



# Logical Domains (LDoms) 1.0.2 管理ガイド

---

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

Part No. 820-4452-10  
2008 年 3 月, Revision A

コメントの送付: <http://www.sun.com/hwdocs/feedback>

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

米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) は、本書に記述されている技術に関する知的所有権を有していません。これら知的所有権には、<http://www.sun.com/patents> に掲載されているひとつまたは複数の米国特許、および米国ならびにその他の国におけるひとつまたは複数の特許または出願中の特許が含まれています。

本書およびそれに付随する製品は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および本書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品のフォント技術を含む第三者のソフトウェアは、著作権法により保護されており、提供者からライセンスを受けているものです。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

本製品は、株式会社モリサワからライセンス供与されたリュウミン L-KL (Ryumin-Light) および中ゴシック BBB (GothicBBB-Medium) のフォント・データを含んでいます。

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリョービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人日本規格協会文字フォント開発・普及センターからライセンス供与されたタイプフェースマスタをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、フォントとして無断複製することは禁止されています。

Sun, Sun Microsystems, Java, JumpStart, OpenBoot, Sun Fire, Netra, SunSolve, Sun BluePrints, Sun Blade, Sun Ultra, Sun VTS は、米国およびその他の国における米国 Sun Microsystems 社のサービスマーク、商標、もしくは登録商標です。サンロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャーに基づくものです。

Adobe PostScript のロゴは、Adobe Systems, Incorporated の商標です。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

ATOK は、株式会社ジャストシステムの登録商標です。ATOK8 は、株式会社ジャストシステムの著作物であり、ATOK8 にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。ATOK Server/ATOK12 は、株式会社ジャストシステムの著作物であり、ATOK Server/ATOK12 にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun™ Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザーおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザーインターフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

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

このマニュアルに記載されている製品および情報は、米国の輸出規制法に従うものであり、その他の国の輸出または輸入に関する法律が適用される場合もあります。核、ミサイル、化学生物兵器、または核の海上での最終使用あるいは最終使用者は、直接的または間接的にかかわらず厳重に禁止されています。米国の通商禁止対象国、または拒否された人物および特別認定国リストにかぎらず、米国の輸出禁止リストに指定されている実体への輸出または再輸出は、厳重に禁止されています。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本書には、技術的な誤りまたは誤植のある可能性があります。また、本書に記載された情報には、定期的に変更が行われ、かかる変更は本書の最新版に反映されます。さらに、米国サンまたは日本サンは、本書に記載された製品またはプログラムを、予告なく改良または変更することがあります。

本製品が、外国為替および外国貿易管理法(外為法)に定められる戦略物資等(貨物または役務)に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: Logical Domains (LDoms) 1.0.2 Administration Guide

Part No: 820-3598-10

Revision A



Adobe PostScript

# 目次

---

- はじめに xvii
- 1. Logical Domains ソフトウェアの概要 1
  - ハイパーバイザと論理ドメイン 1
  - Logical Domains Manager 3
    - 論理ドメインの役割 4
    - コマンド行インタフェース 4
    - 仮想入出力 5
      - 仮想ネットワーク 5
      - 仮想ストレージ 6
      - 仮想コンソール 6
    - 動的再構成 6
    - 遅延再構成 6
    - 持続的な構成 7
- 2. セキュリティー 9
  - セキュリティ上の考慮事項 9
  - Solaris Security Toolkit および Logical Domains Manager 10
    - 強化 11
    - 論理ドメインの最小化 12

承認	12
監査	13
適合性	14
3. ソフトウェアのインストールおよび有効化	15
Solaris OS のアップグレード	15
Logical Domains の制約データベースファイルの保存および復元	15
制御ドメインでの Live Upgrade の使用	16
LDoms 1.0.2 ソフトウェアへのアップグレード	16
▼ LDoms 1.0 から LDoms 1.0.2 ソフトウェアへアップグレードする	16
制御ドメインへのソフトウェアの新規インストール	18
▼ Solaris 10 OS をインストールする	19
▼ システムファームウェアをアップグレードする	19
▼ FTP サーバーを使用せずに、システムファームウェアをアップグレードする	21
▼ システムファームウェアをダウングレードする	21
Logical Domains Manager および Solaris Security Toolkit のダウンロード	22
▼ Logical Domains Manager、Solaris Security Toolkit、および Logical Domains MIB をダウンロードする	22
Logical Domains Manager および Solaris Security Toolkit のインストール	23
インストールスクリプトを使用した Logical Domains Manager 1.0.2 および Solaris Security Toolkit 4.2 ソフトウェアのインストール	23
▼ オプションを指定せずに install-ldm スクリプトを使用してインストールする	25
▼ -d オプションを指定して install-ldm スクリプトを使用してインストールする	28
▼ -d none オプションを指定して install-ldm スクリプトを使用してインストールする	29
▼ -p オプションを指定して install-ldm スクリプトを使用してインストールする	30
JumpStart を使用した Logical Domains Manager 1.0.2 および Solaris Security Toolkit 4.2 ソフトウェアのインストール	30

- ▼ JumpStart サーバーを設定する 30
- ▼ JumpStart ソフトウェアを使用してインストールする 31
- Logical Domains Manager および Solaris Security Toolkit ソフトウェアの手動インストール 33
  - ▼ Logical Domains Manager (LDoms) 1.0.2 ソフトウェアを手動でインストールする 33
  - ▼ (省略可能) Solaris Security Toolkit 4.2 ソフトウェアを手動でインストールする 34
  - ▼ (省略可能) 制御ドメインを手動で強化する 34
  - ▼ 強化の妥当性検査を行う 35
  - ▼ 強化を取り消す 35
- Logical Domains Manager デーモンの有効化 36
  - ▼ Logical Domains Manager デーモンを有効にする 36
- ユーザーアカウントに対する承認およびプロファイルの作成と役割の割り当て 37
  - ユーザー承認の管理 37
    - ▼ ユーザーの承認を追加する 37
    - ▼ ユーザーのすべての承認を削除する 38
  - ユーザープロファイルの管理 38
    - ▼ ユーザーのプロファイルを追加する 38
    - ▼ ユーザーのすべてのプロファイルを削除する 39
  - ユーザーへの役割の割り当て 39
    - ▼ 役割を作成し、ユーザーにその役割を割り当てる 39
- 4. サービスと論理ドメインの設定 41
  - 出力メッセージ 41
    - Sun UltraSPARC T1 プロセッサ 41
    - Sun UltraSPARC T2 プロセッサ 42
  - デフォルトのサービスの作成 42
    - ▼ デフォルトのサービスを作成する 42

- 制御ドメインの初期構成 44
  - ▼ 制御ドメインを設定する 44
- 論理ドメインを使用するための再起動 46
  - ▼ 論理ドメインを使用するために再起動する 46
- 制御ドメインまたはサービスドメインとその他のドメイン間のネットワークの有効化 46
  - ▼ 仮想スイッチを主インターフェースとして構成する 47
- 仮想ネットワーク端末サーバーデーモンの有効化 48
  - ▼ 仮想ネットワーク端末サーバーデーモンを有効にする 48
- ゲストドメインの作成と起動 49
  - ▼ ゲストドメインを作成して起動する 49
- ゲストドメインの JumpStart 53
- 5. その他の情報とタスク 55
  - CLI で名前を入力する場合の制限 55
    - ファイル名 (*file*) と変数名 (*var\_name*) 55
    - 仮想ディスクサーバー *file|device* および仮想スイッチデバイス名 55
    - 構成名 (*config\_name*) 55
    - その他のすべての名前 56
  - `ldm list` サブコマンドの使用 56
    - マシンが読み取り可能な出力 56
      - ▼ `ldm` サブコマンドの構文の使用法を表示する 56
    - フラグの定義 59
    - 利用統計情報の定義 60
    - さまざまなリストの例 60
      - ▼ ソフトウェアのバージョンを表示する (`-v`) 60
      - ▼ 省略形式のリストを生成する 60
      - ▼ 長形式のリストを生成する (`-l`) 61
      - ▼ 拡張リストを生成する (`-e`) 62

▼	解析可能でマシンが読み取り可能なリストを生成する (-p)	64
▼	ドメインの状態を表示する	64
▼	変数を一覧表示する	65
▼	バインドを一覧表示する	65
▼	構成を一覧表示する	66
▼	デバイスを一覧表示する	66
▼	サービスを一覧表示する	68
	制約の一覧表示	68
▼	1つのドメインの制約を一覧表示する	68
▼	制約をXML形式で一覧表示する	69
▼	制約をマシンが読み取り可能な形式で一覧表示する	70
	ドメインの負荷が大きい場合に <code>ldm stop-domain</code> コマンドがタイムアウトする可能性 がある	71
	仮想ネットワークデバイスに対応する Solaris ネットワークインタフェース名の 判定	72
▼	Solaris OS ネットワークインタフェース名を確認する	72
	自動または手動による MAC アドレスの割り当て	73
	Logical Domains ソフトウェアに割り当てられる MAC アドレスの範囲	74
	自動割り当てのアルゴリズム	74
	重複した MAC アドレスの検出	75
	解放された MAC アドレス	76
	手動による MAC アドレスの割り当て	76
▼	手動で MAC アドレスを割り当てる	76
	CPU およびメモリーアドレスのマッピング	77
	CPU マッピング	77
▼	CPU 番号を確認する	77
	メモリーのマッピング	78
▼	実メモリーアドレスを確認する	78
	CPU およびメモリーのマッピングの例	78

複数の論理ドメインを使用するための分割 PCI Express バスの構成	80
▼ 分割 PCI 構成を作成する	81
PCI バスでの I/O MMU バイパスモードの有効化	83
コンソールグループの使用	84
▼ 複数のコンソールを 1 つのグループにまとめる	84
サーバー間での論理ドメインの移動	85
▼ 移動するドメインを設定する	85
▼ ドメインを移動する	86
論理ドメインの削除	86
▼ すべてのゲスト論理ドメインを削除する	86
論理ドメインを使用した Solaris OS の操作	87
ドメイン化を有効にした場合、Solaris OS の起動後に OpenBoot ファームウェアを使用できない	88
サーバーの電源の再投入	88
▼ 現在の論理ドメイン構成を SC に保存する	88
OpenBoot <code>power-off</code> コマンドの結果	88
Solaris OS のブレークの結果	88
制御ドメインの停止または再起動の結果	89
仮想ディスクで一部の <code>format(1M)</code> コマンドオプションが機能しない	90
LDoms と ALOM CMT の使用	91
▼ 論理ドメインの構成をデフォルトまたは別の構成にリセットする	92
BSM 監査の有効化と使用	92
▼ <code>enable-bsm.fin</code> 終了スクリプトを使用する	93
▼ Solaris OS の <code>bsmconv(1M)</code> コマンドを使用する	93
▼ BSM 監査が有効であることを確認する	94
▼ 監査を無効にする	94
▼ 監査の出力を表示する	94
▼ 監査ログを切り替える	95
NAT およびルーティング用の仮想スイッチおよびサービスドメインの構成	95



▼ ドメインが外部に接続できるように仮想スイッチを設定する	95
仮想ディスクと ZFS の使用	96
ZFS ボリュームでの仮想ディスクの作成	96
▼ ZFS ボリュームで仮想ディスクを作成する	96
仮想ディスクでの ZFS の使用	98
▼ 仮想ディスクで ZFS を使用する	98
起動ディスクとしての ZFS の使用	99
▼ 起動ディスクとして ZFS を使用する	100
論理ドメイン環境でのボリュームマネージャーの使用	101
ボリュームマネージャーでの仮想ディスクの使用	102
SVM での仮想ディスクの使用	103
VxVM のインストール時の仮想ディスクの使用	104
仮想ディスクでのボリュームマネージャーの使用	104
仮想ディスクでの ZFS の使用	105
仮想ディスクでの SVM の使用	105
仮想ディスクでの VxVM の使用	105
論理ドメイン環境での IPMP の構成	105
論理ドメインの IPMP グループへの仮想ネットワークデバイスの構成	106
サービスドメインでの IPMP の構成と使用	107
用語集	109



# 図目次

---

- 図 1-1            2つの論理ドメインをサポートするハイパーバイザ    2
- 図 5-1            個別の仮想スイッチインスタンスに接続された2つの仮想ネットワーク    106
- 図 5-2            異なるサービスドメインに接続された各仮想ネットワークデバイス    107
- 図 5-3            IPMP グループの一部として構成された2つのネットワークインタフェース    108



# 表目次

---

表 1-1	論理ドメインの役割	4
表 2-1	ldm サブコマンドおよびユーザー承認	13
表 5-1	制御 (primary) ドメインの停止または再起動時に予想される動作	89



# コード例

---

コード例 3-1	ダウンロードした Logical Domains 1.0.2 ソフトウェアのディレクトリ構造	22
コード例 3-2	LDoms 用に強化された Solaris 構成の場合の出力	25
コード例 3-3	カスタマイズされた構成プロファイルを選択した場合の出力	26
コード例 3-4	install-ldm -d スクリプトが正常に実行された場合の出力	28
コード例 3-5	install-ldm -d none スクリプトが正常に実行された場合の出力	29
コード例 5-1	ldm のすべてのサブコマンドの構文の使用法	56
コード例 5-2	インストールされているソフトウェアのバージョン	60
コード例 5-3	すべてのドメインの省略形式のリスト	60
コード例 5-4	すべてのドメインの長形式のリスト	61
コード例 5-5	すべてのドメインの拡張リスト	62
コード例 5-6	マシンが読み取り可能なリスト	64
コード例 5-7	ドメインの状態	64
コード例 5-8	ドメインの変数のリスト	65
コード例 5-9	ドメインのバインドのリスト	65
コード例 5-10	構成のリスト	66
コード例 5-11	すべてのサーバーリソースのリスト	66
コード例 5-12	サービスのリスト	68
コード例 5-13	1 つのドメインの制約のリスト	68
コード例 5-14	ドメインの XML 形式の制約	69
コード例 5-15	マシンが読み取り可能な形式のすべてのドメインの制約	70





# はじめに

---

『Logical Domains (LDoms) 1.0.2 管理ガイド』では、サポートされるサーバー、ブレード、およびサーバーモジュールでの Logical Domains Manager 1.0.2 ソフトウェアの概要、セキュリティー上の考慮事項、インストール、構成、変更、および一般的なタスクの実行に関する詳細な情報や手順について説明します。一覧については、『Logical Domains (LDoms) 1.0.2 リリースノート』の「サポートされるプラットフォーム」を参照してください。このマニュアルは、UNIX® システムおよび Solaris™ オペレーティングシステム (Solaris OS) の実践的な知識がある、これらのサーバーのシステム管理者を対象としています。

---

## お読みになる前に

UNIX のコマンドや手順および Solaris オペレーティングシステムの実践的な知識がない場合は、使用しているシステムハードウェアに付属の Solaris OS ユーザーおよびシステム管理者用のマニュアルを読んで、UNIX システム管理のトレーニング受講を検討してください。

---

## マニュアルの構成

**第 1 章**では、Logical Domains ソフトウェアの概要について説明します。

**第 2 章**では、Solaris Security Toolkit と、Solaris Security Toolkit を使用して論理ドメインで Solaris OS のセキュリティーを実現する方法について説明します。

**第 3 章**では、Logical Domains Manager ソフトウェアをアップグレードまたはインストールして、有効にするための詳細な手順について説明します。

第 4 章では、サービスおよび論理ドメインを設定するための詳細な手順について説明します。

第 5 章では、論理ドメインを管理するために **Logical Domains** ソフトウェアを使用する際の、一般的なタスクの実行に関するその他の情報および手順について説明します。

**用語集**は、LDoms 固有の略語、頭字語、用語、およびそれらの定義についての一覧です。

---

## UNIX コマンド

このマニュアルには、システムの停止、システムの起動、およびデバイスの構成などに使用する基本的な UNIX コマンドと操作手順に関する説明は含まれていない可能性があります。これらについては、以下を参照してください。

- 使用しているシステムに付属のソフトウェアマニュアル
- 下記にある Solaris オペレーティングシステムのマニュアル

<http://docs.sun.com>

---

# シェルプロンプトについて

シェル	プロンプト
UNIX の C シェル	<i>machine-name%</i>
UNIX の Bourne シェルと Korn シェル	\$
スーパーユーザー (シェルの種類を問わない)	#

---

# 書体と記号について

書体または記号*	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例。	.login ファイルを編集します。 ls -a を実行します。 % You have mail.
<b>AaBbCc123</b>	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して表します。	% <b>su</b> Password:
<i>AaBbCc123</i>	コマンド行の可変部分。実際の名前や値と置き換えてください。	<i>rm filename</i> と入力します。
『 』	参照する書名を示します。	『Solaris ユーザーマニュアル』
「 」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」を参照。 この操作ができるのは「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	% <b>grep</b> <b>``#define</b> \ <b>XV_VERSION_STRING'</b>

\* 使用しているブラウザにより、これらの設定と異なって表示される場合があります。

## 関連マニュアル

『Logical Domains (LDoms) 1.0.2 管理ガイド』および『Logical Domains (LDoms) 1.0.2 リリースノート』は、次の URL から入手できます。

<http://docs.sun.com>

『Beginners Guide to LDoms: Understanding and Deploying Logical Domains』は、次の Sun BluePrints™ サイトで参照できます。

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

使用しているサーバー、ソフトウェア、または Solaris OS に関連するマニュアルは、次の URL で参照できます。

<http://docs.sun.com>

必要なマニュアルを検索するには、「検索」ボックスに使用しているサーバー、ソフトウェア、または Solaris OS の名前を入力します。

用途	タイトル	Part No.	形式	場所
LDoms のリリースノート	『Logical Domains (LDoms) 1.0.2 リリースノート』	820-4458-10	HTML PDF	オンラ イン
Ldoms の Solaris マニュアルページ	Solaris 10 Reference Manual Collection: <ul style="list-style-type: none"><li>• drd(1M) マニュアルページ</li><li>• vntsd(1M) マニュアルページ</li></ul>	なし	HTML	オンラ イン
LDoms マニュアルページ	ldm(1M) マニュアルページ	なし	SGML	オンラ イン
	『Logical Domains (LDoms) Manager 1.0.1 マニュアルページガイド』	820-3453-10	PDF	オンラ イン
Logical Domains ソフトウェアの基本	『Beginners Guide to LDoms: Understanding and Deploying Logical Domains』	820-0832-20	PDF	オンラ イン
LDoms MIB の管理	『Logical Domains (LDoms) MIB 1.0.1 管理ガイド』	820-3456-10	HTML PDF	オンラ イン
LDoms MIB のリリースノート	『Logical Domains (LDoms) MIB 1.0.1 リリースノート』	820-3462-10	HTML PDF	オンラ イン
Solaris OS のインストール、JumpStart™ の使用、SMF の使用など	Solaris 10 Collection	なし	HTML PDF	オンラ イン

