw0@primary		network@1	08:00:20:ab:9a:11
DISK					
NAME		VOLUME		DEVICE	SERVER
mydisk-a		myvol-a@vds0		disk@0	primary
mydisk-b		myvol-b@vds0		disk@1	primary
VDPCC					
NAME		SERVICE			
myvdpcc-a		vdpcs0@primary			
myvdpcc-b		vdpcs0@primary			
VCONS					
NAME		SERVICE		PORT	
mygroup		vcc0@primary		5000	

▼ To Generate an Extended List (-e)

- To generate an extended list of all domains, do the following.

**CODE EXAMPLE 6-5** Extended List for all Domains

primary\$ ldm list -e							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-t-cv		1	768M	0.0%	0s
VCPU							
VID	PID	UTIL	STRAND				
0	0	0.0%	100%				
MEMORY							
RA		PA		SIZE			
0x4000000		0x4000000		768M			
IO							
DEVICE		PSEUDONYM		OPTIONS			

**CODE EXAMPLE 6-5**    Extended List for all Domains (*Continued*)

pci@780                      bus_a							
pci@7c0                      bus_b                      bypass=on							
VLDC							
NAME							
primary							
VCC							
NAME                      PORT-RANGE							
vcc0                      5000-5100							
VSW							
NAME                      MAC                      NET-DEV                      DEVICE                      MODE							
vsw0                      08:00:20:aa:bb:e0    e1000g0                      switch@0                      prog,promisc							
vsw1                      08:00:20:aa:bb:e1    routed							
VDS							
NAME                      VOLUME                      OPTIONS                      DEVICE							
vds0                      myvol-a                      slice                      /disk/a							
myvol-b    /disk/b							
myvol-c                      ro,slice,excl                      /disk/c							
vds1                      myvol-d    /disk/d							
VDPCS							
NAME							
vdpcs0							
vdpcs1							
VLDCC							
NAME                      SERVICE    DESC							
hvctl                      primary@primary    hvctl							
vldcc0                      primary@primary    ds							
-----							
NAME                      STATE                      FLAGS                      CONS                      VCPU                      MEMORY                      UTIL                      UPTIME							
ldg1                      bound                      -----                      5000                      1                      512M							
VCPU							
VID                      PID                      UTIL                      STRAND							
0                      1    100%							
MEMORY							
RA                                      PA                                      SIZE							
0x4000000                      0x34000000                      512M							
VLDCC							
NAME                                      SERVICE    DESC							

**CODE EXAMPLE 6-5**    Extended List for all Domains (*Continued*)

vldcc0	primary@primary		ds	
NETWORK				
NAME	SERVICE	DEVICE	MAC	
mynet-b	vsw0@primary	network@0	08:00:20:ab:9a:12	
mynet-a	vsw0@primary	network@1	08:00:20:ab:9a:11	
DISK				
NAME	VOLUME	DEVICE	SERVER	
mydisk-a	myvol-a@vds0	disk@0	primary	
mydisk-b	myvol-b@vds0	disk@1	primary	
VDPCC				
NAME	SERVICE			
myvdpcc-a	vdpcs0@primary			
myvdpcc-b	vdpcs0@primary			
VCONS				
NAME	SERVICE	PORT		
mygroup	vcc0@primary	5000		

▼ To Generate a Parseable, Machine-Readable List (-p)

- To generate a parseable, machine-readable list of all domains, do the following.

**CODE EXAMPLE 6-6**    Machine-Readable List

```
primary$ ldm list -p
VERSION 1.0
DOMAIN|name=primary|state=active|flags=-t-cv|cons=|ncpu=1|mem=805306368|util=
0.0|uptime=0
DOMAIN|name=ldg1|state=bound|flags=-----|cons=5000|ncpu=1|mem=536870912|util=
|uptime=
```

▼ To Show the Status of a Domain

- To look at the status of a domain (for example, guest domain ldg1), do the following.

**CODE EXAMPLE 6-7**    Domain Status

```
primary# ldm list-domain ldg1
NAME          STATE   FLAGS   CONS   VCPU  MEMORY  UTIL  UPTIME
ldg1          active -t---   5000    8     1G     0.3%  2m
```

## ▼ To List a Variable

- To list a variable (for example, `boot-device`) for a domain (for example, `ldg1`), do the following.

**CODE EXAMPLE 6-8** Variable List for a Domain

```
primary$ ldm list-variable boot-device ldg1
boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
```

## ▼ To List Bindings

- To list resources that are bound for a domain (for example, `ldg1`) do the following.

**CODE EXAMPLE 6-9** Bindings List for a Domain

```
primary$ ldm list-bindings ldg1
NAME                STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
ldg1                 bound    -----  5000    1       512M

VCPU
  VID    PID    UTIL  STRAND
  0      1      100%

MEMORY
  RA              PA              SIZE
  0x4000000       0x34000000      512M

NETWORK
  NAME                SERVICE                DEVICE    MAC
  mynet-b             vsw0@primary           network@0 08:00:20:ab:9a:12
    PEER
    vsw0@primary       08:00:20:aa:bb:e0
  mynet-a@ldg1        08:00:20:ab:9a:11
  mynet-c@ldg2        08:00:20:ab:9a:22
  NAME                SERVICE                DEVICE    MAC
  mynet-a             vsw0@primary           network@1 08:00:20:ab:9a:11
    PEER
    vsw0@primary       08:00:20:aa:bb:e0
  mynet-b@ldg1        08:00:20:ab:9a:12
  mynet-c@ldg2        08:00:20:ab:9a:22

DISK
  NAME                VOLUME                DEVICE    SERVER
  mydisk-a            myvol-a@vds0          disk@0    primary
  mydisk-b            myvol-b@vds0          disk@1    primary
```

#### CODE EXAMPLE 6-9 Bindings List for a Domain *(Continued)*

VDPCC		
NAME	SERVICE	
myvdpcc-a	vdpcs0@primary	
myvdpcc-b	vdpcs0@primary	
VCONS		
NAME	SERVICE	PORT
mygroup	vcc0@primary	5000

## ▼ To List Configurations

- To list logical domain configurations that have been stored on the SC, do the following.

#### CODE EXAMPLE 6-10 Configurations List

```
primary$ ldm list-config
factory-default [current]
initial [next]
```

### Meaning of Labels

The labels to the right of the configuration name mean the following:

- current - configuration currently being used
- next - configuration to be used at the next power cycle

## ▼ To List Devices

- To list all server resources, bound and unbound, do the following.

#### CODE EXAMPLE 6-11 List of All Server Resources

```
primary$ ldm list-devices -a
VCPU
  PID  %FREE
  0      0
  1      0
  2      0
  3      0
  4     100
  5     100
  6     100
  7     100
```



**CODE EXAMPLE 6-11** List of All Server Resources *(Continued)*

8	100		
9	100		
10	100		
11	100		
12	100		
13	100		
14	100		
15	100		
16	100		
17	100		
18	100		
19	100		
20	100		
21	100		
22	100		
23	100		
24	100		
25	100		
26	100		
27	100		
28	100		
29	100		
30	100		
31	100		
MAU			
CPUSET		BOUND	
(0, 1, 2, 3)		ldg2	
(4, 5, 6, 7)			
(8, 9, 10, 11)			
(12, 13, 14, 15)			
(16, 17, 18, 19)			
(20, 21, 22, 23)			
(24, 25, 26, 27)			
(28, 29, 30, 31)			
MEMORY			
PA	SIZE	BOUND	
0x0	512K	_sys_	
0x80000	1536K	_sys_	
0x200000	62M	_sys_	
0x4000000	768M	primary	
0x34000000	512M	ldg1	
0x54000000	8M	_sys_	
0x54800000	2G	ldg2	
0xd4800000	29368M		

**CODE EXAMPLE 6-11** List of All Server Resources *(Continued)*

IO				
	DEVICE	PSEUDONYM	BOUND	OPTIONS
	pci@780	bus_a	yes	
	pci@7c0	bus_b	yes	bypass=on

## ▼ To List Services

- To list the services that are available, do the following.