用途	タイトル	Part No.	形式	場所
セキュリティ	『Solaris Security Toolkit 4.2 管理マニュアル』	819-3789-10	HTML PDF	オンライン
セキュリティ	『Solaris Security Toolkit 4.2 リファレンスマニュアル』	819-3793-10	HTML PDF	オンライン
セキュリティ	『Solaris Security Toolkit 4.2 ご使用にあたって』	819-3796-10	HTML PDF	オンライン
セキュリティ	『Solaris Security Toolkit 4.2 マニュアルページガイド』	819-3794-10	HTML PDF	オンライン

## マニュアル、サポート、およびトレーニング

Sun のサービス	URL
マニュアル	<a href="http://docs.sun.com">http://docs.sun.com</a>
サポート	<a href="http://jp.sun.com/support">http://jp.sun.com/support</a>
トレーニング	<a href="http://jp.sun.com/training">http://jp.sun.com/training</a>

## Sun 以外の Web サイト

このマニュアルで紹介する Sun 以外の Web サイトが使用可能かどうかについては、Sun は責任を負いません。このようなサイトやリソース上、またはこれらを経由して利用できるコンテンツ、広告、製品、またはその他の資料についても、Sun は保証しておらず、法的責任を負いません。また、このようなサイトやリソース上、またはこれらを経由して利用できるコンテンツ、商品、サービスの使用や、それらへの依存に関連して発生した実際の損害や損失、またはその申し立てについても、Sun は一切の責任を負いません。

---

## コメントをお寄せください

マニュアルの品質改善のため、お客様からのご意見およびご要望をお待ちしております。コメントは下記よりお送りください。

<http://www.sun.com/hwdocs/feedback>

ご意見をお寄せいただく際には、下記のタイトルと Part No. を記載してください。

『Logical Domains (LDoms) 1.0.2 管理ガイド』、Part No. 820-4452-10。

# 第1章

## Logical Domains ソフトウェアの概要

---

この章では、Logical Domains ソフトウェアの概要について説明します。Sun の Logical Domains テクノロジを使用するために必要な Solaris OS のすべての機能は、Solaris 10 11/06 release 以上と必須パッチを追加することで使用できるようになります。しかし、論理ドメインを使用するには、システムファームウェアおよび Logical Domains Manager も必要です。詳細は、『Logical Domains (LDoms) 1.0.2 リリースノート』の「必須および推奨されるソフトウェア」を参照してください。

---

## ハイパーバイザと論理ドメイン

この節では、SPARC® ハイパーバイザと、SPARC ハイパーバイザがサポートする論理ドメインの概要について説明します。

SPARC ハイパーバイザは、小さなファームウェア層で、オペレーティングシステムを記述できる安定した仮想化マシンアーキテクチャーを提供します。ハイパーバイザを使用する Sun サーバーでは、論理オペレーティングシステムの活動をハイパーバイザが制御できるようにするためのハードウェア機能が用意されています。

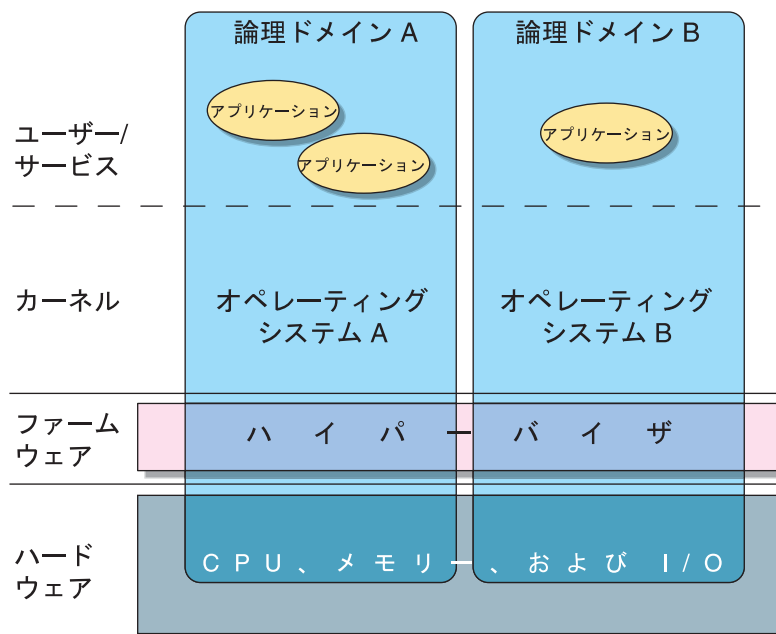
論理ドメインは、独自のオペレーティングシステム、リソース、および単一のコンピュータシステム内での識別情報を持つ個別の論理グループです。各論理ドメインは独立して作成、削除、再構成、および再起動ことができ、そのときサーバーの電源の再投入は必要ありません。パフォーマンスおよびセキュリティー上の理由から、さまざまなアプリケーションソフトウェアを異なる論理ドメイン上で動作させて、アプリケーションの独立性を維持することができます。

各論理ドメインは、ハイパーバイザがそのドメインに対して利用可能にしたサーバーリソースに対してのみ、監視および対話が許可されています。システム管理者は、Logical Domains Manager を使用して、ハイパーバイザが制御ドメインを介して実行する処理を指定します。つまり、ハイパーバイザは、サーバーのリソースをパーティションに分割し、限定的なサブセットを複数のオペレーティングシステム環境に提供

します。これは、論理ドメインを作成する場合の基本的なメカニズムです。次の図に、2つの論理ドメインをサポートするハイパーバイザを示します。また、Logical Domains の機能を構成する次の層についても示します。

- アプリケーションまたはユーザー/サービス
- カーネルまたはオペレーティングシステム
- ファームウェアまたはハイパーバイザ
- ハードウェア (CPU、メモリー、I/O など)

図 1-1 2つの論理ドメインをサポートするハイパーバイザ



特定の SPARC ハイパーバイザがサポートする各論理ドメインの数と機能は、サーバーによって異なります。ハイパーバイザは、サーバー全体の CPU、メモリー、および I/O リソースのサブセットを特定の論理ドメインに割り当てることができます。これにより、それぞれが独自の論理ドメイン内にある複数のオペレーティングシステムを同時にサポートすることができます。リソースは、任意に細分化して個々の論理ドメイン間で再配置できます。たとえば、メモリーは 8K バイトの単位で論理ドメインに割り当てることができます。

各仮想マシンは、次のような独自のリソースを持つ完全に独立したマシンとして管理できます。

- カーネル、パッチ、およびチューニングパラメータ
- ユーザーアカウントおよび管理者
- ディスク



- ネットワークインタフェース、MAC アドレス、および IP アドレス

各仮想マシンは、サーバーの電源の再投入を必要とすることなく、互いに独立して停止、起動、および再起動できます。

ハイパーバイザソフトウェアは、論理ドメイン間の分離を維持する役割を果たします。また、ハイパーバイザソフトウェアは、論理ドメインが相互に通信できるように論理ドメインチャネル (LDC) も提供します。論理ドメインチャネルを使用することで、ドメインはネットワークサービスやディスクサービスなどのサービスを相互に提供できます。

システムコントローラは物理マシンを監視および実行しますが、仮想マシンは管理しません。Logical Domains Manager が仮想マシンを実行します。

---

## Logical Domains Manager

Logical Domains Manager は、論理ドメインを作成および管理するために使用します。Logical Domains Manager は、サーバーごとに 1 つだけ存在できます。Logical Domains Manager は、論理ドメインを物理リソースに割り当てます。

## 論理ドメインの役割

論理ドメインはすべて同じですが、論理ドメインに対して指定する役割だけが異なります。論理ドメインで実行可能ないくつかの役割を、次に示します。

表 1-1 論理ドメインの役割

ドメインの役割	説明
制御ドメイン	Logical Domains Manager を実行するドメイン。ほかの論理ドメインを作成および管理したり、ほかのドメインに仮想リソースを割り当てたりすることができます。制御ドメインは、サーバーごとに 1 つだけ存在できません。Logical Domains ソフトウェアのインストール時に作成される初期ドメインが制御ドメインで、primary という名前になります。
サービスドメイン	仮想スイッチ、仮想コンソール端末集配信装置、仮想ディスクサーバーなどの仮想デバイスサービスをほかのドメインに提供するドメイン。
I/O ドメイン	PCI Express コントローラ内のネットワークカードなどの物理 I/O デバイスに対して、直接の所有権を持ち、直接アクセスできるドメイン。I/O ドメインが制御ドメインを兼ねる場合は、デバイスを仮想デバイスの形式でほかのドメインと共有します。設定できる I/O ドメインの数は、使用しているプラットフォームアーキテクチャーによって異なります。たとえば、Sun UltraSPARC® T1 プロセッサを使用している場合、最大 2 つの I/O ドメインを設定できますが、そのうち 1 つは制御ドメインを兼ねる必要があります。
ゲストドメイン	制御ドメインによって管理され、I/O ドメインおよびサービスドメインのサービスを使用するドメイン。

既存のシステムがあり、オペレーティングシステムおよびその他のソフトウェアがサーバーですでに実行されている場合、Logical Domains Manager をインストールするとそれが制御ドメインになります。制御ドメインの設定後に一部のアプリケーションを制御ドメインから削除したり、システムの使用効率を最大限にするためにアプリケーションの負荷をドメイン間で分散したりすることができます。

## コマンド行インタフェース

Logical Domains Manager では、システム管理者が論理ドメインを作成および構成するためにコマンド行インタフェースが用意されています。CLI には、単一のコマンド `ldm(1M)` と、複数のサブコマンドがあります。

Logical Domains Manager CLI を使用するには、Logical Domains Manager デーモン `ldmd` が実行中である必要があります。`ldm(1M)` コマンドとそのサブコマンドについては、`ldm(1M)` マニュアルページ、および『Logical Domains (LDoms) Manager マニュアルページガイド』で詳しく説明しています。`ldm(1M)` マニュアルページは `SUNWldm` パッケージの一部で、`SUNWldm` パッケージのインストール時にインストールされます。

ldm コマンドを実行するには、使用している UNIX の \$PATH 変数に /opt/SUNWldm/bin ディレクトリが指定されている必要があります。ldm(1M) マニュアルページを参照するには、変数 \$MANPATH にディレクトリパス /opt/SUNWldm/man を追加します。それぞれ次のようになります。

```
$ PATH=$PATH:/opt/SUNWldm/bin; export PATH (Bourne または K シェル用)
$ MANPATH=$MANPATH:/opt/SUNWldm/man; export MANPATH
% set PATH=($PATH /opt/SUNWldm/bin) (C シェル用)
% set MANPATH=($MANPATH /opt/SUNWldm/man)
```

## 仮想入出力

Logical Domains 環境では、管理者は Sun Fire™ または SPARC Enterprise T1000/T2000 サーバー上で、最大 32 のドメインをプロビジョニングすることができます。各ドメインには専用の CPU およびメモリーを割り当てることができますが、それらのシステムにある限られた数の I/O バスや物理 I/O スロットを使用して、ディスクデバイスやネットワークデバイスに対する排他アクセスをすべてのドメインに提供することは不可能です。PCI Express® (PCI-E) バスを 2 つに分割することで一部の物理デバイスは共有できますが (80 ページの「複数の論理ドメインを使用するための分割 PCI Express バスの構成」を参照)、排他的なデバイスアクセスをすべてのドメインに提供するには十分ではありません。このように物理 I/O デバイスへの直接アクセスが不足している状況は、仮想化 I/O モデルを実装することで対処されます。

直接 I/O アクセスを行わないすべての論理ドメインは、サービスドメインと通信する仮想 I/O デバイスを使用して構成されます。サービスドメインは、物理デバイスまたはその機能へのアクセスをエクスポートするサービスを実行します。このようなクライアントサーバーモデルで、仮想 I/O デバイスは、論理ドメインチャンネル (LDC) と呼ばれるドメイン間通信チャンネルを使用して、相互に、またはサービスの対象と通信します。Logical Domains 1.0.2 ソフトウェアの仮想化 I/O 機能には、仮想のネットワーク、ストレージ、およびコンソールのサポートが含まれます。

## 仮想ネットワーク

仮想ネットワークのサポートは、仮想ネットワークおよび仮想ネットワークスイッチデバイスという 2 つのコンポーネントを使用して実装されます。仮想ネットワーク (vnet) デバイスは、Ethernet デバイスをエミュレートし、ポイントツーポイントチャンネルを使用してシステム内のほかの vnet デバイスと通信します。仮想スイッチ (vsw) デバイスは、主に仮想ネットワークのすべての受信パケットおよび送信パケットのマルチプレクサとして機能します。vsw デバイスは、サービスドメインの物理ネットワークアダプタに直接接続し、仮想ネットワークの代わりにパケットを送受信します。vsw デバイスは、単純なレイヤー 2 スイッチとしても機能し、システム内で vsw デバイスに接続された vnet デバイス間でパケットをスイッチします。

## 仮想ストレージ

仮想ストレージインフラストラクチャーを使用することで、論理ドメインは、クライアントサーバーモデルでは論理ドメインに直接割り当てられないブロックレベルのストレージにアクセスできます。これは、ブロック型デバイスインタフェースとしてエクスポートを行う仮想ディスククライアント (vdc) と、仮想ディスククライアントの代わりにディスク要求を処理して、その要求をサービスドメイン上に存在する物理ストレージに送信する仮想ディスクサービス (vds) の 2 つのコンポーネントで構成されます。クライアントドメインでは仮想ディスクは通常のディスクとして認識されますが、すべてのディスク操作は仮想ディスクサービスを介して物理ディスクに転送されます。

## 仮想コンソール

Logical Domains 環境では、primary ドメインを除くすべてのドメインからのコンソール I/O は、システムコントローラではなく、仮想コンソール端末集配信装置 (vcc) および仮想ネットワーク端末サーバーを実行しているサービスドメインにリダイレクトされます。仮想コンソール端末集配信装置サービスは、すべてのドメインのコンソールトラフィックの端末集配信装置として機能します。また、仮想ネットワーク端末サーバーデーモン (vntsd) とのインタフェースを提供し、UNIX ソケットを使用して各コンソールへのアクセスをエクスポートします。

## 動的再構成

動的再構成 (DR) は、オペレーティングシステムの動作中にリソースを追加または削除することができる機能です。Solaris 10 OS では、仮想 CPU (vcpu) の追加および削除のみをサポートしています。メモリーおよび入出力の動的再構成は、Solaris 10 OS ではサポートされていません。Logical Domains Manager CLI で動的再構成機能を使用するには、変更するドメインで Logical Domains 動的再構成デーモン drd(1M) が動作している必要があります。

## 遅延再構成

即座に有効になる動的再構成処理とは対照的に、遅延再構成処理は、OS の次回の再起動後、または OS が動作していない場合は論理ドメインの停止および起動後に有効になります。add-vcpu、set-vcpu、および remove-vcpu サブコマンドを除く、アクティブな論理ドメインでの追加または削除処理は、遅延再構成処理とみなされます。また、アクティブな論理ドメインでの set-vswitch サブコマンドも遅延再構成処理とみなされます。

Sun UltraSPARC T1 プロセッサを使用している場合で、Logical Domains Manager が先にインストールされて有効になっているとき、または構成が factory-default に復元されているときは、LDoms Manager は構成モードで動作します。このモードでは、再構成要求は受け入れられてキューに入れられますが、処理されません。これにより、実行中のマシンの状態には影響を与えずに新しい構成が生成されて SC に格納されるため、結果として I/O ドメインの遅延再構成や再起動のような制限によって妨げられることがなくなります。

特定の論理ドメインで遅延再構成が処理中になると、その論理ドメインが再起動するまで、または停止して起動するまで、その論理ドメインに対するその他の再構成要求も延期されます。また、ある論理ドメインに対して未処理の遅延再構成がある場合、その他の論理ドメインに対する再構成要求は厳しく制限され、適切なエラーメッセージを表示して失敗します。

アクティブな論理ドメインでの仮想 I/O デバイスの削除の試みは遅延再構成処理として扱われますが、一部の構成変更は即座に発生します。つまり、関連する Logical Domains Manager CLI 操作が呼び出されるとすぐ、デバイスは実際に機能を停止します。

Logical Domains Manager のサブコマンド `remove-reconf` は、遅延再構成処理を取り消します。遅延再構成処理は、`ldm list-domain` コマンドを使用して一覧表示することができます。遅延再構成機能の使用法の詳細は、`ldm(1M)` マニュアルページまたは『Logical Domains (LDoms) Manager マニュアルページガイド』を参照してください。

---

注 – その他の `ldm remove-*` コマンドが仮想 I/O デバイスで実行されている場合は、`ldm remove-reconf` コマンドを使用できません。このような状況では、`ldm remove-reconf` コマンドは失敗します。

---

## 持続的な構成

Logical Domains Manager CLI コマンドを使用して、論理ドメインの現在の構成をシステムコントローラ (SC) に格納することができます。構成の追加、使用する構成の指定、構成の削除、およびシステムコントローラ上の構成の表示を行うことができます。`ldm(1M)` マニュアルページまたは『Logical Domains (LDoms) Manager マニュアルページガイド』を参照してください。さらに、起動する構成を選択できる `ALOM CMT Version 1.3` コマンドもあります ([91 ページの「LDoms と ALOM CMT の使用」](#)を参照)。



## 第2章

# セキュリティー

---

この章では、Solaris Security Toolkit ソフトウェアの概要と、このソフトウェアを使用して論理ドメインで Solaris OS をセキュリティー保護する方法について説明します。

---

## セキュリティー上の考慮事項

Solaris Security Toolkit ソフトウェアは、非公式には JumpStart™ Architecture and Security Scripts (JASS) ツールキットとも呼ばれ、セキュリティー保護された Solaris OS システムの構築および維持を行うために、自動化され、拡張性の高いスケーラブルな機構を提供します。Solaris Security Toolkit は、Logical Domains Manager の制御ドメインを含む、サーバーの管理には不可欠なデバイスのセキュリティー保護を実現します。

Solaris Security Toolkit 4.2 ソフトウェアパッケージ SUNWjass では、install-ldm スクリプトを次のように使用することで、制御ドメイン上の Solaris オペレーティングシステムをセキュリティー保護する手段を提供します。

- Logical Domains Manager インストールスクリプト (install-ldm) および Logical Domains Manager に固有の制御ドライバ (ldm\_control-secure.driver) を使用することにより、Solaris Security Toolkit が制御ドメインを自動的に強化します。
- インストールスクリプトの使用時に代替ドライバを選択します。
- インストールスクリプトの使用時にドライバを選択せず、独自の Solaris 強化を適用します。

SUNWjass パッケージは、Logical Domains (LDoms) Manager 1.0.2 ソフトウェアパッケージ SUNWldm と一緒に同梱されており、Sun のソフトウェアダウンロード Web サイトから入手できます。Logical Domains Manager 1.0.2 ソフトウェアをダウンロードしてインストールすると同時に、Solaris Security Toolkit 4.2 ソフトウェアパッケージをダウンロードしてインストールするオプションがあります。Solaris

Security Toolkit 4.2 ソフトウェアパッケージには、Solaris Security Toolkit ソフトウェアを Logical Domains Manager とともに使用できるようにするための必須パッチが含まれています。ソフトウェアがインストールされたら、Solaris Security Toolkit 4.2 ソフトウェアを使用してシステムを強化できます。第 3 章では、Solaris Security Toolkit をインストールおよび構成し、制御ドメインを強化する方法について説明しています。

Solaris Security Toolkit によって提供されるセキュリティー機能のうち、Logical Domains Manager のユーザーが使用可能な機能を次に示します。

- **強化** – 必須パッチが適用された Solaris Security Toolkit 4.2 ソフトウェアを使用して、Solaris Security Toolkit ソフトウェアを Logical Domains Manager とともに使用できるようにして、システムのセキュリティーを向上するように Solaris OS 構成を変更します。
- **最小化** – LDoms および LDoms Management Information Base (MIB) のサポートに必要な最小限の主要な Solaris OS パッケージをインストールします。
- **承認** – Logical Domains Manager 用に変更された Solaris OS の役割に基づくアクセス制御 (RBAC) を使用して承認を設定します。
- **監査** – Logical Domains Manager 用に変更された Solaris OS 基本セキュリティーモジュール (BSM) を使用してシステムのセキュリティーの変更元を識別し、何が、いつ、誰によって行われ、どのような影響があるのかを判断します。
- **適合性** – Solaris Security Toolkit の監査機能を使用して、システムの構成が事前に定義されたセキュリティープロファイルに適合しているかどうかを判断します。

---

## Solaris Security Toolkit および Logical Domains Manager

第 3 章では、Solaris Security Toolkit をインストールして Logical Domains Manager とともに使用する方法について説明しています。Solaris Security Toolkit は、制御ドメインにインストールします。制御ドメインでは Logical Domains Manager が実行されています。また、Solaris Security Toolkit は、ほかの論理ドメインにインストールすることもできます。唯一の違いは、制御ドメインを強化する場合は `ldm_control-secure.driver` ドライバを使用し、ほかの論理ドメインを強化する場合は `secure.driver` などの別のドライバを使用することです。これは、`ldm_control-secure.driver` が制御ドメイン専用であるためです。`ldm_control-secure.driver` は、`secure.driver` をベースにして、Logical Domains Manager で使用できるようにカスタマイズおよびテストしたものです。`secure.driver` の詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。



## 強化

Solaris Security Toolkit が制御ドメイン上の Solaris OS を強化するために使用するドライバ (`ldm_control-secure.driver`) は、Logical Domains Manager が OS で実行できるように特別な変更を加えたものです。`ldm_control-secure.driver` は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』で説明している `secure.driver` に類似しています。

`ldm_control-secure.driver` は、Logical Domains Manager ソフトウェアを実行しているシステムの制御ドメインに対する基準構成を提供します。これは、Solaris OS ドメインで通常よりも少ないシステムサービスを提供することで、通常の用途ではなく Logical Domains Manager の処理のために制御ドメインを確保することを目的としています。

`install-ldm` スクリプトにより、Logical Domains Manager ソフトウェアがまだインストールされていない場合にはこのソフトウェアがインストールされ、使用可能になります。

`secure.driver` から変更された、その他の重要な事項の概要を次に示します。

- Telnet サーバーは実行できません。代わりに Secure Shell (`ssh`) を使用できます。また、Telnet クライアントを使用して、Logical Domains 仮想ネットワーク端末サーバーデーモン (`vntsd`) によって開始された仮想コンソールにアクセスすることはできます。たとえば、ローカルシステムの TCP ポート 5001 で待機している仮想コンソールが実行中の場合は、次のようにアクセスすることができます。

```
# telnet localhost 5001
```

`vntsd` を有効にする方法については、[36 ページの「Logical Domains Manager デーモンの有効化」](#)を参照してください。これは自動的に有効になりません。

- 次の終了スクリプトが追加されています。これらを使用して、Logical Domains Manager をインストールおよび起動できます。追加されたスクリプトの一部は、カスタマイズしたすべてのドライバに追加する必要がありますが、省略可能なものもあります。スクリプトには、必須であるか任意であるかが示されています。
  - `install-ldm.fin` - `SUNWldm` パッケージをインストールします。(必須)
  - `enable-ldmd.fin` - Logical Domains Manager デーモン (`ldmd`) を有効にします。(必須)
  - `enable-ssh-root-login.fin` - スーパーユーザーが Secure Shell (`ssh`) を使用して直接ログインできるようにします。(任意)
- 次のファイルが変更されました。これらの変更のカスタマイズしたドライバへの組み込みは省略可能であるため、任意として示されています。
  - `/etc/ssh/sshd_config-root` アカountのアクセスがネットワーク全体で許可されます。このファイルはどのドライバでも使用されません。(任意)
  - `/etc/ipf/ipf.conf` - UDP ポート 161 (SNMP) が開かれます。(任意)

- /etc/host.allow - Secure Shell デーモン (sshd) がローカルサブネットだけでなくネットワーク全体に対してオープンになります。(任意)
- 次の終了スクリプトが無効 (コメントアウト) になっています。カスタマイズしたすべてのドライバで、disable-rpc.fin スクリプトをコメントにしてください。その他の変更は省略可能です。スクリプトには、必須であるか任意であるかが示されています。
  - enable-ipfilter.fin - IP フィルタ (ネットワークパケットフィルタの一種) が有効ではありません。(任意)
  - disable-rpc.fin - 遠隔手続き呼び出し (RPC) サービスが有効のままです。RPC サービスは、ネットワーク情報サービス (NIS)、ネットワークファイルシステム (NFS) などのほかの多くのシステムサービスによって使用されます。(必須)
  - disable-sma.fin - システム管理エージェント (NET-SNMP) が有効のままです。(任意)
  - disable-ssh-root-login.fin - ssh root ログインを無効にできません。
  - set-term-type.fin - 不要な旧バージョンのスクリプト。(任意)

## 論理ドメインの最小化

Solaris OS は、要件に合わせてさまざまな数のパッケージを組み合わせて構成できます。最小化では、このようなパッケージのセットを、必要なアプリケーションを実行するために最低限必要な数にまで減らします。最小化により、セキュリティに脆弱性があるおそれのあるソフトウェアの数が減り、インストールされたソフトウェアにパッチを正しく適用し続けることに付随する労力の度合いも軽減されるため、最小化は重要です。論理ドメインの最小化処理では、任意のドメインを完全にサポートし続ける、最小化された Solaris OS をインストールするための JumpStart™ サポートが提供されています。

Solaris Security Toolkit では、LDoms の論理ドメインを最小化するために使用する JumpStart プロファイル minimal-ldm\_control.profile が用意されています。このプロファイルにより、LDoms および LDoms MIB のサポートに必要なすべての Solaris OS パッケージがインストールされます。LDoms MIB を制御ドメインで使用する場合は、LDoms および Solaris Security Toolkit パッケージのインストール後に、個別にそのパッケージを追加する必要があります。これは、ほかのソフトウェアとともに自動的にインストールされません。LDoms MIB のインストールおよび使用法の詳細は、『Logical Domains (LDoms) MIB 1.0.1 管理ガイド』を参照してください。

## 承認

Logical Domains Manager の承認には、次の 2 つのレベルがあります。

- 読み取り — 構成を表示できますが、変更できません。
- 読み取りおよび書き込み — 構成を表示および変更できます。

変更は、Solaris OS に加えられるのではなく、Logical Domains Manager のインストール時にパッケージスクリプト `postinstall` を使用することで、承認ファイルに追加されます。同様に、承認エントリは、パッケージスクリプト `preremove` によって削除されます。

ldm サブコマンドと、そのコマンドの実行に必要な対応するユーザー承認を次の表に示します。

表 2-1 ldm サブコマンドおよびユーザー承認

ldm サブコマンド*	ユーザー承認
<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>

\* 追加、表示、削除、または設定できるすべてのリソースを指します。

## 監査

Logical Domains Manager CLI コマンドの監査は、Solaris OS 基本セキュリティーモジュール (BSM) 監査によって実行されます。Solaris OS BSM 監査の詳細は、Solaris 10 の『Solaris のシステム管理 (セキュリティーサービス)』を参照してください。

Logical Domains Manager に対する BSM 監査は、デフォルトでは有効ではありませんが、インフラストラクチャーは用意されています。BSM 監査は、次の 2 つのいずれかの方法で使用できます。

- Solaris Security Toolkit の `enable-bsm.fin` 終了スクリプトを実行します。
- Solaris OS の `bsmconv(1M)` コマンドを使用します。

Logical Domains Manager で BSM 監査を使用する場合の有効化、検証、無効化、出力の表示、およびログの切り替えの詳細は、[92 ページの「BSM 監査の有効化と使用」](#)を参照してください。

## 適合性

Solaris Security Toolkit には、独自の監査機能があります。Solaris Security Toolkit ソフトウェアは、事前に定義されたセキュリティープロファイルと比較して、Solaris OS が動作しているあらゆるシステムのセキュリティー状況を自動的に検証することができます。この適合性機能の詳細は、『Solaris Security Toolkit 4.2 管理マニュアル』の「システムのセキュリティーの監査」を参照してください。

## 第3章

---

# ソフトウェアのインストールおよび有効化

---

この章では、サポートされるサーバー上の制御ドメインに Logical Domains Manager 1.0.2 ソフトウェアおよびその他のソフトウェアをインストールして有効にする方法について説明します。サポートされるサーバーの一覧については、『Logical Domains (LDoms) 1.0.2 リリースノート』の「サポートされるプラットフォーム」を参照してください。

使用しているプラットフォームに応じて、この章の必要な部分を使用できます。新しい Sun UltraSPARC T2 プラットフォームで Logical Domains ソフトウェアを使用する場合は、すべてのソフトウェアがプリインストールされた状態で出荷されているはずです。

---

## Solaris OS のアップグレード

この節では、Logical Domains の制約データベースファイルの保存および復元と、制御ドメインでの Live Upgrade の実行に必要な情報について説明します。

### Logical Domains の制約データベースファイルの保存および復元

制御ドメインでオペレーティングシステムをアップグレードするたびに、`/var/opt/SUNWldm/ldom-db.xml` で参照できる Logical Domains の制約データベースファイルを保存および復元する必要があります。

---

**注** – また、ディスクスワップなど、制御ドメインのファイルデータを破損するその他の操作を行うときは、`/var/opt/SUNWldm/ldom-db.xml` ファイルも保存および復元する必要があります。

---

## 制御ドメインでの Live Upgrade の使用

制御ドメインで Live Upgrade を使用する場合は、`/etc/lu/synclist` ファイルに次の行を追加することを検討してください。

```
/var/opt/SUNWldm/ldom-db.xml OVERWRITE
```

これによって、データベースがアクティブなブート環境から新しいブート環境に自動的にコピーされます。`/etc/lu/synclist` と、ブート環境間でのファイルの同期については、『Solaris 10 8/07 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「ブート環境間でのファイルの同期」を参照してください。

---

## LDoms 1.0.2 ソフトウェアへのアップグレード

既存の LDoms 1.0.1 の設定は、LDoms 1.0.2 ソフトウェアでも機能するため、LDoms 1.0.1 ソフトウェアから LDoms 1.0.2 ソフトウェアへアップグレードする場合には、次の手順を実行する必要はありません。ただし、既存の LDoms 1.0 の設定を LDoms 1.0.2 ソフトウェアで使用する場合は、次の手順を使用する必要があります。

### ▼ LDoms 1.0 から LDoms 1.0.2 ソフトウェアへアップグレードする

既存の LDoms 1.0 の設定は、LDoms 1.0.2 ソフトウェアでは機能しません。次の手順では、`ldm start-domain` コマンドに XML 制約ファイルおよび `-i` オプションを使用して、構成を保存および再構築する方法について説明します。この方法では、実際のバインドは保持されず、それらのバインドを作成するために使用した制約だけが保持されます。つまり、この手順を行うと、ドメインは同じ仮想リソースを持ちますが、同じ物理リソースにバインドされるとはかぎりません。

基本的な処理は、各ドメインの制約情報を XML ファイルに保存することです。アップグレード後に、この XML ファイルを Logical Domains Manager に対して再実行して、必要な設定を再構築できます。この手順は、制御ドメインではなく、ゲストドメインに対して有効です。制御 (primary) ドメインの制約を XML ファイルに保存することはできますが、それを `ldm start-domain -i` コマンドに指定することはできません。

1. 最新バージョンの Solaris OS に更新します。詳細は、[19 ページの「Solaris 10 OS をインストールする」](#)の手順 2 を参照してください。

2. 各ドメインで、ドメインの制約を含む XML ファイルを作成します。

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

3. システムコントローラに格納されている論理ドメイン構成をすべて一覧表示します。

```
# ldm ls-config
```

4. システムコントローラに格納されているそれぞれの論理ドメイン構成を削除します。

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

5. Logical Domains Manager デーモン (ldmd) を無効にします。

```
# svcadm disable ldmd
```

6. Logical Domains Manager パッケージ (SUNWldm) を削除します。

```
# pkgrm SUNWldm
```

7. Solaris Security Toolkit パッケージ (SUNWjass) を使用している場合はこれを削除します。

```
# pkgrm SUNWjass
```

8. システムのファームウェアをフラッシュ更新します。手順全体については、[19 ページの「システムファームウェアをアップグレードする」](#)または [21 ページの「FTP サーバーを使用せずに、システムファームウェアをアップグレードする」](#)を参照してください。

9. LDom 1.0.2 ソフトウェアパッケージをダウンロードします。

Logical Domains Manager、Solaris Security Toolkit、Logical Domains MIB のダウンロードおよびインストールの手順については、[22 ページの「Logical Domains Manager、Solaris Security Toolkit、および Logical Domains MIB をダウンロードする」](#)を参照してください。

10. primary ドメインを手動で再構成します。手順については、[44 ページの「制御ドメインを設定する」](#)を参照してください。

11. 手順 2 で作成した各ゲストドメインの XML ファイルに対して、次のコマンドを実行します。

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

---

## 制御ドメインへのソフトウェアの新規インストール

Logical Domains Manager ソフトウェアのインストール時に最初に作成されるドメインが、制御ドメインになります。この最初のドメインには、`primary` という名前が付けられます。この名前は変更できません。次の主要コンポーネントが制御ドメインにインストールされます。

- Solaris 10 OS。必要に応じて、『Logical Domains (LDoms) 1.0.2 リリースノート』で推奨されるパッチを追加します。19 ページの「[Solaris 10 OS をインストールする](#)」を参照してください。
- 使用している Sun UltraSPARC T1 プラットフォームのシステムファームウェア version 6.5、または使用している Sun UltraSPARC T2 プラットフォームのシステムファームウェア version 7.0。19 ページの「[システムファームウェアをアップグレードする](#)」を参照してください。
- Logical Domains Manager 1.0.2 ソフトウェア。23 ページの「[Logical Domains Manager および Solaris Security Toolkit のインストール](#)」を参照してください。
- (省略可能) Solaris Security Toolkit 4.2 ソフトウェア。23 ページの「[Logical Domains Manager および Solaris Security Toolkit のインストール](#)」を参照してください。
- (省略可能) Logical Domains (LDoms) Management Information Base (MIB) ソフトウェアパッケージ。LDoms MIB のインストールおよび使用法の詳細は、『Logical Domains (LDoms) MIB 1.0.1 管理ガイド』を参照してください。

Solaris OS およびシステムファームウェアは、Logical Domains Manager をインストールする前に、使用しているサーバーにインストールされている必要があります。Solaris OS、システムファームウェア、および Logical Domains Manager をインストールしたあとで、元のドメインが制御ドメインになります。



## ▼ Solaris 10 OS をインストールする

Solaris 10 OS がまだインストールされていない場合はこれをインストールします。このバージョンの Logical Domains ソフトウェアで使用する必要のある Solaris 10 OS を調べるには、『Logical Domains (LDoms) 1.0.2 リリースノート』の「必須および推奨されるソフトウェア」を参照してください。Solaris OS をインストールする詳細な手順については、使用している Solaris 10 OS のインストールマニュアルを参照してください。インストール内容は、使用しているシステムの要件に合わせて調整できます。

---

**注** – 論理ドメインの場合は、ディスク全体、またはブロック型デバイスとしてエクスポートされたファイルのみに Solaris OS をインストールできます。

---

### 1. Solaris 10 OS をインストールします。

最小化は省略可能です。Solaris Security Toolkit には、次に示す、Logical Domains ソフトウェア用の JumpStart 最小化プロファイルが含まれています。

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

### 2. Solaris 10 11/06 OS をインストールしている場合は、必須パッチをインストールします。必須パッチの一覧については、『Logical Domains (LDoms) 1.0.2 リリースノート』の「Solaris 10 11/06 OS の必須パッチ」を参照してください。

---

**注** – ゲストドメインに英語版以外のオペレーティングシステムをインストールする場合、コンソールの端末は、その OS のインストーラが必要とするロケールである必要があります。たとえば、Solaris OS インストーラには EUC ロケールが必要で、Linux インストーラには Unicode ロケールが必要である場合があります。

---

## ▼ システムファームウェアをアップグレードする

使用しているプラットフォームのシステムファームウェアは、SunSolve サイトから入手できます。

<http://sunsolve.sun.com>

サポートされるサーバーに必要なシステムファームウェアについては、『Logical Domains (LDoms) 1.0.2 リリースノート』の「システムファームウェアの必須パッチ」を参照してください。

この手順では、システムコントローラで `flashupdate(1M)` コマンドを使用してシステムファームウェアをアップグレードする方法について説明します。

- ローカル FTP サーバーへアクセスできない場合は、21 ページの「FTP サーバーを使用せずに、システムファームウェアをアップグレードする」を参照してください。
- 制御ドメインからシステムファームウェアを更新する場合は、使用しているシステムファームウェアのリリースノートを参照してください。