**CODE EXAMPLE 6-12** Services List

primary\$ <b>ldm list-services</b>				
VDS				
	NAME	VOLUME	OPTIONS	DEVICE
	primary-vds0			
VCC				
	NAME	PORT-RANGE		
	primary-vcc0	5000-5100		
VSW				
	NAME	MAC	NET-DEV	DEVICE MODE
	primary-vsw0	00:14:4f:f9:68:d0	e1000g0	switch@0 prog,promisc

## Listing Constraints

To the Logical Domains Manager, constraints are one or more resources you want to have assigned to a particular domain. You either receive all the resources you ask to be added to a domain or you get none of them, depending upon the available resources. The `list-constraints` subcommand lists those resources you requested assigned to the domain.

## ▼ To List Constraints for One Domain

- To list constraints for one domain (for example, `ldg1`) do the following.

**CODE EXAMPLE 6-13** Constraints List for One Domain

primary\$ <b>ldm list-constraints ldg1</b>	
DOMAIN	
ldg1	
VCPU	
COUNT	

**CODE EXAMPLE 6-13** Constraints List for One Domain (*Continued*)

1

MEMORY				
SIZE				
512M				
NETWORK				
NAME	SERVICE	DEVICE	MAC	
mynet-b	vsw0	network@0	08:00:20:ab:9a:12	
mynet-b	vsw0	network@0	08:00:20:ab:9a:12	
DISK				
NAME	VOLUME			
mydisk-a	myvol-a@vds0			
mydisk-b	myvol-b@vds0			
VDPCC				
NAME	SERVICE			
myvdpcc-a	vdpcs0@primary			
myvdpcc-b	vdpcs0@primary			
VCONS				
NAME	SERVICE			
mygroup	vcc0			

## ▼ To List Constraints in XML Format

- To list constraints in XML format for a particular domain (for example, `ldg1`), do the following.

**CODE EXAMPLE 6-14** Constraints for a Domain in XML Format

```
primary$ ldm list-constraints -x ldg1
<?xml version="1.0"?>
<LDM_interface version="1.0">
  <data version="2.0">
    <ldom>
      <ldom_info>
        <ldom_name>ldg1</ldom_name>
      </ldom_info>
      <cpu>
        <number>8</number>
      </cpu>
      <memory>
        <size>1G</size>
      </memory>
```

**CODE EXAMPLE 6-14** Constraints for a Domain in XML Format *(Continued)*

```
<network>
  <vnet_name>vnet0</vnet_name>
  <service_name>primary-vsw0</service_name>
  <mac_address>01:14:4f:fa:0f:55</mac_address>
</network>
<disk>
  <vdisk_name>vdisk0</vdisk_name>
  <service_name>primary-vds0</service_name>
  <vol_name>vol0</vol_name>
</disk>
<var>
  <name>boot-device</name>
  <value>/virtual-devices@100/channel-devices@200/disk@0:a</value>
</var>
<var>
  <name>nvrarc</name>
  <value>dealias vnet0 /virtual-devices@100/channel-devices@200/
network@0</value>
</var>
<var>
  <name>use-nvrarc?</name>
  <value>true</value>
</var>
</ldom>
</data>
</LDM_interface>
```

## ▼ To List Constraints in a Machine-Readable Format

- To list constraints for all domains in a parseable format, do the following.

**CODE EXAMPLE 6-15** Constraints for All Domains in a Machine-Readable Format

```
primary$ ldm list-constraints -p
VERSION 1.0
DOMAIN|name=primary
MAC|mac-addr=00:03:ba:d8:b1:46
VCPU|count=4
MEMORY|size=805306368
IO
|dev=pci@780|alias=
|dev=pci@7c0|alias=
VDS|name=primary-vds0
|vol=disk-ldg2|opts=|dev=/ldoms/nv72-ldg2/disk
```

```
|vol=vol0|opts=|dev=/ldoms/nv72-ldg1/disk
VCC|name=primary-vcc0|port-range=5000-5100
VSW|name=primary-vsw0|mac-addr=|net-dev=e1000g0|dev=switch@0
DOMAIN|name=ldg1
VCPU|count=8
MEMORY|size=1073741824
VARIABLES
|boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
|nvramrc=devalias vnet0 /virtual-devices@100/channel-devices@200/network@0
|use-nvramrc?=true
VNET|name=vnet0|dev=network@0|service=primary-vsw0|mac-addr=01:14:4f:fa:0f:55
VDISK|name=vdisk0|vol=vol0@primary-vds0
```



# The `ldm stop-domain` Command Can Time Out If the Domain Is Heavily Loaded

An `ldm stop-domain` command can time out before the domain completes shutting down. When this happens, an error similar to the following is returned by the Logical Domains Manager.

```
LDom ldg8 stop notification failed
```

However, the domain could still be processing the shutdown request. Use the `ldm list-domain` command to verify the status of the domain. For example:

# <code>ldm list-domain ldg8</code>							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg8	active	s----	5000	22	3328M	0.3%	1d 14h 31m

The preceding list shows the domain as active, but the `s` flag indicates that the domain is in the process of stopping. This should be a transitory state.

The following example shows the domain has now stopped.

# <code>ldm list-domain ldg8</code>							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg8	bound	-----	5000	22	3328M		

---

# Determining the Solaris Network Interface Name Corresponding to a Virtual Network Device

There is no way to determine the Solaris OS network interface name on a guest, corresponding to a given virtual device, directly from the output provided by the `ldm list-*` commands. However, you can do this by using a combination of the output from `ldm list -l` command and the entries under `/devices` on the Solaris OS guest.

## ▼ To Find Solaris OS Network Interface Name

In this example, guest domain `ldg1` contains two virtual network devices, `net-a` and `net-c`. To find the Solaris OS network interface name in `ldg1` that corresponds to `net-c`, do the following.

1. Use the `ldm` command to find the virtual network device instance for `net-c`.

```
# ldm list -l ldg1
...
NETWORK
NAME          SERVICE          DEVICE          MAC
net-a         primary-vsw0@primary  network@0       00:14:4f:f8:91:4f
net-c         primary-vsw0@primary  network@2       00:14:4f:f8:dd:68
...
#
```

The virtual network device instance for `net-c` is `network@2`.

2. To find the corresponding network interface on `ldg1`, log into `ldg1` and find the entry for this instance under `/devices`.

```
# uname -n
ldg1
# find /devices/virtual-devices@100 -type c -name network@2\*
/devices/virtual-devices@100/channel-devices@200/network@2:vnet1
#
```

The network interface name is the part of the entry after the colon; that is, `vnet1`.

**3. Plumb vnet1 to see that it has the MAC address 00:14:4f:f8:dd:68 as shown in the `ldm list -l` output for `net-c` in Step 1.**

```
# ifconfig vnet1
vnet1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 0.0.0.0 netmask 0
        ether 0:14:4f:f8:dd:68
#
```

---

## Assigning MAC Addresses Automatically or Manually

You must have enough media access control (MAC) addresses to assign to the number of logical domains, virtual switches, and virtual networks you are going to use. You can have the Logical Domains Manager automatically assign MAC addresses to a logical domain, a virtual network (vnet), and a virtual switch (vswitch), or you can manually assign MAC addresses from your own pool of assigned MAC addresses. The `ldm` subcommands that set MAC addresses are `add-domain`, `add-vsw`, `set-vsw`, `add-vnet`, and `set-vnet`. If you do not specify a MAC address in these subcommands, the Logical Domains Manager assigns one automatically.