サポートされるサーバーのシステムファームウェアのインストールおよび更新については、そのサーバーの管理マニュアルまたはプロダクトノートを参照してください。

1. システムコントローラに接続されたシリアルまたはネットワークのいずれかの管理ポートを使用して、ホストサーバーを停止して電源を切ります。

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

2. 使用しているサーバーに応じて、flashupdate(1M) コマンドを使用してシステムファームウェアをアップグレードします。

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

各表記の意味は次のとおりです。

- *IP-address* は、使用している FTP サーバーの IP アドレスです。
- *path* は、システムファームウェアイメージを入手できる SunSolve<sup>sm</sup> 内の場所または独自のディレクトリです。
- *x\_x\_x* は、システムファームウェアのバージョン番号です。
- *nn* は、このリリースに適用されるビルド番号です。
- *server-name* は、使用しているサーバーの名前です。たとえば、Sun Fire T2000 サーバーの *server-name* は、Sun\_Fire\_T2000 です。

3. システムコントローラをリセットします。

```
sc> resetsc -y
```

4. ホストサーバーの電源を入れて起動します。

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

## ▼ FTP サーバーを使用せずに、システムファームウェアをアップグレードする

システムコントローラにファームウェアをアップロードするためのローカル FTP サーバーにアクセスできない場合は、`sysfwdownload` ユーティリティーを使用できます。このユーティリティーは、システムファームウェアアップグレードパッケージとともに SunSolve サイトで提供されています。

<http://sunsolve.sun.com>

1. Solaris OS 内で次のコマンドを実行します。

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

2. Solaris OS インスタンスを停止します。

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

3. システムコントローラの電源を切り、ファームウェアを更新します。

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

4. システムコントローラをリセットして電源を入れます。

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

## ▼ システムファームウェアをダウングレードする

Logical Domains ソフトウェアで使用するようシステムファームウェアをアップグレードしたあとで、ファームウェアを Logical Domains に対応していない元のファームウェアにダウングレードできます。

- `flashupdate(1M)` コマンドを実行して、Logical Domains に対応していない元のファームウェアへのパスを指定します。

---

# Logical Domains Manager および Solaris Security Toolkit のダウンロード

## ▼ Logical Domains Manager、Solaris Security Toolkit、および Logical Domains MIB をダウンロードする

1. Sun のソフトウェアダウンロードサイトから、Logical Domains Manager パッケージ (SUNWldm)、Solaris Security Toolkit (SUNWjass) とインストールスクリプト (install-ldm)、および Logical Domains Management Information Base パッケージ (SUNWldmib.v) を含む tar ファイル (LDoms\_Manager-1\_0\_2.zip) をダウンロードします。ソフトウェアは、次の Web サイトから入手できます。

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

2. zip ファイルを解凍します。

```
$ unzip LDoms_Manager-1_0_2.zip
```

ダウンロードしたソフトウェアのディレクトリ構造は、次のようになります。

コード例 3-1      ダウンロードした Logical Domains 1.0.2 ソフトウェアのディレクトリ構造

```
LDoms_Manager-1_0_2/  
  Install/  
    install-ldm  
  Legal/  
    Ldoms_1.0.2_Entitlement.txt  
    Ldoms_1.0.2_SLA_Entitlement.txt  
  Product/  
    SUNWjass/  
    SUNWldm.v/  
    SUNWldmib.v  
  README
```

---

# Logical Domains Manager および Solaris Security Toolkit のインストール

Logical Domains Manager および Solaris Security Toolkit ソフトウェアをインストールするには、次の 3 つの方法があります。

- インストールスクリプトを使用してパッケージおよびパッチをインストールします。この方法では、Logical Domains Manager および Solaris Security Toolkit ソフトウェアの両方が自動的にインストールされます。[23 ページの「インストールスクリプトを使用した Logical Domains Manager 1.0.2 および Solaris Security Toolkit 4.2 ソフトウェアのインストール」](#)を参照してください。
- JumpStart を使用してパッケージをインストールします。[30 ページの「JumpStart を使用した Logical Domains Manager 1.0.2 および Solaris Security Toolkit 4.2 ソフトウェアのインストール」](#)を参照してください。
- 各パッケージを手動でインストールします。[33 ページの「Logical Domains Manager および Solaris Security Toolkit ソフトウェアの手動インストール」](#)を参照してください。

---

**注** – LDoms および Solaris Security Toolkit パッケージをインストールしたあとで、LDoms MIB ソフトウェアパッケージを手動でインストールする必要があります。これは、ほかのパッケージとともに自動的にインストールされません。LDoms MIB のインストールおよび使用法の詳細は、『[Logical Domains \(LDoms\) MIB 1.0.1 管理ガイド](#)』を参照してください。

---

## インストールスクリプトを使用した Logical Domains Manager 1.0.2 および Solaris Security Toolkit 4.2 ソフトウェアのインストール

install-ldm インストールスクリプトを使用する場合、スクリプトの実行方法を指定する選択肢がいくつかあります。それぞれの選択肢について、次の手順で説明します。

- オプションを指定せずに install-ldm スクリプトを使用すると、自動的に次の処理を行います。
  - Solaris OS リリースが Solaris OS 10 11/06 であることを確認します。
  - パッケージのサブディレクトリである SUNWldm/ および SUNWjass/ が存在することを確認します。

- 前提条件となる Solaris Logical Domains ドライバパッケージの SUNWldomr および SUNWldomu が存在することを確認します。
- SUNWldm および SUNWjass パッケージがインストールされていないことを確認します。

---

**注** – インストール中に、スクリプトが SUNWjass の以前のバージョンを検出した場合は、これを削除する必要があります。使用している Solaris OS のこれまでの強化を元に戻す必要はありません。

---

- Logical Domains Manager 1.0.2 ソフトウェア (SUNWldm パッケージ) をインストールします。
- 必須パッチを含む Solaris Security Toolkit 4.2 ソフトウェア (SUNWjass パッケージ) をインストールします。
- すべてのパッケージがインストールされていることを確認します。
- Logical Domains Manager デーモン ldmd を有効にします。
- Solaris Security Toolkit の `ldm_control-secure.driver`、または `-secure.driver` で終わるその他のドライバのうち選択したものを使用して、制御ドメインで Solaris OS を強化します。
- オプション `-d` を指定して `install-ldm` スクリプトを使用すると、`-secure.driver` で終わるドライバ以外の Solaris Security Toolkit ドライバを指定できます。このオプションでは、前述の選択肢で示したすべての機能と次の追加オプションを自動的に実行します。
  - Solaris Security Toolkit のカスタマイズしたドライバ (たとえば `server-secure-myname.driver`) を使用して、制御ドメインで Solaris OS を強化します。
- オプション `-d` と `none` を指定して `install-ldm` スクリプトを使用すると、Solaris Security Toolkit を使用して制御ドメインで動作している Solaris OS を強化しないことを指定します。このオプションは、前述の選択肢で示した強化以外のすべての機能を自動的に実行します。Solaris Security Toolkit の使用を省略することはお勧めしません。別の処理を使用して制御ドメインを強化する場合にかぎり、この使用を省略するようにしてください。
- オプション `-p` を指定して `install-ldm` スクリプトを使用すると、Logical Domains Manager デーモン (ldmd) の有効化および Solaris Security Toolkit の実行といったインストール後の処理のみを実行することを指定します。たとえば、SUNWldm および SUNWjass パッケージがサーバーにプリインストールされている場合に、このオプションを使用します。30 ページの「[-p オプションを指定して install-ldm スクリプトを使用してインストールする](#)」を参照してください。

## ▼ オプションを指定せずに install-ldm スクリプトを使用してインストールする

- オプションを指定せずにインストールスクリプトを実行します。

インストールスクリプトは、SUNWldm パッケージの一部で、Install サブディレクトリにあります。

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

- a. 1 つ以上のパッケージがすでにインストールされている場合は、次のメッセージが表示されます。

```
# 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.
```

インストール後の処理のみを実行する場合は、30 ページの「[-p オプションを指定して install-ldm スクリプトを使用してインストールする](#)」に進みます。

- b. 処理が正常に実行されると、次の例のようなメッセージが表示されます。

- コード例 3-2 は、次のデフォルトのセキュリティープロファイルを選択した場合に、install-ldm スクリプトが正常に実行されたことを示しています。

a) Hardened Solaris configuration for LDoms (recommended)

- コード例 3-3 は、次のセキュリティープロファイルを選択した場合に、install-ldm スクリプトが正常に実行されたことを示しています。

c) Your custom-defined Solaris security configuration profile

選択肢として表示されるドライバは、名前が `-secure.driver` で終わるドライバです。名前が `-secure.driver` で終わらない、カスタマイズしたドライバを書き込む場合は、install-ldm `-d` オプションでカスタマイズしたドライバを指定する必要があります。28 ページの「[-d オプションを指定して install-ldm スクリプトを使用してインストールする](#)」を参照してください。

コード例 3-2 LDoms 用に強化された Solaris 構成の場合の出力

```
# 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
```

コード例 3-2 LDomS 用に強化された Solaris 構成の場合の出力 (続き)

```
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
/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.
```

コード例 3-3 カスタマイズされた構成プロファイルを選択した場合の出力

```
# 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
```



コード例 3-3 カスタマイズされた構成プロファイルを選択した場合の出力 (続き)

```
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.
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
```

コード例 3-3 カスタマイズされた構成プロファイルを選択した場合の出力 (続き)

```
/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.
```

▼ **-d オプションを指定して install-ldm スクリプトを使用してインストールする**

- Solaris Security Toolkit のカスタマイズされた強化ドライバ (たとえば、`server-secure-myname.driver`) を指定するには、`-d` オプションを指定してインストールスクリプトを実行します。

インストールスクリプトは、SUNWldm パッケージの一部で、Install サブディレクトリにあります。

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

処理が正常に実行されると、コード例 3-4 のようなメッセージが表示されます。

コード例 3-4 `install-ldm -d` スクリプトが正常に実行された場合の出力

```
# 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.

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
```

コード例 3-4 `install-ldm -d` スクリプトが正常に実行された場合の出力 (続き)

```
/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.
```

### ▼ `-d none` オプションを指定して `install-ldm` スクリプトを使用してインストールする

- Solaris Security Toolkit ドライバを使用してシステムを強化しないことを指定するには、`-d none` オプションを指定してインストールスクリプトを実行します。  
インストールスクリプトは、SUNWldm パッケージの一部で、Install サブディレクトリにあります。

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

処理が正常に実行されると、コード例 3-5 のようなメッセージが表示されます。

コード例 3-5 `install-ldm -d none` スクリプトが正常に実行された場合の出力

```
# 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.
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.
```

## ▼ -p オプションを指定して install-ldm スクリプトを使用してインストールする

SUNWldm および SUNWjass パッケージがサーバーにプリインストールされており、Logical Domains Manager デーモン (ldmd) の有効化および Solaris Security Toolkit の実行といったインストール後の処理を行う必要がある場合は、このオプションを使用できます。

- システムを強化するために ldmd の有効化および Solaris Security Toolkit の実行といったインストール後の処理のみを実行するには、-p オプションを指定してインストールスクリプトを実行します。

```
# 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.
```

## JumpStart を使用した Logical Domains Manager 1.0.2 および Solaris Security Toolkit 4.2 ソフトウェアのインストール

JumpStart の使用法の詳細は、『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。



**注意** – ネットワークインストール中は、仮想コンソールから接続を解除しないでください。

## ▼ JumpStart サーバーを設定する

- JumpStart サーバーがすでに設定されている場合は、この管理ガイドの [31 ページ](#) の「[JumpStart ソフトウェアを使用してインストールする](#)」に進んでください。
- JumpStart サーバーがまだ設定されていない場合は、これを設定する必要があります。

この手順の詳細は、『Solaris 10 11/06 インストールガイド (カスタム JumpStart/上級編)』を参照してください。このインストールガイドは、次の URL で入手できます。

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

- 『Solaris 10 11/06 インストールガイド (カスタム JumpStart/上級編)』の第 3 章「カスタム JumpStart インストールの準備 (作業)」を参照し、次の手順を実行します。
  - 「作業マップ: カスタム JumpStart インストールの準備」で作業マップを確認します。
  - 「ネットワーク上のシステム用のプロファイルサーバーの作成」の手順に従って、ネットワークに接続されたシステムを設定します。
  - 「rules ファイルの作成」の手順に従って、rules ファイルを作成します。
- 「rules ファイルの妥当性を検査する」の手順に従って、rules ファイルの妥当性検査を行います。

Solaris Security Toolkit では、プロファイルおよび終了スクリプトが提供されています。プロファイルおよび終了スクリプトの詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

## ▼ JumpStart ソフトウェアを使用してインストールする

- Solaris Security Toolkit パッケージ (SUNWjass) をダウンロードしたディレクトリに移動します。

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

- SUNWjass をインストールして、JumpStart (jumpstart) ディレクトリ構造を作成します。

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

- テキストエディタを使用して、ネットワーク環境を反映するように /jumpstart/opt/SUNWjass/Sysidcfg/Solaris\_10/sysidcfg ファイルを変更します。
- /jumpstart/opt/SUNWjass/Drivers/user.init.SAMPLE ファイルを /jumpstart/opt/SUNWjass/Drivers/user.init ファイルにコピーします。

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

- パスを反映するように user.init ファイルを編集します。

- JumpStart のインストール中に Solaris Security Toolkit パッケージ (SUNWjass) を対象のシステムにインストールするには、`user.init` ファイルで定義した `JASS_PACKAGE_MOUNT` ディレクトリにこのパッケージを配置する必要があります。次に例を示します。

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

- JumpStart のインストール中に Logical Domains Manager パッケージ (SUNWldm.v) を対象のシステムにインストールするには、`user.init` ファイルで定義した `JASS_PACKAGE_MOUNT` ディレクトリにダウンロード領域からこのパッケージを配置する必要があります。次に例を示します。

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

- マルチホーム JumpStart サーバーで問題が発生した場合は、`user.init` ファイル内の `JASS_PACKAGE_MOUNT` および `JASS_PATCH_MOUNT` に関する 2 つのエントリを、`JASS_HOME_DIR/Patches` および `JASS_HOME_DIR/Packages` ディレクトリへの正しいパスに変更します。詳細は、`user.init.SAMPLE` ファイル内のコメントを参照してください。
- Logical Domains Manager 制御ドメインの基本ドライバとして `ldm_control-secure.driver` を使用します。  
使用するドライバを変更する方法の詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』の第 4 章を参照してください。`ldm_control-secure.driver` に対応する Solaris Security Toolkit のメインドライバは、`secure.driver` です。
- `ldm_control-secure.driver` への変更が完了したら、`rules` ファイルに適切なエントリを作成します。
  - LDoms 制御ドメインを最小化する場合は、`rules` ファイル内の `minimal-ldm-control.profile` を次のように指定します。

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

---

注 – LDoms および Solaris Security Toolkit パッケージをインストールしたあとで、LDoms MIB ソフトウェアパッケージを手動でインストールする必要があります。これは、ほかのパッケージとともに自動的にインストールされません。LDoms MIB のインストールおよび使用法の詳細は、『Logical Domains (LDoms) MIB 1.0.1 管理ガイド』を参照してください。

---

- LDoms 制御ドメインを最小化しない場合は、エント리는次のようになります。

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

11. JumpStart のインストール中の強化を取り消すには、次の SMF コマンドを実行して Logical Domains Manager を再起動する必要があります。

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

## Logical Domains Manager および Solaris Security Toolkit ソフトウェアの手動インストール

Logical Domains Manager および Solaris Security Toolkit ソフトウェアを手動でインストールするには、次の手順を実行します。

- [33 ページの「Logical Domains Manager \(LDoms\) 1.0.2 ソフトウェアを手動でインストールする」](#)
- [34 ページの「\(省略可能\) Solaris Security Toolkit 4.2 ソフトウェアを手動でインストールする」](#)
- [34 ページの「\(省略可能\) 制御ドメインを手動で強化する」](#)

### ▼ Logical Domains Manager (LDoms) 1.0.2 ソフトウェアを手動でインストールする

Sun のソフトウェアダウンロードサイトから、Logical Domains Manager 1.0.2 ソフトウェアの SUNWldm パッケージをダウンロードします。具体的な手順については、[22 ページの「Logical Domains Manager、Solaris Security Toolkit、および Logical Domains MIB をダウンロードする」](#)を参照してください。

1. pkgadd(1M) コマンドを使用して、SUNWldm.v パッケージをインストールします。-G オプションを使用して大域ゾーンのみパッケージをインストールするよう指定し、-d オプションを使用して SUNWldm.v パッケージを含むディレクトリのパスを指定します。

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

2. 対話型プロンプトのすべての質問に対して、y (はい) と答えます。

3. `pkginfo(1)` コマンドを使用して、Logical Domains Manager 1.0.2 ソフトウェア用の `SUNWldm` パッケージがインストールされていることを確認します。

バージョン (REV) 情報の例を次に示します。

```
# pkginfo -l SUNWldm | grep VERSION
VERSION=1.0.2,REV=2007.08.23.10.20
```

## ▼ (省略可能) Solaris Security Toolkit 4.2 ソフトウェアを手動でインストールする

システムをセキュリティー保護するには、`SUNWjass` パッケージをダウンロードしてインストールします。必須パッチ (122608-03 および 125672-01) は、`SUNWjass` パッケージに含まれています。ソフトウェアのダウンロードに関する詳細は、[22 ページの「Logical Domains Manager、Solaris Security Toolkit、および Logical Domains MIB をダウンロードする」](#)を参照してください。

Logical Domains Manager ソフトウェアを使用する場合のセキュリティーに関する考慮事項の詳細は、このマニュアルの [第 2 章](#)を参照してください。さらに詳細を確認するには、次の URL で Solaris Security Toolkit 4.2 のマニュアルを参照できます。

<http://docs.sun.com>

1. `pkgadd(1M)` コマンドを使用して、`SUNWjass` パッケージをインストールします。

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

2. `pkginfo(1)` コマンドを使用して、Solaris Security Toolkit 4.2 ソフトウェアの `SUNWjass` パッケージがインストールされていることを確認します。

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

## ▼ (省略可能) 制御ドメインを手動で強化する

Solaris Security Toolkit 4.2 パッケージがすでにインストールされている場合にかぎり、この手順を実行してください。

---

**注** – Solaris Security Toolkit を使用して制御ドメインを強化すると、多くのシステムサービスが無効になり、ネットワークアクセスに一定の制限が生じます。詳細は、このマニュアルの [xx ページの「関連マニュアル」](#)を参照して、Solaris Security Toolkit 4.2 のマニュアルで確認してください。