The advantage to having the Logical Domains Manager assign the MAC addresses is that it utilizes the block of MAC addresses dedicated for use with logical domains. Also, the Logical Domains Manager detects and prevents MAC address collisions with other Logical Domains Manager instances on the same subnet. This frees you from having to manually manage your pool of MAC addresses.

MAC address assignment happens as soon as a logical domain is created or a network device is configured into a domain. In addition, the assignment is persistent until the device, or the logical domain itself, is removed.

The following topics are addressed in this section:

- [“Range of MAC Addresses Assigned to Logical Domains Software” on page 98](#)
- [“Automatic Assignment Algorithm” on page 98](#)
- [“Duplicate MAC Address Detection” on page 99](#)
- [“Freed MAC Addresses” on page 99](#)

# Range of MAC Addresses Assigned to Logical Domains Software

Logical domains have been assigned the following block of 512K MAC addresses:

00:14:4F:F8:00:00 ~ 00:14:4F:FF:FF:FF

The lower 256K addresses are used by the Logical Domains Manager for **automatic MAC address allocation**, and you *cannot* manually request an address in this range:

00:14:4F:F8:00:00 - 00:14:4F:FB:FF:FF

You can use the upper half of this range for **manual MAC address allocation**:

00:14:4F:FC:00:00 - 00:14:4F:FF:FF:FF

## Automatic Assignment Algorithm

When you do not specify a MAC address in creating logical domain or a network device, the Logical Domains Manager automatically allocates and assigns a MAC address to that logical domain or network device. To obtain this MAC address, the Logical Domains Manager iteratively attempts to select an address and then checks for potential collisions.

Before selecting a potential address, the Logical Domains Manager first looks to see if it has a recently freed, automatically assigned address saved in a database for this purpose (see [“Freed MAC Addresses” on page 99](#)). If so, the Logical Domains Manager selects its candidate address from the database.

If no recently freed addresses are available, the MAC address is randomly selected from the 256K range of addresses set aside for this purpose. The MAC address is selected randomly to lessen the chance of a duplicate MAC address being selected as a candidate.

The address selected is then checked against other Logical Domains Managers on other systems to prevent duplicate MAC addresses from actually being assigned. The algorithm employed is described in [“Duplicate MAC Address Detection” on page 99](#). If the address is already assigned, the Logical Domains Manager iterates, choosing another address, and again checking for collisions. This continues until a MAC address is found that is not already allocated, or a time limit of 30 seconds has elapsed. If the time limit is reached, then the creation of the device fails, and an error message similar to the following is shown.

Automatic MAC allocation failed. Please set the vnet MAC address manually.



# Duplicate MAC Address Detection

To prevent the same MAC address from being allocated to different devices, one Logical Domains Manager checks with other Logical Domains Managers on other systems by sending a multicast message over the control domain's default network interface, including the address that the Logical Domain Manager wants to assign to the device. The Logical Domains Manager attempting to assign the MAC address waits for one second for a response back. If a different device on another LDom-enabled system has already been assigned that MAC address, the Logical Domains Manager on that system sends back a response containing the MAC address in question. If the requesting Logical Domains Manager receives a response, it knows the chosen MAC address has already been allocated, chooses another, and iterates.

By default, these multicast messages are sent only to other managers on the same subnet; the default time-to-live (TTL) is 1. The TTL can be configured using the Service Management Facilities (SMF) property `ldmd/hops`.

Each Logical Domains Manager is responsible for:

- Listening for multicast messages
- Keeping track of MAC addresses assigned to its domains
- Looking for duplicates
- Responding so that duplicates do not occur

If the Logical Domains Manager on a system is shut down for any reason, duplicate MAC addresses could occur while the Logical Domains Manager is down.

Automatic MAC allocation occurs at the time the logical domain or network device is created and persists until the device or the logical domain is removed.

## Freed MAC Addresses

When a logical domain or a device associated with an automatic MAC address is removed, that MAC address is saved in a database of recently freed MAC addresses for possible later use on that system. These MAC addresses are saved to prevent the exhaustion of Internet Protocol (IP) addresses from a Dynamic Host Configuration Protocol (DHCP) server. When DHCP servers allocate IP addresses, they do so for a period of time (the lease time). The lease duration is often configured to be quite long, generally hours or days. If network devices are created and removed at a high rate without the Logical Domains Manager reusing automatically allocated MAC addresses, the number of MAC addresses allocated could soon overwhelm a typically configured DHCP server.

When a Logical Domains Manager is requested to automatically obtain a MAC address for a logical domain or network device, it first looks to the freed MAC address database to see if there is a previously assigned MAC address it can reuse. If there is a MAC address available from this database, the duplicate MAC address detection algorithm is run. If the MAC address had not been assigned to someone else since it was previously freed, it will be reused and removed from the database. If a collision is detected, the address is simply removed from the database. The Logical Domains Manager then either tries the next address in the database or if none is available, randomly picks a new MAC address.

---

## CPU and Memory Address Mapping

The Solaris Fault Management Architecture (FMA) reports CPU errors in terms of physical CPU numbers and memory errors in terms of physical memory addresses.

If you want to determine within which logical domain an error occurred and the corresponding virtual CPU number or real memory address within the domain, then you must perform a mapping.

### CPU Mapping

The domain and the virtual CPU number within the domain, which correspond to a given physical CPU number, can be determined with the following procedures.

#### ▼ To Determine the CPU Number

1. Generate a long parseable list for all domains.

```
primary$ ldm ls -l -p
```

2. Look for the entry in the list's `VCPU` sections that has a `pid` field equal to the physical CPU number.
  - If you find such an entry, the CPU is in the domain the entry is listed under, and the virtual CPU number within the domain is given by the entry's `vid` field.
  - If you do not find such an entry, the CPU is not in any domain.

# Memory Mapping

The domain and the real memory address within the domain, which correspond to a given physical memory address (PA), can be determined as follows.

## ▼ To Determine the Real Memory Address

1. Generate a long parseable list for all domains.

```
primary$ ldm ls -l -p
```

2. Look for the line in the list's MEMORY sections where the PA falls within the inclusive range  $pa$  to  $(pa + size - 1)$ ; that is,  $pa \leq PA < (pa + size - 1)$ .

Here  $pa$  and  $size$  refer to the values in the corresponding fields of the line.

- If you find such an entry, the PA is in the domain the entry is listed under and the corresponding real address within the domain is given by  $ra + (PA - pa)$ .
- If you do not find such an entry, the PA is not in any domain.

## Examples of CPU and Memory Mapping

Suppose you have a logical domain configuration as shown in [CODE EXAMPLE 6-16](#), and you want to determine the domain and the virtual CPU corresponding to physical CPU number 5, and the domain and the real address corresponding to physical address 0x7e816000.

Looking through the VCPU entries in the list for the one with the `pid` field equal to 5, you can find the following entry under logical domain `ldg1`.

```
|vid=1|pid=5|util=29|strand=100
```

Hence, the physical CPU number 5 is in domain `ldg1` and within the domain it has virtual CPU number 1.

Looking through the MEMORY entries in the list, you can find the following entry under domain `ldg2`.

```
ra=0x8000000|pa=0x78000000|size=1073741824
```

Where  $0x78000000 \leq 0x7e816000 \leq (0x78000000 + 1073741824 - 1)$ ; that is,  $pa \leq PA < (pa + size - 1)$ .

Hence, the PA is in domain ldg2 and the corresponding real address is 0x8000000 + (0x7e816000 - 0x78000000) = 0xe816000.