---



1. `ldm_control-secure.driver` を使用して強化します。

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

システムを強化するために、ほかのドライバを使用できます。また、ドライバをカスタマイズして、使用している環境のセキュリティーを調整することもできます。ドライバとそのカスタマイズ方法の詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

2. 対話型プロンプトのすべての質問に対して、`y` (はい) と答えます。
3. 強化を有効にするため、サーバーを停止してから再起動します。

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

## ▼ 強化の妥当性検査を行う

- Logical Domains 強化ドライバ (`ldm_control-secure.driver`) によって、強化が適切に適用されたかどうかを確認します。

別のドライバについて確認する場合は、次のコマンド例のドライバ名を置き換えてください。

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

## ▼ 強化を取り消す

1. Solaris Security Toolkit によって適用された構成の変更を取り消します。

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

Solaris Security Toolkit によって、どの強化の実行を取り消すかが尋ねられます。

2. 取り消す強化の実行を選択します。
3. 構成の強化の取り消しが行われるように、システムを再起動します。

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

---

注 – JumpStart のインストール中に実行された強化を取り消すには、次の SMF コマンドを実行して Logical Domains Manager および仮想ネットワーク端末サーバーデーモンを再起動する必要があります。

---

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

## Logical Domains Manager デーモンの有効化

インストールスクリプト `install-ldm` を使用すると、Logical Domains Manager デーモン (`ldmd`) が自動的に有効になります。Logical Domains Manager ソフトウェアが手動でインストールされた場合は、Logical Domains Manager デーモンの `ldmd` を有効にする必要があります。これにより、論理ドメインの作成、変更、および制御が可能になります。

### ▼ Logical Domains Manager デーモンを有効にする

1. `svcadm(1M)` コマンドを使用して、Logical Domains Manager デーモンの `ldmd` を有効にします。

```
# svcadm enable ldmd
```

2. `ldm list` コマンドを使用して、Logical Domains Manager デーモンが実行中であることを確認します。

次のようなメッセージが表示されます。これは、`factory-default` 構成の場合です。`primary` ドメインは `active` になっています。これは、Logical Domains Manager が実行中であることを意味します。

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

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

---

# ユーザーアカウントに対する承認およびプロファイルの作成と役割の割り当て

Logical Domains Manager 用に変更された Solaris OS の役割に基づくアクセス制御 (RBAC) を使用して、ユーザーアカウントに対する承認およびプロファイルを設定し、役割を割り当てます。RBAC の詳細は、Solaris 10 System Administrator Collection を参照してください。

Logical Domains Manager の承認には、次の 2 つのレベルがあります。

- 読み取り – 構成を表示できますが、変更できません。
- 読み取りおよび書き込み – 構成を表示および変更できます。

Solaris OS の `/etc/security/auth_attr` ファイルには、次の Logical Domains エントリが自動的に追加されます。

- `Solaris.ldoms.::LDoms Administration::`
- `Solaris.ldoms.grant::Delegate Ldoms Configuration::`
- `Solaris.ldoms.read::View Ldoms Configuration::`
- `Solaris.ldoms.write::Manage Ldoms Configuration::`

## ユーザー承認の管理

### ▼ ユーザーの承認を追加する

必要に応じて次の手順を使用して、Logical Domains Manager ユーザーに対する承認を `/etc/security/auth_attr` ファイルに追加します。スーパーユーザーには `solaris.*` 承認がすでに設定されているため、スーパーユーザーは `solaris.ldoms.*` 承認の承認をすでに持っています。

1. `ldm(1M)` のサブコマンドを使用するために承認を必要とするユーザーごとに、ローカルユーザーアカウントを作成します。

---

注 – ユーザーの Logical Domains Manager 承認を追加するには、そのユーザーに対してローカル (非 LDAP) アカウントを作成する必要があります。詳細は、Solaris 10 System Administrator Collection を参照してください。

---

2. ユーザーによるアクセスを可能にする `ldm(1M)` のサブコマンドに応じて、次のいずれかを実行します。

`ldm(1M)` コマンドとそれらのユーザー承認の一覧は、表 2-1 を参照してください。

- `usermod(1M)` コマンドを使用して、ユーザーの読み取り専用承認を追加します。

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

- `usermod(1M)` コマンドを使用して、ユーザーの読み取りおよび書き込み承認を追加します。

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

## ▼ ユーザーのすべての承認を削除する

- ローカルユーザーアカウントのすべての承認を削除します (使用できる唯一のオプション)。

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

## ユーザープロファイルの管理

SUNWldm パッケージによって、`/etc/security/prof_attr` ファイルにシステムで定義された 2 つの RBAC プロファイルが追加されます。これらは、スーパーユーザー以外による Logical Domains Manager へのアクセスを承認するために使用されます。2 つの LDoms 固有のプロファイルは次のとおりです。

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

次の手順を使用して、前述のいずれかのプロファイルをユーザーアカウントに割り当てることができます。

## ▼ ユーザーのプロファイルを追加する

- ローカルユーザーアカウントに管理プロファイル (たとえば、LDoms Management) を追加します。

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

## ▼ ユーザーのすべてのプロフィールを削除する

- ローカルユーザーアカウントのすべてのプロフィールを削除します (使用できる唯一のオプション)。

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

## ユーザーへの役割の割り当て

この手順を使用する利点は、特定の役割が割り当てられたユーザーだけがその役割になることができることです。役割にパスワードが設定されている場合は、その役割になるときにパスワードが必要になります。これにより、2層のセキュリティーが実現します。ユーザーに役割が割り当てられていない場合、ユーザーがその正しいパスワードを知っていたとしても、`su role_name` コマンドを実行してその役割になることはできません。

## ▼ 役割を作成し、ユーザーにその役割を割り当てる

1. 役割を作成します。

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

2. 役割にパスワードを割り当てます。

```
# passwd ldm_read
```

3. ユーザー (たとえば `user_1`) に役割を割り当てます。

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

4. ユーザー (`user_1`) にパスワードを割り当てます。

```
# passwd user_1
```

5. `ldm_read` アカウントになるために、`user_1` アカウントに対するアクセス権のみを割り当てます。

```
# su user_1
```

6. プロンプトが表示されたら、ユーザーのパスワードを入力します。

7. ユーザー ID を確認して、ldm\_read 役割にアクセスします。

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

8. 読み取り承認を持つ ldm サブコマンドに対して、ユーザーにアクセス権を提供します。

```
# su ldm_read
```

9. プロンプトが表示されたら、ユーザーのパスワードを入力します。
10. id コマンドを入力してユーザーを表示します。

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

## 第4章

---

# サービスと論理ドメインの設定

---

この章では、デフォルトのサービス、制御ドメイン、およびゲストドメインの設定に必要な手順について説明します。

---

## 出力メッセージ

デフォルトのサービスの作成や制御 (primary) ドメインの設定に使用するコマンドで表示される出力メッセージは、プラットフォームによって異なります。

- Sun UltraSPARC T1 プロセッサ
- Sun UltraSPARC T2 プロセッサ

## Sun UltraSPARC T1 プロセッサ

Sun UltraSPARC T1 プロセッサを搭載したサーバーを使用している場合は、primary ドメインの設定コマンドのあとで、次のような通知が表示されます。

```
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 プロセッサ

最初の操作 – Sun UltraSPARC T2 プロセッサを搭載したサーバーを使用している場合は、`primary` ドメインのいずれかのデバイスまたはサービスに対して最初の操作を実行したあとで、次のようなメッセージが表示されます。

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

再起動までの以降の操作 – Sun UltraSPARC T2 プロセッサを搭載したサーバーを使用している場合は、再起動するまで `primary` ドメインに対して設定コマンドを実行するごとに、次のような通知が表示されます。

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

---

## デフォルトのサービスの作成

あとで使用できるように、次のデフォルトの仮想サービスを最初に作成する必要があります。

- `vdiskserver` – 仮想ディスクサーバー
- `vswitch` – 仮想スイッチサービス
- `vconscn` – 仮想コンソール端末集配信装置サービス

### ▼ デフォルトのサービスを作成する

1. 論理ドメインに仮想ディスクをインポートできるように、仮想ディスクサーバー (`vds`) を作成します。  
たとえば、次のコマンドを使用して、仮想ディスクサーバー (`primary-vds0`) を制御ドメイン (`primary`) に追加します。

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



2. 仮想ネットワーク端末サーバーデーモン (vntsd) が使用する仮想コンソール端末集配装置サービス (vcc) を、すべての論理ドメインコンソールの端末集配装置として作成します。

たとえば、次のコマンドを使用して、ポートの範囲が 5000 ~ 5100 までの仮想コンソール端末集配装置サービス (primary-vcc0) を、制御ドメイン (primary) に追加します。

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

3. 論理ドメインの仮想ネットワーク (vnet) デバイス間でネットワークを有効にするには、仮想スイッチサービス (vsw) を作成します。各論理ドメインが仮想スイッチを使用して外部と通信する必要がある場合は、GLDv3 準拠のネットワークアダプタを仮想スイッチに割り当てます。

たとえば、次のコマンドを使用して、ネットワークアダプタドライバ e1000g0 の仮想スイッチサービス (primary-vsw0) を、制御ドメイン (primary) に追加します。

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

このコマンドによって、仮想スイッチに MAC アドレスが自動的に割り当てられます。ldm add-vsw コマンドに、オプションとして独自の MAC アドレスを指定できます。ただし、この場合、指定した MAC アドレスが既存の MAC アドレスと競合していないことの確認は、ユーザーが責任を持って行います。

追加された仮想スイッチが、基本となる物理アダプタに代わり主ネットワークインタフェースとなる場合は、動的ホスト構成プロトコル (DHCP) サーバーによってドメインに同じ IP アドレスが割り当てられるように、仮想スイッチに物理アダプタの MAC アドレスを割り当てる必要があります。[46 ページの「制御ドメインまたはサービスドメインとその他のドメイン間のネットワークの有効化」](#)を参照してください。

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

4. list-services サブコマンドを使用して、サービスが作成されたことを確認します。次のように出力されるはずです。

```
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

---

## 制御ドメインの初期構成

最初に、すべてのシステムリソースが制御ドメインに割り当てられます。その他の論理ドメインを作成できるように、一部のリソースを解放する必要があります。

---

注 – 以降の例の出力で LDoms Manager が構成モードで実行されているという注意事項は、Sun UltraSPARC T1 プロセッサにのみ適用されます。

---

### ▼ 制御ドメインを設定する

---

注 – この手順には、制御ドメイン用に設定するリソースの例も含まれています。ここで示す数値は単なる例であり、使用される値が制御ドメインに適していない場合があります。

---

#### 1. 暗号化リソースを制御ドメインに割り当てます。

---

注 – 制御ドメインに暗号化デバイスが割り当てられている場合は、CPU を動的に再構成することはできません。そのため、暗号化デバイスを使用していない場合は、`set-mau` を 0 にします。

---

次の例では、1 つの暗号化リソースが制御ドメイン `primary` に割り当てられます。これによって、残りの暗号化リソースをゲストドメインで使用できるようになります。

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

2. 仮想 CPU を制御ドメインに割り当てます。

たとえば、次のコマンドでは、4 つの仮想 CPU が制御ドメイン `primary` に割り当てられます。これにより、残りの仮想 CPU をゲストドメインで使用できるようになります。

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

3. メモリーを制御ドメインに割り当てます。

たとえば、次のコマンドでは、1G バイトのメモリーが制御ドメイン `primary` に割り当てられます。これにより、残りのメモリーをゲストドメインで使用できるようになります。

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

---

注 – ZFS を使用せずにディスクサービスを提供する場合、メモリーは 1G バイトで十分であるはずですが、ZFS を使用してディスクサービスを提供する場合は、完全なコアの 4 つの仮想 CPU と、4G バイト以上のメモリーを割り当てます。I/O 負荷が高い場合は、追加の完全なコアの割り当てが必要になる場合があります。

---

4. システムコントローラ (SC) に論理ドメインの機械構成を追加します。

たとえば、次のコマンドを使用して `initial` という名前の構成を追加します。

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

---

注 – 現在、SC に保存できる構成数の上限は 8 つです。この数には、`factory-default` 構成は含まれません。

---

5. 次回の再起動時に構成が使用できる状態であることを確認します。

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

この `list` サブコマンドでは、現在は `factory-default` 構成設定が使用されており、再起動したあとで `initial` 構成設定が使用されることを示しています。

---

## 論理ドメインを使用するための再起動

構成の変更を有効にして、ほかの論理ドメインで使用できるようにリソースを解放するには、制御またはサービスを再起動する必要があります。

### ▼ 論理ドメインを使用するために再起動する

- primary ドメインを停止して再起動します。この例では、primary はサービスドメインでもあります。

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

---

注 - 指定されたコマンドを使用すると、再起動により変更が有効になりますが、`ldm list-config` コマンドでは再起動する前と同じ出力が表示されます。`ldm list-config` コマンドで表示される構成を更新するには、電源を切断してから再投入する必要があります。

---

---

## 制御ドメインまたはサービスドメインとその他のドメイン間のネットワークの有効化

デフォルトでは、システムの制御ドメインまたはサービスドメインとその他のドメイン間のネットワークは無効になっています。これを有効にするために、仮想スイッチデバイスをネットワークデバイスとして構成するようにしてください。仮想スイッチは、基本となる物理デバイス (この例では `e1000g0`) に代わり主インタフェースとして構成するか、ドメインの追加のネットワークインタフェースとして構成することができます。

---

注 - この手順によってドメインへのネットワーク接続が一時的に中断される可能性があるため、次の構成手順はドメインのコンソールから実行してください。

---

## ▼ 仮想スイッチを主インタフェースとして構成する

1. すべてのインタフェースのアドレス指定情報を表示します。

```
primary# ifconfig -a
```

2. 仮想スイッチを plumb します。この例では、構成する仮想スイッチは vsw0 です。

```
primary# ifconfig vsw0 plumb
```

3. (省略可能) ドメイン内のすべての仮想スイッチインスタンスのリストを取得するために、仮想スイッチインスタンスを一覧で表示できます。

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

4. 仮想スイッチ (net-dev) に割り当てられた物理ネットワークデバイスを unplumb します。この例では、物理ネットワークデバイスは e1000g0 です。

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

5. 物理ネットワークデバイス (e1000g0) のプロパティを仮想スイッチ (vsw0) デバイスに移行するには、次のいずれかを実行します。

- ネットワークが静的 IP アドレスを使用して構成されている場合は、vsw0 に対して e1000g0 の IP アドレスとネットマスクを再利用します。

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

- ネットワークが DHCP を使用して構成されている場合は、vsw0 に対して DHCP を有効にします。

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

6. 必要な構成ファイルに修正を加えて、この変更内容を確定します。

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

---

注 – 必要に応じて、物理ネットワークデバイスと同様に仮想スイッチも構成できます。この場合、手順 2 で記載されているように仮想スイッチを `plumb` して、物理デバイスは、`unplumb` しません (手順 4 をスキップする)。そのあと、仮想スイッチは、静的 IP アドレスを使用するか、DHCP サーバーから動的 IP アドレスを取得して構成する必要があります。

---

---

## 仮想ネットワーク端末サーバーデーモンの有効化

各論理ドメインの仮想コンソールにアクセスするには、仮想ネットワーク端末サーバーデーモン (`vntsd`) を有効にする必要があります。このデーモンの使用法の詳細は、Solaris 10 Reference Manual Collection または `vntsd(1M)` マニュアルページを参照してください。

### ▼ 仮想ネットワーク端末サーバーデーモンを有効にする

---

注 – `vntsd` を有効にする前に、制御ドメインにデフォルトのサービス `vconscn` が作成されていることを確認してください。詳細は、[42 ページの「デフォルトのサービスの作成」](#)を参照してください。

---

1. `svcadm(1M)` コマンドを使用して、仮想ネットワーク端末サーバーデーモン `vntsd(1M)` を有効にします。

```
# svcadm enable vntsd
```

2. `svcs(1)` コマンドを使用して、`vntsd` が有効であることを確認します。

```
# 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)
```

## ゲストドメインの作成と起動

ゲストドメインでは、sun4v プラットフォームとハイパーバイザによって提供される仮想デバイスの両方を認識するオペレーティングシステムを実行する必要があります。現在は、Solaris 10 11/06 以上の OS である必要があります。必要になる可能性がある特定のパッチについては、『Logical Domains (LDoms) 1.0.2 リリースノート』を参照してください。デフォルトのサービスを作成し、制御ドメインからリソースを再度割り当てたら、ゲストドメインを作成して起動できます。

### ▼ ゲストドメインを作成して起動する

#### 1. 論理ドメインを作成します。

たとえば、次のコマンドを使用して `ldg1` という名前のゲストドメインを作成します。

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

#### 2. CPU をゲストドメインに追加します。

たとえば、次のコマンドを使用して 4 つの仮想 CPU をゲストドメイン `ldg1` に追加します。

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

#### 3. メモリーをゲストドメインに追加します。

たとえば、次のコマンドを使用して 512M バイトのメモリーをゲストドメイン `ldg1` に追加します。

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

#### 4. 仮想ネットワークデバイスをゲストドメインに追加します。

たとえば、次のコマンドを使用して、次のように指定した仮想ネットワークデバイスをゲストドメイン `ldg1` に追加します。

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

各表記の意味は次のとおりです。

- `vnet1` は、後続の `set-vnet` または `remove-vnet` サブコマンドで参照するためにこの仮想ネットワークデバイスのインスタンスに割り当てられる、論理ドメインで一意的なインタフェース名です。
- `primary-vsw0` は、接続する既存のネットワークサービス (仮想スイッチ) の名前です。

#### 5. 仮想ディスクサーバーによってゲストドメインに仮想ディスクとしてエクスポートされるデバイスを指定します。

物理ディスク、ディスクスライス、ボリューム、またはファイルをブロック型デバイスとしてエクスポートできます。このリリースの Logical Domains ソフトウェアでは、ループバック (`lofi`) デバイスのブロック型デバイスとしてのエクスポートはサポートされていません。物理ディスクとファイルの例を次に示します。

- 物理ディスクの例。最初の例では、次の指定で物理ディスクを追加します。

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

各表記の意味は次のとおりです。

- `/dev/dsk/c0t0d0s2` は、実際の物理デバイスのパス名です。デバイスを追加する場合、パス名にはデバイス名を組み合わせる必要があります。
- `vol1` は、仮想ディスクサーバーに追加するデバイスに指定する必要がある一意的な名前です。デバイス名は、この仮想ディスクサーバーによってクライアントにエクスポートされ追加されるため、デバイス名はこの仮想ディスクサーバーのインスタンスに対して一意である必要があります。デバイスを追加する場合、デバイス名には実際のデバイスのパス名を組み合わせる必要があります。
- `primary-vds0` は、このデバイスを追加する仮想ディスクサーバーの名前です。
- ファイルの例。この 2 つめの例では、ファイルをブロック型デバイスとしてエクスポートします。

```
primary$ ldm add-vdsdev path-to-file/filename vol1@primary-vds0
```

各表記の意味は次のとおりです。



- `path-to-file/filename` は、ブロック型デバイスとしてエクスポートされる実際のファイルのパス名です。デバイスを追加する場合、パス名にはデバイス名を組み合わせる必要があります。
- `vol1` は、仮想ディスクサーバーに追加するデバイスに指定する必要がある一意の名前です。デバイス名は、この仮想ディスクサーバーによってクライアントにエクスポートされ追加されるため、デバイス名はこの仮想ディスクサーバーのインスタンスに対して一意である必要があります。デバイスを追加する場合、デバイス名には実際のデバイスのパス名を組み合わせる必要があります。
- `primary-vds0` は、このデバイスを追加する仮想ディスクサーバーの名前です。

## 6. 仮想ディスクをゲストドメインに追加します。

次の例では、仮想ディスクをゲストドメイン `ldg1` に追加します。

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

各表記の意味は次のとおりです。

- `vdisk1` は、仮想ディスクの名前です。
- `vol1` は、接続する既存の仮想ディスクサーバーのデバイスの名前です。
- `primary-vds0` は、接続する既存の仮想ディスクサーバーの名前です。

---

**注** – 仮想ディスクは、さまざまな種類の物理デバイス、ボリューム、またはファイルで構成される総称的なブロック型デバイスです。仮想ディスクは SCSI ディスクと同義ではありません。そのため、ディスクラベル内のターゲット ID は除外されません。論理ドメインの仮想ディスクの形式は、`cNdNsN` です。cN は仮想コントローラ、dN は仮想ディスク番号、および sN はスライスを示します。

---

## 7. ゲストドメインの `auto-boot` および `boot-device` 変数を設定します。

最初の例のコマンドは、ゲストドメイン `ldg1` の `auto-boot\?` を `true` に設定します。

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

2 つめの例のコマンドは、ゲストドメイン `ldg1` の `boot-device` を `vdisk` に設定します。

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

8. ゲストドメイン `ldg1` にリソースをバインドし、ドメインを一覧表示してリソースがバインドされていることを確認します。

```
primary$ ldm bind-domain ldg1
primary$ ldm list-domain ldg1
NAME           STATE      FLAGS    CONS   VCPU  MEMORY  UTIL  UPTIME
ldg1           bound     -----  5001   4     512M
```

9. ゲストドメインのコンソールのポートを見つけるために、前述の `list-domain` サブコマンドの出力を調べます。

CONS という見出しの下で、論理ドメインゲスト 1 (`ldg1`) のコンソール出力がポート 5001 にバインドされていることがわかります。

10. ゲストドメイン `ldg1` を起動します。

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

11. ゲストドメインのコンソールに接続します。この処理は、いくつかの方法で実行できます。

- 制御ドメインにログインして、ローカルホストのコンソールポートに直接接続できます。

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

- また、`vntsd(1M)` の SMF マニフェストで有効になっている場合は、ネットワークを介してゲストコンソールに接続できます。次に例を示します。

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

サービス管理機能マニフェストは、サービスが記述された XML ファイルです。SMF マニフェストの作成については、*Solaris 10 System Administrator Collection* を参照してください。

---

注 – コンソールを使用してゲストドメインの英語版以外の OS にアクセスするには、コンソールの端末が、その OS が必要とするロケールになっている必要があります。

---

---

## ゲストドメインの JumpStart

ゲストドメインの JumpStart を行う場合、次の 2 つの例で示すように、正規の Solaris OS の JumpStart 手順にあるプロファイルの構文を LDom 固有の JumpStart 手順に変更して、通常の JumpStart 手順を使用します。

通常の **JumpStart** のプロファイル

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

論理ドメインの仮想ディスクデバイス名は、デバイス名にターゲット ID (tN) が含まれないという点で、物理ディスクデバイス名とは異なります。通常の cNtNdNsN 形式の代わりに、仮想ディスクデバイス名は cNdNsN という形式になります。ここで、cN は仮想コントローラ、dN は仮想ディスク番号、および sN はスライスを示します。次のプロファイルの例のように、使用する JumpStart プロファイルを修正して、この変更を反映してください。

論理ドメインで使用される実際のプロファイル

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



## 第5章

---

# その他の情報とタスク

---

この章では、ここまでの章では説明していない Logical Domains ソフトウェアの使用に関する情報とタスクについて説明します。

---

## CLI で名前を入力する場合の制限

次の節では、Logical Domains Manager CLI で名前を入力する場合の制限について説明します。

### ファイル名 (*file*) と変数名 (*var\_name*)

- 最初の文字は、英字、数字、またはスラッシュ (/) である必要があります。
- 以降の文字は、英字、数字、または句読点である必要があります。

### 仮想ディスクサーバー *file|device* および仮想スイッチデバイス名

- 英字、数字、または句読点を含む必要があります。

### 構成名 (*config\_name*)

システムコントローラに格納されている構成に割り当てる論理ドメイン構成名 (*config\_name*) は、64 文字以下である必要があります。

## その他のすべての名前

論理ドメイン名 (*ldom*)、サービス名 (*vswitch\_name*、*service\_name*、*vdpcs\_service\_name*、および *vcc\_name*)、仮想ネットワーク名 (*if\_name*)、および仮想ディスク名 (*disk\_name*) などのその他の名前は、次のような形式である必要があります。

- 最初の文字は、英字または数字である必要があります。
- 以降の文字は、英字、数字、または「`_-+#.::~~()`」のいずれかの文字である必要があります。

---

## ldm list サブコマンドの使用

この節では、ldm サブコマンドの構文の使用法、フラグや利用統計情報などの出力項目の定義、および出力例について説明します。

### マシンが読み取り可能な出力

ldm list コマンドの出力を使用するスクリプトを作成する場合は、常に `-p` オプションを使用してマシンが読み取り可能な形式で出力を生成します。詳細は、[64 ページ](#)の「[解析可能でマシンが読み取り可能なリストを生成する \(-p\)](#)」を参照してください。

### ▼ ldm サブコマンドの構文の使用法を表示する

- ldm のすべてのサブコマンドの構文の使用法を表示するには、次のコマンドを実行します。

コード例 5-1 ldm のすべてのサブコマンドの構文の使用法

```
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
```

コード例 5-1 ldm のすべてのサブコマンドの構文の使用法 (続き)

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

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

devices
  list-devices [-a] [-p] [cpu] [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>

mau
  add-mau <number> <ldom>
  set-mau <number> <ldom>
  remove-mau <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>
```

コード例 5-1 ldm のすべてのサブコマンドの構文の使用法 (続き)

```
vconscon    ( vcc )
  add-vconscon port-range=<x>-<y> <vcc_name> <ldom>
  set-vconscon port-range=<x>-<y> <vcc_name>
  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>
  remove-vdisk [-f] <disk_name> <ldom>

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

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

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

vdiskserverdevice ( vdsdev )
  add-vdiskserverdevice [options=<opts>] <file|device>
<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>] <vswitch_name> <ldom>
  set-vswitch [mac-addr=<num>] [net-dev=<device>] <vswitch_name>
  remove-vswitch [-f] <vswitch_name>
```

Verb aliases:



コード例 5-1 ldm のすべてのサブコマンドの構文の使用法 (続き)

```
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
```

## フラグの定義

ドメインの出力に次のフラグを表示できます。

- 可変部分
- c 制御ドメイン
- d 遅延再構成
- n 通常
- s 起動または停止
- t 切り替え
- v 仮想 I/O ドメイン

コマンドに長形式 (-l) オプションを使用すると、フラグは省略されずに表示されます。このオプションを使用しない場合は、略号が表示されます。

リストフラグ値は位置に依存します。次に、左から順に 5 つの列のそれぞれに表示される可能性のある値を示します。

列 1	列 2	列 3	列 4	列 5
s または -	n または t	d または -	c または -	v または -

## 利用統計情報の定義

ldm list コマンドの長形式 (-l) オプションでは、仮想 CPU ごとの利用統計情報 (UTIL) が表示されます。この統計情報は、ゲストオペレーティングシステムの代わりに仮想 CPU が実行に費やした時間の、前回統計を表示した以降の割合です。仮想 CPU は、ハイパーバイザに制御が渡される場合を除き、ゲストオペレーティングシステムに代わって実行するものと考えられます。ゲストオペレーティングシステムが仮想 CPU の制御をハイパーバイザに渡さない場合、ゲストオペレーティングシステムの CPU の利用率は常に 100% として表示されます。

論理ドメインについて報告された利用統計情報は、ドメインの仮想 CPU に対する仮想 CPU 利用率の平均です。

## さまざまなリストの例

### ▼ ソフトウェアのバージョンを表示する (-V)

- 現在インストールされているソフトウェアのバージョンを表示するには、次のコマンドを実行します。例に示すようなリストが出力されます。

コード例 5-2 インストールされているソフトウェアのバージョン

```
primary$ ldm -v

Logical Domain Manager (v 1.0.2)
  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
```

### ▼ 省略形式のリストを生成する

- すべてのドメインの省略形式のリストを生成するには、次のように実行します。

コード例 5-3 すべてのドメインの省略形式のリスト

```
primary$ ldm list

NAME          STATE   FLAGS   CONS   VCPU  MEMORY  UTIL  UPTIME
primary      active -t-cv   5000   4     1G     0.5%  3d 21h 7m
ldg1         active -t---   5000   8     1G     23%   2m
```

## ▼ 長形式のリストを生成する (-1)

- すべてのドメインの長形式のリストを生成するには、次のように実行します。

コード例 5-4      すべてのドメインの長形式のリスト

```
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

VDS
  NAME                VOLUME            OPTIONS        DEVICE
  vds0                myvol-a           slice          /disk/a
                   myvol-b           /disk/b
  vds1                myvol-c           ro,slice,excl /disk/c
                   myvol-d           /disk/d

VDPCS
  NAME
  vdpcs0
  vdpcs1

-----
NAME                STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
ldg1                bound   ------  5000    1      512M

VCPU
```

コード例 5-4 すべてのドメインの長形式のリスト (続き)

```

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

```

## ▼ 拡張リストを生成する (-e)

- すべてのドメインの拡張リストを生成するには、次のように実行します。

コード例 5-5 すべてのドメインの拡張リスト

```

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

```

コード例 5-5 すべてのドメインの拡張リスト (続き)

```

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
hvctl1          primary@primary    hvctl1
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     0x3400000     512M

VLDCC
NAME             SERVICE            DESC

```

コード例 5-5 すべてのドメインの拡張リスト (続き)

vldcc0	primary@primary	ds		
NETWORK				
NAME	SERVICE	DEVICE	MAC	
myinet-b	vsw0@primary	network@0	08:00:20:ab:9a:12	
myinet-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		

### ▼ 解析可能でマシンが読み取り可能なリストを生成する (-p)

- すべてのドメインの解析可能でマシンが読み取り可能なリストを生成するには、次のように実行します。

コード例 5-6 マシンが読み取り可能なリスト

```
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=
```

### ▼ ドメインの状態を表示する

- ドメイン (ゲストドメイン ldg1 など) の状態を表示するには、次のように実行します。

コード例 5-7 ドメインの状態

```
primary# ldm list-domain ldg1
NAME          STATE   FLAGS   CONS   VCPU  MEMORY  UTIL  UPTIME
ldg1         active -t---   5000   8     1G     0.3%  2m
```

## ▼ 変数を一覧表示する

- ドメイン (ldg1 など) の変数 (boot-device など) を一覧表示するには、次のように実行します。

コード例 5-8      ドメインの変数のリスト

```
primary$ ldm list-variable boot-device ldg1
boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
```

## ▼ バインドを一覧表示する

- ドメインにバインドされたリソース (ldg1 など) を一覧表示するには、次のように実行します。

コード例 5-9      ドメインのバインドのリスト

```
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
```

## コード例 5-9 ドメインのバインドのリスト (続き)

VDPCC		
NAME	SERVICE	
myvdpcc-a	vdpcs0@primary	
myvdpcc-b	vdpcs0@primary	
VCONS		
NAME	SERVICE	PORT
mygroup	vcc0@primary	5000

### ▼ 構成を一覧表示する

- SC に格納されている論理ドメイン構成を一覧表示するには、次のように実行します。

#### コード例 5-10 構成のリスト

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

#### ラベルの意味

構成名の右にあるラベルの意味は、次のとおりです。

- current - 現在使用されている構成
- next - 次回電源を再投入するときに使用される構成

### ▼ デバイスを一覧表示する

- すべてのサーバーリソース (バインドされたリソースおよびバインドされていないリソース) を一覧表示するには、次のように実行します。

#### コード例 5-11 すべてのサーバーリソースのリスト

```
primary$ ldm list-devices -a
VCPU
  PID  %FREE
  0     0
  1     0
  2     0
  3     0
  4    100
  5    100
  6    100
  7    100
```



コード例 5-11 全てのサーバーリソースのリスト (続き)

```
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
```

コード例 5-11 すべてのサーバーリソースのリスト (続き)

IO	DEVICE	PSEUDONYM	BOUND	OPTIONS
	pci@780	bus_a	yes	
	pci@7c0	bus_b	yes	bypass=on

## ▼ サービスを一覧表示する

- 使用可能なサービスを一覧表示するには、次のように実行します。

コード例 5-12 サービスのリスト

```
primary$ ldm list-services
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
```

## 制約の一覧表示

Logical Domains Manager に対する制約とは、特定のドメインに割り当てられる 1 つ以上のリソースです。使用可能なリソースに応じて、ドメインに追加するように要求したすべてのリソースを受け取るか、まったく受け取らないかのいずれかです。list-constraints サブコマンドは、ドメインに割り当てるように要求したリソースを一覧表示します。

## ▼ 1 つのドメインの制約を一覧表示する

- 1 つのドメイン (ldg1 など) の制約を一覧表示するには、次のように実行します。

コード例 5-13 1 つのドメインの制約のリスト

```
primary$ ldm list-constraints ldg1
DOMAIN
ldg1

VCPU
COUNT
```

コード例 5-13 1つのドメインの制約のリスト (続き)

```
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
```

▼ 制約を XML 形式で一覧表示する

- 特定のドメイン (ldg1 など) の制約を XML 形式で一覧表示するには、次のように実行します。

コード例 5-14 ドメインの XML 形式の制約

```
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>
```

## コード例 5-14 ドメインの XML 形式の制約 (続き)

```
<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>nvramrc</name>
  <value>devalias vnet0 /virtual-devices@100/channel-devices@200/
network@0</value>
</var>
<var>
  <name>use-nvramrc?</name>
  <value>>true</value>
</var>
</ldom>
</data>
</LDM_interface>
```

### ▼ 制約をマシンが読み取り可能な形式で一覧表示する

- すべてのドメインの制約を解析可能な形式で一覧表示するには、次のように実行します。

## コード例 5-15 マシンが読み取り可能な形式のすべてのドメインの制約

```
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
```

コード例 5-15 マシンが読み取り可能な形式のすべてのドメインの制約 (続き)

```
|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=dealias 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
```

## ドメインの負荷が大きい場合に `ldm stop-domain` コマンドがタイムアウトする可能性がある

`ldm stop-domain` コマンドは、ドメインが完全に停止する前にタイムアウトする可能性があります。このような状況が発生すると、Logical Domains Manager によって次のようなエラーが返されます。

```
LDom ldg8 stop notification failed
```

しかし、ドメインが停止要求をまだ処理している可能性があります。`ldm list-domain` コマンドを使用して、ドメインの状態を確認します。次に例を示します。

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg8          active s---- 5000  22    3328M  0.3%  1d 14h 31m
```

前述のリストには、ドメインがアクティブと表示されていますが、`s` フラグはドメインが停止処理中であることを示しています。これは、一時的な状態であるはずですが。

次の例は、ドメインがすでに停止していることを示しています。

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg8          bound  ----- 5000  22    3328M
```

---

# 仮想ネットワークデバイスに対応する Solaris ネットワークインタフェース名の判定

ldm list-\* コマンドによって提供される出力で、特定の仮想デバイスに対応するゲストの Solaris OS ネットワークインタフェース名を直接判定する方法はありません。ただし、ldm list -l コマンドの出力と、Solaris OS ゲストの /devices 配下のエントリを組み合わせると、これを判定することができます。

## ▼ Solaris OS ネットワークインタフェース名を確認する

この例では、ゲストドメイン ldg1 に 2 つの仮想ネットワークデバイスの net-a と net-c が含まれており、ldg1 で net-c に対応する Solaris OS ネットワークインタフェース名を確認します。

1. ldm コマンドを使用して、net-c の仮想ネットワークデバイスインスタンスを探します。

```
# 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
...
#
```

net-c の仮想ネットワークデバイスインスタンスは network@2 です。

2. ldg1 で対応するネットワークインタフェースを検出するには、ldg1 にログインして、/devices 配下でこのインスタンスに対するエントリを探します。

```
# uname -n
ldg1
# find /devices/virtual-devices@100 -type c -name network@2\*
/devices/virtual-devices@100/channel-devices@200/network@2:vnet1
#
```

ネットワークインタフェース名は、コロンのあとのエントリの部分で、この場合は vnet1 です。

3. vnet1 を plumb して、手順 1 の net-c に対する ldm list -l の出力で示されたように、MAC アドレスが 00:14:4f:f8:dd:68 であることを確認します。