**CODE EXAMPLE 6-16** Long Parseable List of Logical Domains Configurations

```
primary$ ldm ls -l -p
VERSION 1.0
DOMAIN|name=primary|state=active|flags=normal,control,vio-service|cons=
SP|ncpu=4|mem=1073741824|util=0.6|uptime=64801|softstate=Solaris running
VCPU
|vid=0|pid=0|util=0.9|strand=100
|vid=1|pid=1|util=0.5|strand=100
|vid=2|pid=2|util=0.6|strand=100
|vid=3|pid=3|util=0.6|strand=100
MEMORY
|ra=0x8000000|pa=0x8000000|size=1073741824
IO
|dev=pci@780|alias=bus_a
|dev=pci@7c0|alias=bus_b
VDS|name=primary-vds0|nclients=2
|vol=disk-ldg1|opts=|dev=/opt/ldoms/testdisk.1
|vol=disk-ldg2|opts=|dev=/opt/ldoms/testdisk.2
VCC|name=primary-vcc0|nclients=2|port-range=5000-5100
VSW|name=primary-vsw0|nclients=2|mac-addr=00:14:4f:fb:42:5c|net-dev=
e1000g0|dev=switch@0|mode=prog,promisc
VCONS|type=SP
DOMAIN|name=ldg1|state=active|flags=normal|cons=5000|ncpu=2|mem=
805306368|util=29|uptime=903|softstate=Solaris running
VCPU
|vid=0|pid=4|util=29|strand=100
|vid=1|pid=5|util=29|strand=100
MEMORY
|ra=0x8000000|pa=0x48000000|size=805306368
VARIABLES
|auto-boot?=true
|boot-device=/virtual-devices@100/channel-devices@200/disk@0
VNET|name=net|dev=network@0|service=primary-vsw0@primary|mac-addr=
00:14:4f:f9:8f:e6
VDISK|name=vdisk-1|vol=disk-ldg1@primary-vds0|dev=disk@0|server=primary
VCONS|group=group1|service=primary-vcc0@primary|port=5000
DOMAIN|name=ldg2|state=active|flags=normal|cons=5001|ncpu=3|mem=
1073741824|util=35|uptime=775|softstate=Solaris running
VCPU
|vid=0|pid=6|util=35|strand=100
|vid=1|pid=7|util=34|strand=100
|vid=2|pid=8|util=35|strand=100
MEMORY
|ra=0x8000000|pa=0x78000000|size=1073741824
VARIABLES
```

```
| auto-boot?=true  
| boot-device=/virtual-devices@100/channel-devices@200/disk@0  
VNET | name=net | dev=network@0 | service=primary-vsw0@primary | mac-addr=  
00:14:4f:f9:8f:e7  
VDISK | name=vdisk-2 | vol=disk-ldg2@primary-vds0 | dev=disk@0 | server=primary  
VCONS | group=group2 | service=primary-vcc0@primary | port=5000
```

---

## Configuring Split PCI Express Bus to Use Multiple Logical Domains

---

**Note** – For Sun UltraSPARC T-2 based servers, such as the Sun SPARC Enterprise T5120 and T5220 servers, you would assign a Network Interface Unit (NIU) to the logical domain rather than use this procedure.

---

The PCI Express (PCI-E) bus on a Sun UltraSPARC T1-based server consists of two ports with various leaf devices attached to them. These are identified on a server with the names `pci@780 (bus_a)` and `pci@7c0 (bus_b)`. In a multidomain environment, the PCI-E bus can be programmed to assign each leaf to a separate domain using the Logical Domains Manager. Thus, you can enable more than one domain with direct access to physical devices instead of using I/O virtualization.

When the Logical Domains system is powered on, the control (`primary`) domain uses all the physical device resources, so the primary domain owns both the PCI-E bus leaves.



---

**Caution** – All internal disks on the supported servers are connected to a single leaf. If a control domain is booted from an internal disk, do not remove that leaf from the domain. Also, ensure that you are not removing the leaf with the primary network port. If you remove the wrong leaf from the control or service domain, that domain would not be able to access required devices and would become unusable. If the primary network port is on a different bus than the system disk, then move the network cable to an onboard network port and use the Logical Domains Manager to reconfigure the virtual switch (`vsw`) to reflect this change.

---

## ▼ To Create a Split PCI Configuration

The example shown here is for a Sun Fire T2000 server. This procedure also can be used on other Sun UltraSPARC T1-based servers, such as a Sun Fire T1000 server and a Netra T2000 server. The instructions for different servers might vary slightly from these, but you can obtain the basic principles from the example. Mainly, you need to retain the leaf that has the boot disk and remove the other leaf from the primary domain and assign it to another domain.

1. Verify that the primary domain owns both leaves of the PCI Express bus.

```
primary# ldm list-bindings primary
...
IO
    DEVICE          PSEUDONYM      OPTIONS
    pci@780         bus_a
    pci@7c0         bus_b
    ...
```

2. Determine the device path of the boot disk, which needs to be retained.

```
primary# df /
/                (/dev/dsk/c1t0d0s0 ): 1309384 blocks  457028 files
```

3. Determine the physical device to which the block device c1t0d0s0 is linked.

```
primary# ls -l /dev/dsk/c1t0d0s0
lrwxrwxrwx  1 root    root          65 Feb  2 17:19 /dev/dsk/c1t0d0s0 -> ../
../devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@0,0:a
```

In this example, the physical device for the boot disk for domain primary is under the leaf pci@7c0, which corresponds to our earlier listing of bus\_b. This means that we can assign bus\_a (pci@780) of the PCI-Express bus to another domain.

4. Check /etc/path\_to\_inst to find the physical path of the onboard network ports.

```
primary# grep e1000g /etc/path_to_inst
```

5. Remove the leaf that does not contain the boot disk (pci@780 in this example) from the primary domain.

```
primary# ldm remove-io pci@780 primary
```

6. Add this split PCI configuration (split-cfg in this example) to the system controller.

```
primary# ldm add-config split-cfg
```

This configuration (split-cfg) is also set as the next configuration to be used after the reboot.

---

**Note** – Currently, there is a limit of 8 configurations that can be saved on the SC, not including the factory-default configuration.

---

7. Reboot the primary domain so that the change takes effect.

```
primary# shutdown -i6 -g0 -y
```

8. Add the leaf (pci@780 in this example) to the domain (ldg1 in this example) that needs direct access.

```
primary# ldm add-io pci@780 ldg1
```

Notice: the LDom Manager is running in configuration mode. Any configuration changes made will only take effect after the machine configuration is downloaded to the system controller and the host is reset.

If you have an Infiniband card, you might need to enable the bypass mode on the pci@780 bus. See [“Enabling the I/O MMU Bypass Mode on a PCI Bus” on page 106](#) for information on whether you need to enable the bypass mode.

9. Reboot domain ldg1 so that the change takes effect.

All domains must be inactive for this reboot. If you are configuring this domain for the first time, the domain will be inactive.

```
ldg1# shutdown -i6 -g0 -y
```

10. Confirm that the correct leaf is still assigned to the primary domain and the correct leaf is assigned to domain ldg1.

primary# <b>ldm list-bindings primary</b>							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv	SP	4	4G	0.4%	18h 25m
...							
IO							
DEVICE		PSEUDONYM					
pci@7c0		bus_b					
...							
-----							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg1	active	-n---	5000	4	2G	10%	35m
...							
IO							
DEVICE		PSEUDONYM					
pci@780		bus_a					
...							

This output confirms that the PCI-E leaf bus\_b and the devices below it are assigned to domain primary, and bus\_a and its devices are assigned to ldg1.

---

## Enabling the I/O MMU Bypass Mode on a PCI Bus

If you have an Infiniband Host Channel Adapter (HCA) card, you might need to turn the I/O memory management unit (MMU) bypass mode on. By default, Logical Domains software controls PCI-E transactions so that a given I/O device or PCI-E option can only access the physical memory assigned within the I/O domain. Any attempt to access memory of another guest domain is prevented by the I/O MMU. This provides a higher level of security between the I/O domain and all other domains. However, in the rare case where a PCI-E or PCI-X option card does not load or operate with the I/O MMU bypass mode off, this option allows you to turn the I/O MMU bypass mode on. However, if you turn the bypass mode on, there no longer is a hardware-enforced protection of memory accesses from the I/O domain.

The `bypass=on` option turns on the I/O MMU bypass mode. This bypass mode should be enabled only if the respective I/O domain and I/O devices within that I/O domain are trusted by all guest domains. This example turns on the bypass mode.

```
primary# ldm add-io bypass=on pci@780 ldg1
```



The output shows `bypass=on` under `OPTIONS`.

---

## Using Console Groups

The virtual network terminal server daemon, `vntsd(1M)`, enables you to provide access for multiple domain consoles using a single TCP port. At the time of domain creation, the Logical Domains Manager assigns a unique TCP port to each console by creating a new default group for that domain's console. The TCP port is then assigned to the console group as opposed to the console itself. The console can be bound to an existing group using the `set-vcons` subcommand.

### ▼ To Combine Multiple Consoles Into One Group

#### 1. Bind the consoles for the domains into one group.

The following example shows binding the console for three different domains (`ldg1`, `ldg2`, and `ldg3`) to the same console group (`group1`).

```
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg1
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg2
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg3
```

#### 2. Connect to the associated TCP port (`localhost` at port 5000 in this example).

```
# telnet localhost 5000
primary-vnts-group1: h, l, c{id}, n{name}, q:
```

You are prompted to select one of the domain consoles.

#### 3. List the domains within the group by selecting `l` (list).

```
primary-vnts-group1: h, l, c{id}, n{name}, q: l
DOMAIN ID      DOMAIN NAME      DOMAIN STATE
0               ldg1             online
1               ldg2             online
2               ldg3             online
```

---

**Note** – To re-assign the console to a different group or `vcc` instance, the domain must be unbound; that is, it has to be in the inactive state. Refer to the Solaris 10 OS `vntsd(1M)` man page for more information on configuring and using SMF to manage `vntsd` and using console groups.

---

---

## Moving a Logical Domain From One Server to Another

You can move a logical domain, which is not running, from one server to another. Before you move the domain, if you set up the same domain on two servers, the domain would be easier to move. In fact, you do not have to move the domain itself; you only have to unbind and stop the domain on one server and bind and start the domain on the other server.

### ▼ To Set Up Domains to Move

1. Create a domain with the same name on two servers; for example, create `domainA1` on `serverA` and `serverB`.
2. Add a virtual disk server device and a virtual disk to both servers. The virtual disk server opens the underlying device for export as part of the bind.
3. Bind the domain only on one server; for example, `serverA`. Leave the domain inactive on the other server.

### ▼ To Move the Domain

1. Unbind and stop the domain on `serverA`.
2. Bind and start the domain on `serverB`.

---

**Note** – No resources are used until you bind the domain.

---

---

# Removing Logical Domains

This section describes how to remove all guest domains and revert to a single OS instance that controls the whole server.

## ▼ To Remove All Guest Logical Domains

1. List all the logical domain configurations on the system controller.

```
primary# ldm ls-config
```

2. Remove all configurations (*config\_name*) previously saved to the system controller (SC). Use the following command for each such configuration.

```
primary# ldm rm-config config_name
```

Once you remove all the configurations previously saved to the SC, the factory-default domain would be the next one to use when the control domain (primary) is rebooted.

3. Stop all guest domains using the `-a` option.

```
primary# ldm stop-domain -a
```

4. List all domains to see all the resources attached to guest domains.

```
primary# ldm ls
```

5. Release all the resources attached to guest domains. To do this, use the `ldm unbind-domain` command for each guest domain (*ldom*) configured in your system.

---

**Note** – You might not be able to unbind an I/O domain in a split-PCI configuration if it is providing services required by the control domain. In this situation, skip this step.

---

```
primary# ldm unbind-domain ldom
```

## 6. Stop the control domain.

```
primary# shutdown -i1 -g0 -y
```

## 7. Power-cycle the system controller so that the factory-default configuration is reloaded.

```
SC> poweroff  
SC> poweron
```

---

# Operating the Solaris OS With Logical Domains

This section describes the changes in behavior in using the Solaris OS that occur once a configuration created by the Logical Domains Manager is instantiated; that is, domaining is enabled.

---

**Note** – Any discussion about whether domaining is enabled pertains only to Sun UltraSPARC T1-based platforms. Otherwise, domaining is always enabled.

---

## OpenBoot Firmware Not Available After Solaris OS Has Started If Domaining Is Enabled

Domaining is enabled once a logical domains configuration created by the Logical Domains Manager is instantiated. If domaining is enabled, the OpenBoot™ firmware is not available after the Solaris OS has started, because it is removed from memory.

To reach the `ok` prompt from the Solaris OS, you must halt the domain. You can use the Solaris OS `halt` command to halt the domain.

## Power-Cycling a Server

Whenever performing any maintenance on a system running LDoms software that requires power-cycling the server, you must save your current logical domain configurations to the SC first.

## ▼ To Save Your Current Logical Domain Configurations to the SC

- Use the following command.

```
# ldm add-config config_name
```

## Result of an OpenBoot power-off Command

The OpenBoot™ `power-off` command does *not* power down a system. To power down a system while in OpenBoot firmware, use your system controller's or system processor's `poweroff` command. The OpenBoot `power-off` command displays the following message.

```
NOTICE: power-off command is not supported, use appropriate  
NOTICE: command on System Controller to turn power off.
```

## Result of Solaris OS Breaks

If domaining is not enabled, the Solaris OS normally goes to the OpenBoot prompt after a break is issued. The behavior described in this section is seen in two situations:

1. You press the L1-A key sequence when the input device is set to keyboard.
2. You enter the `send break` command when the virtual console is at the `telnet` prompt.

If domaining is enabled, you receive the following prompt after these types of breaks.

```
c)ontinue, s)ync, r) reboot, h)alt?
```

Type the letter that represents what you want the system to do after these types of breaks.

## Results from Halting or Rebooting the Control Domain

The following table shows the expected behavior of halting or rebooting the control (primary) domain.

---

**Note** – The question in [TABLE 6-1](#) regarding whether domaining is enabled pertains only to the Sun UltraSPARC T1 processors. Otherwise, domaining is always enabled.

---

**TABLE 6-1** Expected Behavior of Halting or Rebooting the Control (primary) Domain

Command	Domaining Enabled?	Other Domain Configured?	Behavior
halt	Disabled	N/A	<b>For Sun UltraSPARC T1 Processors:</b> Drops to the ok prompt.
	Enabled	No	<b>For Sun UltraSPARC T1 Processors:</b> System either resets and goes to the OpenBoot ok prompt or goes to the following prompt: r)ebboot, o)k prompt, or h)alt? <b>For Sun UltraSPARC T2 Processors:</b> Host powered off and stays off until powered on at the SC.
	Enabled	Yes	Soft resets and boots up if the variable auto-boot?=true. Soft resets and halts at ok prompt if the variable auto-boot?=false.
reboot	Disabled	N/A	<b>For Sun UltraSPARC T1 Processors:</b> Powers off and powers on the host.
	Enabled	No	<b>For Sun UltraSPARC T1 Processors:</b> Powers off and powers on the host. <b>For Sun UltraSPARC T2 Processors:</b> Reboots the host, no power off.
	Enabled	Yes	<b>For Sun UltraSPARC T1 Processors:</b> Powers off and powers on the host. <b>For Sun UltraSPARC T2 Processors:</b> Reboots the host, no power off.
shutdown -i 5	Disabled	N/A	<b>For Sun UltraSPARC T1 Processors:</b> Powers off the host.
	Enabled	No	Host powered off, stays off until powered on at the SC.
	Enabled	Yes	Soft resets and reboots.