```
# 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
#
```

---

## 自動または手動による MAC アドレスの割り当て

使用する予定の論理ドメイン、仮想スイッチ、および仮想ネットワークに割り当てられるだけの十分な数のメディアアクセス制御 (MAC) アドレスが必要です。Logical Domains Manager によって論理ドメイン、仮想ネットワーク (vnet)、および仮想スイッチ (vswitch) に自動的に MAC アドレスを割り当てるか、割り当てられた MAC アドレスの自身のプールから手動で MAC アドレスを割り当てることができます。MAC アドレスを設定する ldm のサブコマンドは、add-domain、add-vsw、set-vsw、add-vnet、および set-vnet です。これらのサブコマンドで MAC アドレスを指定しない場合は、Logical Domains Manager が自動的に MAC アドレスを割り当てます。

Logical Domains Manager に MAC アドレスの割り当てを実行させる利点は、論理ドメインで使用するための専用の MAC アドレスのブロックを利用できることです。また、Logical Domains Manager は、同じサブネットにあるほかの Logical Domains Manager インスタンスと競合する MAC アドレスを検出し、これを回避します。これにより、手動で MAC アドレスのプールを管理する必要がなくなります。

論理ドメインが作成されたり、ドメインにネットワークデバイスが構成されたりするとすぐに、MAC アドレスの割り当てが発生します。また、割り当ては、デバイスまたは論理ドメイン自体が削除されるまで保持されます。

この節では、次の項目について説明します。

- [74 ページの「Logical Domains ソフトウェアに割り当てられる MAC アドレスの範囲」](#)
- [74 ページの「自動割り当てのアルゴリズム」](#)
- [75 ページの「重複した MAC アドレスの検出」](#)
- [76 ページの「解放された MAC アドレス」](#)
- [76 ページの「手動による MAC アドレスの割り当て」](#)

## Logical Domains ソフトウェアに割り当てられる MAC アドレスの範囲

論理ドメインには、次の 512K の MAC アドレスのブロックが割り当てられています。

00:14:4F:F8:00:00 ~ 00:14:4F:FF:FF:FF

下位の 256K のアドレスは、Logical Domains Manager による MAC アドレスの自動割り当てに使用されるため、この範囲のアドレスを手動で要求することはできません。

00:14:4F:F8:00:00 ~ 00:14:4F:FB:FF:FF

MAC アドレスを手動で割り当てる場合は、この範囲の上位半分を使用できます。

00:14:4F:FC:00:00 ~ 00:14:4F:FF:FF:FF

## 自動割り当てのアルゴリズム

論理ドメインまたはネットワークデバイスの作成時に MAC アドレスを指定しない場合、Logical Domains Manager は MAC アドレスを自動的に確保して、その論理ドメインまたはネットワークデバイスに割り当てます。この MAC アドレスを取得するために、Logical Domains Manager はアドレスの選択を繰り返し試みて、潜在的な競合がないか確認します。

可能性のあるアドレスを選択する前に、Logical Domains Manager は、自動的に割り当てられ、最近解放されたアドレスが、ここで使用するためにデータベースに保存されているかどうかをまず確認します ([76 ページの「解放された MAC アドレス」](#)を参照)。保存されていた場合、Logical Domains Manager はデータベースから候補となるアドレスを選択します。



最近解放されたアドレスが使用できない場合、MAC アドレスはこの用途のために確保された 256K の範囲のアドレスからランダムに選択されます。候補として選択される MAC アドレスが重複する可能性を少なくするために、MAC アドレスはランダムに選択されます。

選択されたアドレスは、ほかのシステムのその他の Logical Domains Manager に対して確認され、重複した MAC アドレスが実際に割り当てられることを防止します。使用されるアルゴリズムは、75 ページの「重複した MAC アドレスの検出」に記載されています。アドレスがすでに割り当てられている場合、Logical Domains Manager は、ほかのアドレスの選択および競合の再確認を繰り返し行います。この動作は、まだ割り当てられていない MAC アドレスが見つかるか、30 秒の制限時間が経過するまで続きます。制限時間に達すると、デバイスの作成が失敗し、次のようなエラーメッセージが表示されます。

```
Automatic MAC allocation failed.Please set the vnet MAC address manually.
```

## 重複した MAC アドレスの検出

同じ MAC アドレスが別のデバイスに割り当てられないようにするために、Logical Domains Manager がデバイスに割り当てようとしているアドレスを含むマルチキャストメッセージを、制御ドメインのデフォルトのネットワークインタフェースを介して送信することで、Logical Domains Manager はほかのシステム上の Logical Domains Manager に確認します。MAC アドレスの割り当てを試行している Logical Domains Manager は、応答が返されるまで 1 秒待機します。LDoms が有効な別のシステムの異なるデバイスにその MAC アドレスがすでに割り当てられている場合は、そのシステムの Logical Domains Manager が対象となっている MAC アドレスを含む応答を送信します。要求を送信した Logical Domains Manager は応答を受け取ると、選択した MAC アドレスがすでに割り当てられていることを認識し、別のアドレスを選択して処理を繰り返します。

デフォルトでは、これらのマルチキャストメッセージは、デフォルトの生存期間 (TTL) が 1 である同じサブネット上のほかのマネージャーにのみ送信されます。TTL は、サービス管理機能 (SMF) プロパティ `1dmd/hops` を使用して設定できます。

各 Logical Domains Manager は、次の処理を行います。

- マルチキャストメッセージの待機
- ドメインに割り当てられた MAC アドレスの追跡
- 重複の検索
- 重複が発生しないようにするための応答

何らかの理由でシステム上の Logical Domains Manager が停止すると、Logical Domains Manager が停止している間に MAC アドレスの重複が発生する可能性があります。

論理ドメインまたはネットワークデバイスが作成されるときに MAC の自動割り当てが行われ、そのデバイスまたは論理ドメインが削除されるまで保持されます。

## 解放された MAC アドレス

自動の MAC アドレスに関連付けられた論理ドメインまたはデバイスが削除されると、その MAC アドレスはそのシステムであとで使用する場合に備えて、最近解放された MAC アドレスのデータベースに保存されます。これらの MAC アドレスを保存して、動的ホスト構成プロトコル (DHCP) サーバーのインターネットプロトコル (IP) アドレスが使い果たされないようにします。DHCP サーバーが IP アドレスを割り当てるとき、しばらくの間 (リース期間中) その動作が行われます。多くの場合、リース期間は非常に長く構成されており、通常は数時間または数日間です。ネットワークデバイスが作成および削除される割合が高く、Logical Domains Manager が自動的に割り当てられた MAC アドレスを再利用しない場合、割り当てられる MAC アドレスの数によって典型的な構成の DHCP サーバーがすぐに圧迫される可能性があります。

Logical Domains Manager は、論理ドメインまたはネットワークデバイスの MAC アドレスを自動的に取得するように要求されると、以前に割り当てられた再利用可能な MAC アドレスが存在するかどうかを確認するために、解放された MAC アドレスデータベースを最初に参照します。このデータベースに使用可能な MAC アドレスが存在する場合、重複した MAC アドレスの検出アルゴリズムが実行されます。以前に解放された MAC アドレスが、そのあと割り当てられていない場合は、その MAC アドレスが再利用され、データベースから削除されます。競合が検出された場合、そのアドレスは単にデータベースから削除されます。Logical Domains Manager は、データベース内の次のアドレスを試行するか、使用可能なアドレスがない場合は、新しい MAC アドレスをランダムに選択します。

## 手動による MAC アドレスの割り当て

次の手順では、手動による MAC アドレスの作成方法について説明します。

### ▼ 手動で MAC アドレスを割り当てる

1. 物理ホストの IP アドレスのサブネットの部分を 16 進形式に変換し、その結果を保存します。

```
# grep hostname /etc/hosts | awk '{print $1}' | awk -F. '{printf("%x", $4)}'
```

27

2. 制御ドメインを除き、存在するドメインの数を確認します。

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

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv	SP	4	768M	0.3%	4h 54m
myldom1	active	-n---	5000	2	512M	1.9%	1h 12m

ゲストドメインが1つあります。作成するドメインを含める必要があるため、ドメインの数は2になります。

3. ベンダー文字列 (0x08020ab) に変換した IP アドレス (27) を追加し、そのあとに論理ドメインの数 (この例では 2) と 10 を加えた数である 12 を追加します。

```
0x08020ab and 27 and 12 = 0x08020ab2712 or 8:0:20:ab:27:12
```

---

## CPU およびメモリーアドレスのマッピング

Solaris の障害管理アーキテクチャー (FMA) では、物理 CPU 番号に関する CPU エラーと、物理メモリーアドレスに関するメモリーエラーを報告します。

エラーが発生した論理ドメインと、そのドメイン内の対応する仮想 CPU 番号または実メモリーアドレスを確認する場合は、マッピングを実行する必要があります。

### CPU マッピング

ドメインとそのドメイン内の仮想 CPU 番号は、特定の物理 CPU 番号に対応しており、次の手順を使用して確認できます。

#### ▼ CPU 番号を確認する

1. すべてのドメインの解析可能な長形式のリストを生成します。

```
primary$ ldm ls -l -p
```

2. リストの VCPU セクションで、物理 CPU 番号に等しい pid フィールドを持つエントリを探します。

- このようなエントリが見つかった場合、CPU はそのエントリが表示されたドメインに存在し、そのドメイン内の仮想 CPU 番号がエントリの vid フィールドに指定されています。
- このようなエントリが見つからない場合、CPU はどのドメインにも存在しません。

## メモリーのマッピング

ドメインとそのドメイン内の実メモリーアドレスは、特定の物理メモリーアドレス (PA) に対応しており、次のように確認できます。

### ▼ 実メモリーアドレスを確認する

1. すべてのドメインの解析可能な長形式のリストを生成します。

```
primary$ ldm ls -l -p
```

2. リストの MEMORY セクションの行を探します。この場合、PA は  $pa$  から  $(pa + size - 1)$  の包括範囲内にあります。つまり、 $pa \leq PA < (pa + size - 1)$  です。
 

ここでの  $pa$  と  $size$  は、その行の対応するフィールドの値を指します。

  - このようなエントリが見つかった場合、PA はそのエントリが表示されたドメインに存在し、そのドメイン内の対応する実アドレスが  $ra + (PA - pa)$  によって求められます。
  - このようなエントリが見つからない場合、PA はどのドメインにも存在しません。

## CPU およびメモリーのマッピングの例

コード例 5-16 に示すような論理ドメインの構成があり、物理 CPU 番号 5 に対応するドメインと仮想 CPU、および物理アドレス 0x7e816000 に対応するドメインと実アドレスを確認すると仮定します。

リストで pid フィールドが 5 である VCPU エントリを探すと、論理ドメイン ldg1 の下に次のエントリが見つかります。

したがって、物理 CPU 番号 5 はドメイン ldg1 に存在し、そのドメイン内には仮想 CPU 番号 1 があります。

```
|vid=1|pid=5|util=29|strand=100
```

リストの MEMORY エントリを探すと、ドメイン ldg2 の下に次のエントリが見つかります。

```
ra=0x8000000|pa=0x78000000|size=1073741824
```

この場合、 $0x78000000 \leq 0x7e816000 \leq (0x78000000 + 1073741824 - 1)$ 、つまり、 $pa \leq PA \leq (pa + size - 1)$  となります。

したがって、PA はドメイン ldg2 にあり、対応する実アドレスは  $0x8000000 + (0x7e816000 - 0x78000000) = 0xe816000$  です。

#### コード例 5-16 論理ドメイン構成の解析可能な長形式のリスト

```
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=0x80000000|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
    
```

## 複数の論理ドメインを使用するための分割 PCI Express バスの構成

**注** – Sun SPARC Enterprise T5120 および T5220 サーバーなどの Sun UltraSPARC T-2 ベースのサーバーの場合は、この手順を使用せずに、論理ドメインにはネットワークインタフェースユニット (NIU) を割り当てます。

Sun UltraSPARC T1 ベースのサーバーの PCI Express (PCI-E) バスは、さまざまなリーフデバイスが接続される 2 つのポートで構成されます。これらは、pci@780 (bus\_a) および pci@7c0 (bus\_b) という名前でサーバーで識別されます。マルチドメイン環境では、Logical Domains Manager を使用して各リーフに個別のドメインを割り当てるように、PCI-E バスをプログラムすることができます。つまり、I/O の仮想化を使用する代わりに、複数のドメインが物理デバイスへ直接アクセスできるようにすることができます。

Logical Domains システムに電源が入ると、制御 (primary) ドメインはすべての物理デバイスリソースを使用します。このため、primary ドメインが PCI-E バスの両方のリーフを所有しています。



**注意** – サポートされたサーバーの内部ディスクはすべて、1 つのリーフに接続されています。制御ドメインが内部ディスクから起動する場合は、ドメインからそのリーフを削除しないでください。また、主ネットワークのポートを持つリーフを削除していないことを確認してください。制御ドメインまたはサービスドメインから誤ったリーフを削除すると、そのドメインは必要なデバイスにアクセスできず、使用不可になり

ます。主ネットワークのポートがシステムディスク以外の異なるバスにある場合は、ネットワークケーブルをボード上のネットワークポートに移動し、Logical Domains Manager を使用して仮想スイッチ (vsw) を再構成してこの変更を反映してください。

## ▼ 分割 PCI 構成を作成する

ここで示す例は、Sun Fire T2000 サーバーの場合です。この手順は、Sun Fire T1000 サーバーおよび Netra T2000 サーバーなどの Sun UltraSPARC T1 ベースのサーバーにも使用できます。別のサーバーではこれらの手順と若干異なる場合がありますが、この例では基本的な方針について理解できます。ほとんどの場合、起動ディスクを持つリーフを保持したまま、その他のリーフを primary ドメインから削除してほかのドメインに割り当てる必要があります。

1. primary ドメインが PCI Express バスの両方のリーフを所有していることを確認します。

```
primary# ldm list-bindings primary
...
IO
    DEVICE                PSEUDONYM        OPTIONS
    pci@780                bus_a
    pci@7c0                bus_b
...
```

2. 起動ディスクのデバイスパスを確認します。これは保持する必要があります。

```
primary# df /
/                (/dev/dsk/c1t0d0s0 ): 1309384 blocks  457028 files
```

3. ブロック型デバイス c1t0d0s0 が接続されている物理デバイスを確認します。

```
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
```

この例では、ドメイン primary の起動ディスクに対する物理デバイスは、前述の bus\_b に対応する、リーフ pci@7c0 の下にあります。つまり、PCI-Express バスの bus\_a (pci@780) を別のドメインに割り当てることができます。

4. `/etc/path_to_inst` を確認して、ボード上のネットワークポートの物理パスを見つけます。

```
primary# grep e1000g /etc/path_to_inst
```

5. `primary` ドメインから起動ディスク (この例では `pci@780`) を含まないリーフを削除します。

```
primary# ldm remove-io pci@780 primary
```

6. この分割 PCI 構成 (この例では `split-cfg`) をシステムコントローラに追加します。

```
primary# ldm add-config split-cfg
```

また、この構成 (`split-cfg`) は、再起動後に使用される次の構成として設定されます。

---

注 – 現在、SC に保存できる構成数の上限は 8 つです。この数には、`factory-default` 構成は含まれません。

---

7. `primary` ドメインを再起動して、変更を有効にします。

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

8. 直接のアクセスが必要なドメイン (この例では `ldg1`) にリーフ (この例では `pci@780`) を追加します。

```
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.
```

Infiniband カードが構成されていると、`pci@780` バスでバイパスモードの有効化が必要になる場合があります。バイパスモードを有効にする必要があるかどうかについては、[83 ページの「PCI バスでの I/O MMU バイパスモードの有効化」](#)を参照してください。



9. ドメイン ldg1 を再起動して、変更を有効にします。

再起動する場合は、すべてのドメインをアクティブでない状態にする必要があります。このドメインをはじめて構成する場合、ドメインはアクティブではありません。

```
ldg1# shutdown -i6 -g0 -y
```

10. 適切なリーフが primary ドメインに割り当てられたままで、適切なリーフがドメイン ldg1 に割り当てられていることを確認します。

```
primary# ldm list-bindings primary
NAME                STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary             active -n-cv  SP    4     4G      0.4%  18h 25m
...
IO
  DEVICE                PSEUDONYM  OPTIONS
pci@7c0                 bus_b
...
-----
NAME                STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1                active -n---  5000  4     2G      10%   35m
...
IO
  DEVICE                PSEUDONYM  OPTIONS
pci@780                 bus_a
...
```

この出力では、PCI-E リーフ bus\_b とその配下のデバイスがドメイン primary に割り当てられており、bus\_a とそのデバイスが ldg1 に割り当てられていることを確認できます。

## PCI バスでの I/O MMU バイパスモードの有効化

Infiniband ホストチャネルアダプタ (HCA) カードが構成されていると、I/O メモリー管理ユニット (MMU) のバイパスモードをオンにする必要がある場合があります。デフォルトでは、Logical Domains ソフトウェアが PCI-E トランザクションを制御して、特定の I/O デバイスまたは PCI-E オプションが I/O ドメイン内で割り当てられた物理メモリーにのみアクセス可能にします。別のゲストドメインのメモリーにアクセスしようとしても、I/O MMU によって阻止されます。これによって、I/O ドメインとその他すべてのドメインの間でより高いレベルのセキュリティーが得られま

す。ただし、I/O MMU バイパスモードがオフの状態では PCI-E または PCI-X オプションカードが読み込まないまたは動作しないまれな状況では、このオプションを使用して I/O MMU バイパスモードをオンに設定できます。ただし、バイパスモードをオンに設定すると、I/O ドメインからのメモリアクセスのハードウェアによる保護が実行されなくなります。

bypass=on オプションは、I/O MMU バイパスモードをオンに設定します。このバイパスモードは、それぞれの I/O ドメインおよびその I/O ドメイン内の I/O デバイスがすべてのゲストドメインに信頼されている場合にのみ有効にする必要があります。この例では、バイパスモードをオンにします。

```
primary# ldm add-io bypass=on pci@780 ldg1
```

出力では、OPTIONS の下に bypass=on が表示されます。

---

## コンソールグループの使用

仮想ネットワーク端末サーバーデーモン vntsd(1M) を使用すると、1 つの TCP ポートを使用して複数のドメインのコンソールにアクセスできるようになります。

Logical Domains Manager は、ドメインの作成時に、そのドメインのコンソール用の新しいデフォルトグループを作成することにより、各コンソールに一意的 TCP ポートを割り当てます。TCP ポートは、コンソール自体ではなくコンソールグループに割り当てられます。コンソールは、set-vcons サブコマンドを使用して既存のグループにバインドできます。

### ▼ 複数のコンソールを 1 つのグループにまとめる

1. ドメインのコンソールを 1 つのグループにバインドします。

次の例では、3 つの異なるドメイン (ldg1、ldg2、ldg3) のコンソールを同じコンソールグループ (group1) にバインドします。

```
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. 関連付けられた TCP ポート (この例ではポート 5000 の localhost) に接続します。

```
# telnet localhost 5000
primary-vnts-group1: h, l, c{id}, n{name}, q:
```

いずれかのドメインコンソールの選択を求めるプロンプトが表示されます。

3. 1 (list) を選択して、グループ内のドメインを一覧表示します。

```
primary-vnts-group1: h, l, c{id}, n{name}, q: 1
DOMAIN ID          DOMAIN NAME        DOMAIN STATE
0                   ldg1               online
1                   ldg2               online
2                   ldg3               online
```