---

---

# Using LDomS With ALOM CMT

The section describes information to be aware of in using Advanced Lights Out Manager (ALOM) chip multithreading (CMT) with the Logical Domains Manager. For more information about using the ALOM CMT software, refer to the *Advanced Lights Out Management (ALOM) CMT v1.3 Guide*.



---

**Caution** – The ALOM CMT documentation refers to only one domain, so you must be aware that the Logical Domains Manager is introducing multiple domains. If a logical domain is restarted, I/O services for guest domains might be unavailable until the control domain has restarted. This is because the control domain functions as a service domain in the Logical Domains Manager 1.0.3 software. Guest domains appear to freeze during the reboot process. Once the control domain has fully restarted, the guest domains resume normal operations. It is only necessary to shut down guest domains when power is going to be removed from the entire server.

---

An additional option is available to the existing ALOM CMT command.

```
bootmode [normal|reset_nvram|bootscript=strong|config="config-name"]
```

The `config="config-name"` option enables you to set the configuration on the next power on to another configuration, including the factory-default shipping configuration.

You can invoke the command whether the host is powered on or off. It takes effect on the next host reset or power on.

## ▼ To Reset the Logical Domain Configuration to the Default or Another Configuration

- **Reset the logical domain configuration on the next power on to the default shipping configuration by executing this command in ALOM CMT software.**

```
sc> bootmode config="factory-default"
```

You also can select other configurations that have been created with the Logical Domains Manager using the `ldm add-config` command and stored on the system controller (SC). The name you specify in the Logical Domains Manager

`ldm add-config` command can be used to select that configuration with the ALOM CMT `bootmode` command. For example, assume you stored the configuration with the name `ldm-config1`.

```
sc> bootmode config="ldm-config1"
```

Refer to the `ldm(1M)` man page or the *Logical Domains (LDDoms) Manager 1.0.3 Man Page Guide* for more information about the `ldm add-config` command.

---

## Enabling and Using BSM Auditing

The Logical Domains Manager uses the Solaris OS Basic Security module (BSM) auditing capability. BSM auditing provides the means to examine the history of actions and events on your control domain to determine what happened. The history is kept in a log of what was done, when it was done, by whom, and what was affected.

If you want to use this auditing capability, this section describes how to enable, verify, disable, print output, and rotate audit logs. You can find further information about BSM auditing in the *Solaris 10 System Administration Guide: Security Services*.

You can enable BSM auditing in one of two ways. When you want to disable auditing, be sure you use the same method that you used in enabling. The two methods are:

- Use the `enable-bsm.fin` finish script in the Solaris Security Toolkit.

The `enable-bsm.fin` script is not used by default by the `ldm_control-secure.driver`. You must enable the finish script in your chosen driver.

- Use the Solaris OS `bsmconv(1M)` command.

Here are the procedures for both methods.

### ▼ To Use the `enable-bsm.fin` Finish Script

1. **Copy the `ldm_control-secure.driver` to `my-ldm.driver`, where `my-ldm.driver` is the name for your copy of the `ldm_control-secure.driver`.**
2. **Copy the `ldm_control-config.driver` to `my-ldm-config.driver`, where `my-ldm-config.driver` is the name for your copy of the `ldm_control-config.driver`.**



3. Copy the `ldm_control-hardening.driver` to *my-ldm-hardening.driver*, where *my-ldm-hardening.driver* is the name for your copy of the `ldm_control-hardening.driver`.
4. Edit *my-ldm.driver* to refer to the new configuration and hardening drivers, *my-ldm-control.driver* and *my-ldm-hardening.driver*, respectively.
5. Edit *my-ldm-hardening.driver*, and remove the pound sign (#) from in front of the following line in the driver.

```
enable-bsm.fin
```

6. Execute *my-ldm.driver*.

```
# /opt/SUNWjass/bin/jass-execute -d my-ldm.driver
```

7. Reboot the Solaris OS for auditing to take effect.

## ▼ To Use the Solaris OS `bsmconv(1M)` Command

1. Add `vs` in the flags: line of the `/etc/security/audit_control` file.
2. Run the `bsmconv(1M)` command.

```
# /etc/security/bsmconv
```

For more information about this command, refer to the Solaris 10 Reference Manual Collection or the man page.

3. Reboot the Solaris Operating System for auditing to take effect.

## ▼ To Verify that BSM Auditing is Enabled

1. Type the following command.

```
# auditconfig -getcond
```

2. Check that `audit condition = auditing` appears in the output.

## ▼ To Disable Auditing

You can disable auditing in one of two ways, depending on how you enabled it. See [“Enabling and Using BSM Auditing” on page 114](#).

### 1. Do one of the following.

- Undo the Solaris Security Toolkit hardening run which enabled BSM auditing.

```
# /opt/SUNWjass/bin/jass-execute -u
```

- Use the Solaris OS `bsmunconv(1M)` command.

```
# /etc/security/bsmunconv
```

### 2. Reboot the Solaris OS for the disabling of auditing to take effect.

## ▼ To Print Audit Output

### ● Use one of the following to print BSM audit output.

- Use the Solaris OS commands `auditreduce(1M)` and `praudit(1M)` to print audit output. For example:

```
# auditreduce -c vs | praudit
# auditreduce -c vs -a 20060502000000 | praudit
```

- Use the Solaris OS `praudit -x` command to print XML output.

## ▼ To Rotate Audit Logs

- Use the Solaris OS `audit -n` command to rotate audit logs.

---

## Supported Network Adapters

In a logical domains environment, the virtual switch service running in a service domain can directly interact with GLDv3-compliant network adapters. Though non-GLDv3 compliant network adapters can be used in these systems, the virtual

switch cannot interface with them directly. See [“Configuring Virtual Switch and Service Domain for NAT and Routing”](#) on page 117 for information about how to use non-GLDv3 compliant network adapters.

## ▼ To Determine If a Network Adapter Is GLDv3-Compliant

1. Use the Solaris OS `dladm(1M)` command, where, for example, `bge0` is the network device name.

```
# dladm show-link bge0
bge0                type: non-vlan    mtu: 1500        device: bge0
```

2. Look at `type`: in the output:

- GLDv3-compliant drivers will have a type of `non-vlan` or `vlan`.
- Non-GLDv3-compliant drivers will have a type of `legacy`.

---

## Configuring Virtual Switch and Service Domain for NAT and Routing

The virtual switch (`vswitch`) is a layer-2 switch, that also can be used as a network device in the service domain. The virtual switch can be configured to act only as a switch between the virtual network (`vnet`) devices in the various logical domains but with no connectivity to a network outside the box through a physical device. In this mode, plumbing the `vswitch` as a network device and enabling IP routing in the service domain enables virtual networks to communicate outside the box using the service domain as a router. This mode of operation is very essential to provide external connectivity to the domains when the physical network adapter is not GLDv3-compliant.

The advantages of this configuration are:

- The virtual switch does not need to use a physical device directly and can provide external connectivity even when the underlying device is not GLDv3-compliant.
- The configuration can take advantage of the IP routing and filtering capabilities of the Solaris OS.

## ▼ To Set Up the Virtual Switch to Provide External Connectivity to Domains

1. **Create a virtual switch with no associated physical device.**

If assigning an address, ensure that the virtual switch has an unique MAC address.

```
primary# ldm add-vsw [mac-addr=xxxxxxxxxxxx] primary-vsw0 primary
```

2. **Plumb the virtual switch as a network device in addition to the physical network device being used by the domain.**

See [“To Configure the Virtual Switch as the Primary Interface”](#) on page 45 for more information about plumbing the virtual switch.

3. **Configure the virtual switch device for DHCP, if needed.**

See [“To Configure the Virtual Switch as the Primary Interface”](#) on page 45 for more information about configuring the virtual switch device for DHCP.

4. **Create the `/etc/dhcp.vsw` file, if needed.**

5. **Configure IP routing in the service domain, and set up required routing tables in all the domains.**

For information about how to do this, refer to the section on “Packet Forwarding and Routing on IPv4 Networks” in Chapter 5 “Configuring TCP/IP Network Services and IPv4 Administration” in the *System Administration Guide: IP Services* in the Solaris Express System Administrator Collection.

---

## Configuring IPMP in a Logical Domains Environment

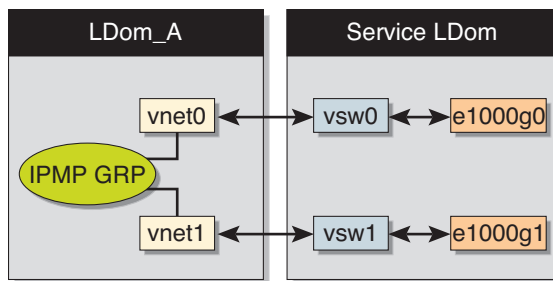
Internet Protocol Network Multipathing (IPMP) provides fault-tolerance and load balancing across multiple network interface cards. By using IPMP, you can configure one or more interfaces into an IP multipathing group. After configuring IPMP, the system automatically monitors the interfaces in the IPMP group for failure. If an interface in the group fails or is removed for maintenance, IPMP automatically migrates, or fails over, the failed interface’s IP addresses. In a Logical Domains environment, either the physical or virtual network interfaces can be configured for failover using IPMP.

# Configuring Virtual Network Devices into an IPMP Group in a Logical Domain

A logical domain can be configured for fault-tolerance by configuring its virtual network devices to an IPMP group. When setting up an IPMP group with virtual network devices, in a active-standby configuration, set up the group to use probe-based detection. Link-based detection and failover currently are not supported for virtual network devices in Logical Domains 1.0.3 software.

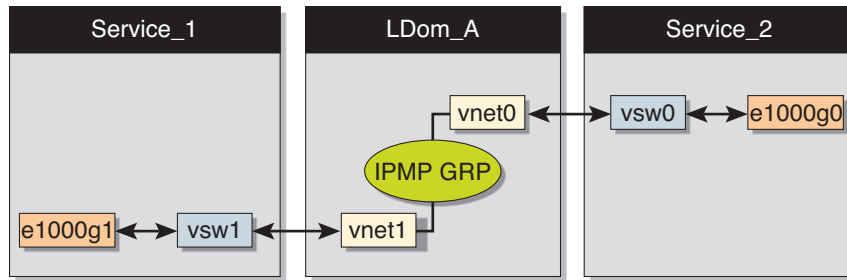
The following diagram shows two virtual networks (`vnet1` and `vnet2`) connected to separate virtual switch instances (`vsw0` and `vsw1`) in the service domain, which, in turn, use two different physical interfaces (`e1000g0` and `e1000g1`). In the event of a physical interface failure, the IP layer in `LDom_A` detects failure and loss of connectivity on the corresponding `vnet` through probe-based detection, and automatically fails over to the secondary `vnet` device.

**FIGURE 6-1** Two Virtual Networks Connected to Separate Virtual Switch Instances



Further reliability can be achieved in the logical domain by connecting each virtual network device (`vnet0` and `vnet1`) to virtual switch instances in different service domains (as shown in the following diagram). Two service domains (`Service_1` and `Service_2`) with virtual switch instances (`vsw1` and `vsw2`) can be set up using a split-PCI configuration. In this case, in addition to network hardware failure, `LDom_A` can detect virtual network failure and trigger a failover following a service domain crash or shutdown.

**FIGURE 6-2** Each Virtual Network Device Connected to Different Service Domains



Refer to the *Solaris 10 System Administration Guide: IP Services* for more information about how to configure and use IPMP groups.

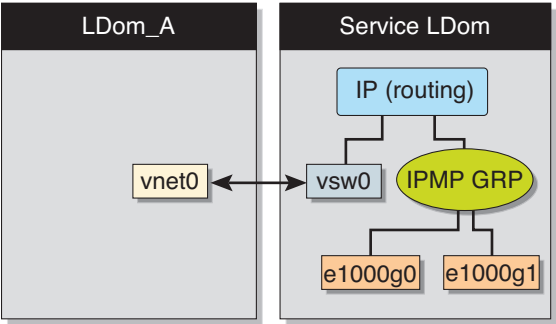
## Configuring and Using IPMP in the Service Domain

Network failure detection and recovery can also be set up in a Logical Domains environment by configuring the physical interfaces in the service domain into a IPMP group. To do this, configure the virtual switch in the service domain as a network device, and configure the service domain itself to act as an IP router. (Refer to the *Solaris 10 System Administration Guide: IP Services* for information on setting up IP routing).

Once configured, the virtual switch sends all packets originating from virtual networks (and destined for an external machine), to its IP layer, instead of sending the packets directly via the physical device. In the event of a physical interface failure, the IP layer detects failure and automatically re-routes packets through the secondary interface.

Since the physical interfaces are directly being configured into a IPMP group, the group can be set up for either link-based or probe-based detection. The following diagram shows two network interfaces (e1000g0 and e1000g1) configured as part of an IPMP group. The virtual switch instance (vsw0) has been plumbed as a network device to send packets to its IP layer.

**FIGURE 6-3** Two Network Interfaces Configured as Part of IPMP Group







# Glossary

---

This list defines terminology, abbreviations, and acronyms in the Logical Domains 1.0.3 documentation.

---

## A

<b>ALOM CMT</b>	Advanced Lights Out Manager chip multithreading, which runs on the system controller and allows you to monitor and control your CMT server
<code>auditreduce(1M)</code>	merge and select audit records from audit trail files
<b>auditing</b>	Using the Solaris OS BSM to identify the source of security changes
<b>authorization</b>	Setting up authorization using the Solaris OS RBAC

---

## B

<code>bge</code>	Broadcom Gigabit Ethernet driver on Broadcom BCM57xx devices
<b>BSM</b>	Basic Security module
<code>bsmconv(1M)</code>	enable the BSM
<code>bsmunconv(1M)</code>	disable the BSM

---

## C

<b>CD</b>	compact disc
<b>CLI</b>	command-line interface
<b>compliance</b>	Determining if a system's configuration is in compliance with a predefined security profile
<b>config</b>	Name of logical domain configuration saved on the system controller
<b>CMT</b>	chip multithreading
<b>constraints</b>	To the Logical Domains Manager, constraints are one or more resources you want to have assigned to a particular domain. You either receive all the resources you ask to be added to a domain or you get none of them, depending upon the available resources.
<b>control domain</b>	Domain that creates and manages other logical domains and services
<b>CPU</b>	central processing unit
<b>CWQ</b>	Control Word Queue; cryptographic unit for Sun UltraSPARC T2-based platforms

---

## D

<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>DMP</b>	Dynamic Multipathing (Veritas)
<b>DR</b>	dynamic reconfiguration
<b>drd(1M)</b>	dynamic reconfiguration daemon for Logical Domains Manager (Solaris 10 OS)
<b>DS</b>	Domain Services module (Solaris 10 OS)
<b>DVD</b>	digital versatile disc

---

## E

<b>e1000g</b>	driver for Intel PRO/1000 Gigabit family of network interface controllers
---------------	---

<b>EFI</b>	extensible firmware interface
<b>ETM</b>	Encoding Table Management module (Solaris 10 OS)

---

## F

<b>FC_AL</b>	Fiber Channel Arbitrated Loop
<b>FMA</b>	Fault Management Architecture
<b>fmd(1M)</b>	Fault Manager daemon (Solaris 10 OS)
<b>format(1M)</b>	disk partitioning and maintenance utility
<b>fmthard(1M)</b>	populate label on hard disks
<b>FTP</b>	File Transfer Protocol

---

## G

<b>guest domain</b>	Uses services from the I/O and service domains and is managed by the control domain.
<b>GLDv3</b>	Generic LAN Driver version 3.

---

## H

<b>hardening</b>	Modifying Solaris OS configuration to improve security
<b>HDD</b>	hard disk drive
<b>hypervisor</b>	Firmware layer interposed between the operating system and the hardware layer

---

## I

- io** I/O devices, such as internal disks and PCI-Express (PCI-E) controllers and their attached adapters and devices
- IB** Infiniband
- IDE** integrated development environment
- I/O domain** Domain that has direct ownership of and direct access to physical I/O devices and that shares those devices to other logical domains in the form of virtual devices
- ioctl** input/output control call
- IP** Internet Protocol
- IPMP** Internet Protocol Network Multipathing
- ISO** International Organization for Standardization

---

## K

- kaio** kernel asynchronous input/output
- KB** kilobyte
- KU** kernel update

---

## L

- LAN** local-area network
- LDAP** Lightweight Directory Access Protocol
- LDC** logical domain channel
- ldm(1M)** Logical Domain Manager utility
- ldmd** Logical Domains Manager daemon
- lofi** loopback file

<b>logical domain</b>	Discrete logical grouping with its own operating system, resources, and identity within a single computer system
<b>Logical Domains (LDom) Manager</b>	Provides a CLI to create and manage logical domains and allocate resources to domains
<b>LUN</b>	logical unit number

---

## M

<b>MAC</b>	media access control address, which LDomS can automatically assign or you can assign manually
<b>MAU</b>	Modular Arithmetic Unit; the cryptographic device for Sun UltraSPARC T1-based platforms
<b>MB</b>	megabyte
<b>MD</b>	machine description in the server database
<code>mem, memory</code>	memory unit - default size in bytes, or specify gigabytes (G), kilobytes (K), or megabytes (M). Virtualized memory of the server that can be allocated to guest domains.
<code>metadb(1M)</code>	create and delete replicas of the SVM metadvice state database
<code>metaset(1M)</code>	configure disk sets
<code>mhd(7I)</code>	multihost disk control operations
<b>MMF</b>	Multimode fiber
<b>MIB</b>	Management Information Base
<b>minimizing</b>	Installing the minimum number of core Solaris OS package necessary
<b>MMU</b>	memory management unit
<code>mtu</code>	maximum transmission unit

---

## N

<b>NAT</b>	Network Address Translation
------------	-----------------------------

<b>NDPSS</b>	Netra Data Plane Software Suite
<code>ndpsldcc</code>	Netra Data Plane Software Logical Domain Channel Client. <i>See also</i> <code>vdpccl</code> .
<code>ndpsldcs</code>	Netra Data Plane Software Logical Domain Channel Service. <i>See also</i> <code>vdpcs</code> .
<b>NFS</b>	Network File System
<b>NIS</b>	Network Information Services
<b>NIU</b>	Network Interface Unit (Sun SPARC Enterprise T5120 and T5220 servers)
<b>NTS</b>	network terminal server
<b>NVRAM</b>	non-volatile random-access memory
<code>nxge</code>	Driver for Sun x8 Express 1/10G Ethernet Adapter

---

## O

<b>OS</b>	operating system
-----------	------------------

---

## P

<b>PA</b>	physical address
<b>PCI</b>	peripheral component interconnect bus
<b>PCI-E</b>	PCI Express bus
<b>PCI-X</b>	PCI Extended bus
<b>PICL</b>	Platform Information and Control Library
<code>picld(1M)</code>	PICL daemon
<code>praudit(1M)</code>	print contents of an audit trail file
<b>PRI</b>	priority

---

## R

<b>RA</b>	real address
<b>RAID</b>	Redundant Array of Inexpensive Disks
<b>RBAC</b>	Role-Based Access Control
<b>RPC</b>	Remote Procedure Call

---

## S

<b>SC</b>	system controller, same as system processor
<b>SCSI</b>	Small Computer System Interface
<b>service domain</b>	Logical domain that provides devices, such as virtual switches, virtual console connectors, and virtual disk servers to other logical domains
<b>SMA</b>	System Management Agent
<b>SMF</b>	Service Management Facility of Solaris 10 OS
<b>SNMP</b>	Simple Network Management Protocol
<b>SP</b>	system processor, same as system controller
<b>SSH</b>	Secure Shell
<b>ssh(1)</b>	Secure Shell command
<b>sshd(1M)</b>	Secure Shell daemon
<b>SunVTS</b>	Sun Validation Test Suite
<b>svcadm(1M)</b>	manipulate service instances
<b>SVM</b>	Solaris Volume Manager

---

## T

<b>TCP</b>	Transmission Control Protocol
------------	-------------------------------

---

## U

<b>UDP</b>	User Datagram Protocol
<b>USB</b>	Universal Serial Bus
<b>uscsi(7D)</b>	user SCSI command interface
<b>UTP</b>	unshielded twisted pair

---

## V

<b>vBSC</b>	virtual blade system controller
<b>vcc, vconscon</b>	virtual console concentrator service with a specific port range to assign to the guest domains
<b>vcons, vconsole</b>	virtual console for accessing system level messages. A connection is achieved by connecting to vconscon service in the control domain at a specific port.
<b>vcpu</b>	virtual central processing unit. Each of the cores of a server are represented as virtual CPUs. For example, an 8-core Sun Fire T2000 Server has 32 virtual CPUs that can be allocated between the logical domains.
<b>vdc</b>	virtual disk client
<b>vdppcc</b>	virtual data plane channel client in an NDPS environment
<b>vdppcs</b>	virtual data plane channel service in an NDPS environment
<b>vdisk</b>	virtual disks are generic block devices backed by different types of physical devices, volumes, or files.
<b>vds, vdiskserver</b>	virtual disk server allows you to import virtual disks into a logical domain.
<b>vdsdev, vdiskserverdevice</b>	virtual disk server device is exported by the virtual disk server. The device can be an entire disk, a slice on a disk, a file, or a disk volume.
<b>vnet</b>	virtual network device implements a virtual Ethernet device and communicates with other vnet devices in the system using the virtual network switch (vswitch).
<b>vntsd(1M)</b>	virtual network terminal server daemon for Logical Domains consoles (Solaris 10 OS)



<b>volfs(7FS)</b>	Volume Management file system
<b>vsw, vswitch</b>	virtual network switch that connects the virtual network devices to the external network and also switches packets between them.
<b>VTOC</b>	volume table of contents
<b>VxDMP</b>	Veritas Dynamic Multipathing
<b>VxVM</b>	Veritas Volume Manager

---

## W

<b>WAN</b>	wide-area network
------------	-------------------

---

## X

<b>XFP</b>	eXtreme Fast Path
<b>XML</b>	Extensible Markup Language

---

## Z

<b>ZFS</b>	Zettabyte File System (Solaris 10 OS)
<b>zpool(1M)</b>	ZFS storage pool
<b>ZVOL</b>	ZFS Volume Emulation Driver