---

注 - コンソールを別のグループまたは vcc インスタンスに再度割り当てるには、ドメインがバインドされていない状態、つまり、アクティブでない状態である必要があります。vntsd を管理するための SMF の構成と使用法、およびコンソールグループの使用法については、Solaris 10 OS の vntsd(1M) マニュアルページを参照してください。

---

---

## サーバー間での論理ドメインの移動

動作中でない論理ドメインを、あるサーバーから別のサーバーに移動できます。ドメインを移動する前に、2 つのサーバーに同じドメインを設定するとドメインを移動しやすくなります。実際にはドメイン自体を移動する必要はありません。一方のサーバーでドメインのバインドを解除してドメインを停止し、もう一方のサーバーでドメインをバインドして起動する必要があるだけです。

### ▼ 移動するドメインを設定する

1. 2 つのサーバーに同じ名前のドメインを作成します。たとえば、serverA と serverB に domainA1 を作成します。
2. 両方のサーバーに、仮想ディスクサーバーデバイスと仮想ディスクを追加します。仮想ディスクサーバーは、バインドの一部としてエクスポート用の基本となるデバイスを開きます。

3. 一方のサーバー (たとえば `serverA`) のドメインのみをバインドします。もう一方のサーバーのドメインはアクティブでない状態のままにしておきます。

## ▼ ドメインを移動する

1. `serverA` のドメインのバインドを解除して、ドメインを停止します。
2. `serverB` のドメインをバインドし、起動します。

ドメインのバインド

---

注 - ドメインをバインドするまで、リソースは使用されません。

---

## 論理ドメインの削除

この節では、すべてのゲストドメインを削除し、サーバー全体を制御する 1 つの OS インスタンスに戻す方法について説明します。

## ▼ すべてのゲスト論理ドメインを削除する

1. システムコントローラのすべての論理ドメイン構成を一覧表示します。

```
primary# ldm ls-config
```

2. 以前システムコントローラ (SC) に保存されたすべての構成 (`config_name`) を削除します。各構成に対して次のコマンドを使用します。

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

以前に SC に保存されたすべての構成を削除すると、`factory-default` ドメインは、制御ドメイン (`primary`) が再起動されるときに使用される次のドメインになります。

3. `-a` オプションを使用して、すべてのゲストドメインを停止します。

```
primary# ldm stop-domain -a
```

- すべてのドメインを一覧表示して、ゲストドメインに接続されたすべてのリソースを確認します。

```
primary# ldm ls
```

- ゲストドメインに接続されたすべてのリソースを解放します。この処理を行うには、システムに構成された各ゲストドメイン (*ldom*) に対して `ldm unbind-domain` コマンドを使用します。

---

注 – 分割 PCI 構成では、制御ドメインが必要とするサービスを I/O ドメインが提供している場合、その I/O ドメインのバインドを解除できないことがあります。この場合は、この手順をスキップします。

---

```
primary# ldm unbind-domain ldom
```

- 制御ドメインを停止します。

```
primary# shutdown -i1 -g0 -y
```

- `factory-default` 構成が再読み込みされるように、システムコントローラの電源を切ってすぐに入れ直します。

```
sc> poweroff  
sc> poweron
```

---

## 論理ドメインを使用した Solaris OS の操作

この節では、Logical Domains Manager によって作成された構成がインスタンス化される時、つまり、ドメイン化が有効になるときに発生する、Solaris OS を使用した場合の動作の変更について説明します。

---

注 – ドメイン化が有効かどうかに関する説明は、Sun UltraSPARC T1 ベースのプラットフォームにのみ関連するものです。それ以外のプラットフォームでは、ドメイン化は常に有効になっています。

---

## ドメイン化を有効にした場合、Solaris OS の起動後に OpenBoot ファームウェアを使用できない

ドメイン化が有効な場合には、Solaris OS を起動したあとに OpenBoot ファームウェアを使用できません。これは、OpenBoot ファームウェアがメモリーから削除されるためです。

Solaris OS から ok プロンプトを表示するには、ドメインを停止する必要があります。Solaris OS の halt コマンドを使用すると、ドメインを停止することができます。

## サーバーの電源の再投入

LDoms ソフトウェアを実行しているシステムでサーバーの電源の再投入を必要とする保守作業を行うときは常に、最初に現在の論理ドメイン構成を SC に保存する必要があります。

### ▼ 現在の論理ドメイン構成を SC に保存する

- 次のコマンドを使用します。

```
# ldm add-config config_name
```

## OpenBoot power-off コマンドの結果

OpenBoot™ power-off コマンドでは、システムの電源が切断されません。OpenBoot ファームウェアを使用しながらシステムの電源を切断するには、システムコントローラまたはシステムプロセッサの poweroff コマンドを使用します。OpenBoot power-off コマンドでは、次のようなメッセージが表示されます。

```
NOTICE: power-off command is not supported, use appropriate  
NOTICE: command on System Controller to turn power off.
```

## Solaris OS のブレークの結果

ドメイン化が有効でない場合にブレークが実行されると、通常、Solaris OS は OpenBoot プロンプトに移行します。この節で説明する動作は、次の 2 つの状況で発生します。

1. 入力デバイスが keyboard に設定されているときに、L1-A キーシーケンスを押した場合。
2. 仮想コンソールが telnet プロンプトにあるときに、send break コマンドを入力した場合。

ドメイン化が有効な場合は、これらのタイプのブレイク後に次のプロンプトが表示されます。

```
c)ontinue, s)ync, r)eboot, h)alt?
```

これらのタイプのブレイク後のシステムの動作を表す文字を入力します。

## 制御ドメインの停止または再起動の結果

次の表に、制御 (primary) ドメインの停止時または再起動時に予想される動作を示します。

**注** – 表 5-1 のドメイン化が有効かどうかに関する質問は、Sun UltraSPARC T1 プロセッサにのみ関連するものです。それ以外のプラットフォームでは、ドメイン化は常に有効になっています。

表 5-1 制御 (primary) ドメインの停止または再起動時に予想される動作

コマンド	ドメイン化が有効か	他のドメインが構成されているか	動作
halt	無効	なし	Sun UltraSPARC T1 プロセッサの場合: ok プロンプトに移行します。
	有効	いいえ	Sun UltraSPARC T1 プロセッサの場合: システムによってリセットが行われるか、OpenBoot の ok プロンプトに移行するか、または次のプロンプトに移行します。 r) reboot, o) k prompt, or h) alt?
	有効	はい	Sun UltraSPARC T2 プロセッサの場合: ホストの電源が切断され、SC で電源が投入されるまで切断されたままです。  変数 auto-boot? が true である場合は、ソフトリセットが行われて起動します。変数 auto-boot? が false である場合は、ソフトリセットが行われて ok プロンプトで停止します。

表 5-1 制御 (primary) ドメインの停止または再起動時に予想される動作 (続き)

コマンド	ドメイン化が有効か	他のドメインが構成されているか	動作
reboot	無効	なし	Sun UltraSPARC T1 プロセッサの場合: ホストの電源が切断され、再投入されます。
	有効	いいえ	Sun UltraSPARC T1 プロセッサの場合: ホストの電源が切断され、再投入されます。 Sun UltraSPARC T2 プロセッサの場合: ホストを再起動し、電源は切断されません。
	有効	はい	Sun UltraSPARC T1 プロセッサの場合: ホストの電源が切断され、再投入されます。 Sun UltraSPARC T2 プロセッサの場合: ホストを再起動し、電源は切断されません。
shutdown -i 5	無効	なし	Sun UltraSPARC T1 プロセッサの場合: ホストの電源が切断されます。
	有効	いいえ	ホストの電源が切断され、SC で電源が投入されるまで切断されたままです。
	有効	はい	ソフトリセットが行われて再起動します。

## 仮想ディスクで一部の format(1M) コマンドオプションが機能しない

仮想ディスクを使用するゲストドメインでは、Solaris OS の format(1M) コマンドが機能しません。

- label、verify、inquiry などの一部のサブコマンドを仮想ディスクで使用すると失敗します。
- format(1M) コマンドで次のようなメッセージが表示される場合があります。
  - Inquiry failed
  - Disk unformatted
  - Current disk is unformatted
  - Drive type unknown
- 拡張ファームウェアインタフェース (EFI) のディスクラベルを持つ仮想ディスクを選択すると、format(1M) コマンドがクラッシュします。
- ゲストドメインで format(1M) コマンドを実行すると、すべての仮想ディスクが正常にフォーマットされ有効なディスクラベルを持っている場合でも、これらのディスクはアンフォーマットとして表示されます。



仮想ディスクのボリューム構成テーブル (VTOC) を取得または設定する場合は、`format(1M)` コマンドの代わりに、`prtvtoc(1M)` コマンドおよび `fmthard(1M)` コマンドを使用してください。また、実ディスクのサービスドメインからは `format(1M)` コマンドを使用できます。

---

## LDoms と ALOM CMT の使用

この節では、Advanced Lights Out Manager (ALOM) チップマルチスレッディング (CMT) を Logical Domains Manager とともに使用する場合の注意事項について説明します。ALOM CMT ソフトウェアの使用については、『Advanced Lights Out Management (ALOM) CMT v1.3 ガイド』を参照してください。



---

**注意** – ALOM CMT のマニュアルでは 1 つのドメインについて説明しているため、Logical Domains Manager では複数のドメインを導入していることに注意する必要があります。論理ドメインが再起動されると、ゲストドメインの I/O サービスは、制御ドメインが再起動されるまで使用できなくなる場合があります。これは、Logical Domains Manager 1.0.2 ソフトウェアでは制御ドメインがサービスドメインとして機能するためです。再起動処理の間は、ゲストドメインが動かなくなっているように見えます。制御ドメインが完全に再起動すると、ゲストドメインは通常の操作を再開します。サーバー全体の電源が切断される場合は、ゲストドメインの停止のみが必要になります。

---

既存の ALOM CMT コマンドでは、追加オプションが使用可能です。

```
bootmode [normal|reset_nvram|bootscript=strong|config="config-name"]
```

`config="config-name"` オプションを使用すると、次の電源投入時の構成を `factory-default` 出荷時構成などの別の構成に設定できます。

ホストの電源が投入されているか切断されているかにかかわらず、このコマンドを実行できます。次のホストリセットまたは電源投入時に有効になります。

## ▼ 論理ドメインの構成をデフォルトまたは別の構成にリセットする

- ALOM CMT ソフトウェアでこのコマンドを実行して、次の電源投入時に論理ドメインの構成をデフォルトの出荷時構成にリセットします。

```
sc> bootmode config="factory-default"
```

また、`ldm add-config` コマンドを使用して Logical Domains Manager で作成され、システムコントローラ (SC) に保存されているほかの構成を選択することもできます。Logical Domains Manager の `ldm add-config` コマンドで指定した名前を使用して、ALOM CMT の `bootmode` コマンドでその構成を選択できます。たとえば、`ldm-config1` という名前の構成が保存されているとすると、次のように指定します。

```
sc> bootmode config="ldm-config1"
```

`ldm add-config` コマンドの詳細は、`ldm(1M)` マニュアルページまたは『Logical Domains (LDoms) Manager 1.0.1 マニュアルページガイド』を参照してください。

---

## BSM 監査の有効化と使用

Logical Domains Manager では、Solaris OS の基本セキュリティーモジュール (BSM) 監査機能を使用します。BSM 監査は、制御ドメインの処理およびイベントの履歴を調べて、何が発生したかを調べるための手段を提供します。履歴は、何が、いつ、誰によって行われ、どのような影響があるかを示すログに保持されます。

この節では、この監査機能を使用する場合に、有効化、検証、無効化、出力の表示、および監査ログの切り替えを行う方法について説明します。BSM 監査の詳細は、Solaris 10 の『Solaris のシステム管理 (セキュリティーサービス)』で参照できます。

BSM 監査は、次の 2 つのいずれかの方法で有効にできます。監査を無効にする場合は、有効にしたときと同じ方法を使用してください。2 つの方法は次のとおりです。

- Solaris Security Toolkit の `enable-bsm.fin` 終了スクリプトを使用します。  
`enable-bsm.fin` スクリプトは、デフォルトでは `ldm_control-secure.driver` では使用されません。選択したドライバで終了スクリプトを有効にする必要があります。
- Solaris OS の `bsmconv(1M)` コマンドを使用します。

ここでは、両方の方法についての手順を示します。

## ▼ enable-bsm.fin 終了スクリプトを使用する

1. `ldm_control-secure.driver` を `my-ldm.driver` にコピーします。ここで `my-ldm.driver` は `ldm_control-secure.driver` のコピーの名前です。
2. `ldm_control-config.driver` を `my-ldm-config.driver` にコピーします。ここで `my-ldm-config.driver` は `ldm_control-config.driver` のコピーの名前です。
3. `ldm_control-hardening.driver` を `my-ldm-hardening.driver` にコピーします。ここで `my-ldm-hardening.driver` は `ldm_control-hardening.driver` のコピーの名前です。
4. `my-ldm.driver` を編集して、新しい構成と強化ドライバを、それぞれ `my-ldm-control.driver` と `my-ldm-hardening.driver` に変更します。
5. `my-ldm-hardening.driver` を編集して、ドライバの次の行の先頭にあるハッシュ記号 (#) を削除します。

```
enable-bsm.fin
```

6. `my-ldm.driver` を実行します。

```
# /opt/SUNWjass/bin/jass-execute -d my-ldm.driver
```

7. Solaris OS を再起動して、監査を有効にします。

## ▼ Solaris OS の bsmconv(1M) コマンドを使用する

1. `/etc/security/audit_control` ファイルの `flags:` 行に `vs` を追加します。
2. `bsmconv(1M)` コマンドを実行します。

```
# /etc/security/bsmconv
```

このコマンドの詳細は、Solaris 10 Reference Manual Collection またはマニュアルページを参照してください。

3. Solaris オペレーティングシステムを再起動して、監査を有効にします。

## ▼ BSM 監査が有効であることを確認する

1. 次のコマンドを入力します。

```
# auditconfig -getcond
```

2. 出力に `audit condition = auditing` が表示されていることを確認します。

## ▼ 監査を無効にする

監査を有効にした方法に応じて、次の 2 つのいずれかの方法で監査を無効にすることができます。92 ページの「BSM 監査の有効化と使用」を参照してください。

1. 次のいずれかを実行します。

- BSM 監査を有効にした Solaris Security Toolkit による強化の実行を取り消します。

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

- Solaris OS の `bsmunconv(1M)` コマンドを使用します。

```
# /etc/security/bsmunconv
```

2. Solaris OS を再起動して、監査を無効にします。

## ▼ 監査の出力を表示する

- BSM 監査の出力を表示するには、次のいずれかの方法を使用します。
  - Solaris OS コマンドの `auditreduce(1M)` と `praudit(1M)` を使用して、監査の出力を表示します。次に例を示します。

```
# auditreduce -c vs | praudit
# auditreduce -c vs -a 20060502000000 | praudit
```

- Solaris OS の `praudit -x` コマンドを使用して、XML 出力を表示します。

## ▼ 監査ログを切り替える

- Solaris OS の `audit -n` コマンドを使用して、監査ログを切り替えます。

---

## NAT およびルーティング用の仮想スイッチおよびサービスドメインの構成

仮想スイッチ (vswitch) はレイヤー 2 スイッチで、サービスドメインでネットワークデバイスとしても使用できます。仮想スイッチは、さまざまな論理ドメインで仮想ネットワーク (vnet) デバイス間のスイッチとしてのみ動作するように構成できますが、物理デバイスを介してネットワークの外部に接続することはできません。このモードで、vswitch をネットワークデバイスとして `plumb` し、サービスドメインで IP ルーティングを有効にすると、仮想ネットワークでサービスドメインをルーターとして使用して外部と通信することができます。このモードでの操作は、物理ネットワークアダプタが GLDv3 に準拠していない場合、ドメインが外部に接続できるようにするために非常に重要です。

この構成の利点は次のとおりです。

- 仮想スイッチは物理デバイスを直接使用する必要がなく、基本となるデバイスが GLDv3 に準拠していない場合でも外部と接続できます。
- この構成では、Solaris OS の IP ルーティングとフィルタリング機能を利用できます。

## ▼ ドメインが外部に接続できるように仮想スイッチを設定する

1. 物理デバイスに関連付けずに仮想スイッチを作成します。

アドレスを割り当てる場合は、仮想スイッチに一意の MAC アドレスが割り当てられるようにしてください。

```
primary# ldm add-vsw [mac-addr=xx:xx:xx:xx:xx:xx] primary-vsw0 primary
```

2. ドメインによって使用される物理ネットワークデバイスに加えて、仮想スイッチをネットワークデバイスとして `plumb` します。

仮想スイッチの `plumb` の詳細は、47 ページの「仮想スイッチを主インタフェースとして構成する」を参照してください。

3. 必要に応じて、DHCP で仮想スイッチデバイスを構成します。  
DHCP での仮想スイッチデバイスの構成については、[47 ページ](#)の「[仮想スイッチを主インタフェースとして構成する](#)」を参照してください。
4. 必要に応じて、`/etc/dhcp.vsw` ファイルを作成します。
5. サービスドメインで IP ルーティングを構成し、すべてのドメインで必要なルーティングテーブルを設定します。  
この実行方法については、Solaris Express System Administrator Collection の『[System Administration Guide: IP Services](#)』の第 5 章「[Configuring TCP/IP Network Services and Ipv4 Addressing \(Tasks\)](#)」の「[Packet Forwarding and Routing on Ipv4 Networks](#)」の節を参照してください。

---

## 仮想ディスクと ZFS の使用

この節では、論理ドメインでの仮想ディスクと Zettabyte File System (ZFS) の使用について、次の項目で説明します。

- [96 ページ](#)の「[ZFS ボリュームでの仮想ディスクの作成](#)」
- [98 ページ](#)の「[仮想ディスクでの ZFS の使用](#)」
- [99 ページ](#)の「[起動ディスクとしての ZFS の使用](#)」

### ZFS ボリュームでの仮想ディスクの作成

次の手順では、サービスドメインで ZFS ボリュームを作成し、そのボリュームをほかのドメインで仮想ディスクとして使用可能にする方法について説明します。この例では、サービスドメインは制御ドメインと同じで、`primary` という名前です。ゲストドメインは、例として `ldg1` という名前が付いています。各手順のプロンプトは、コマンドを実行するドメインを示します。

#### ▼ ZFS ボリュームで仮想ディスクを作成する

1. ZFS ストレージプール (`zpool`) を作成します。

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

2. ZFS ボリュームを作成します。

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

3. zpool (この例では tank1) と ZFS ボリューム (この例では tank/myvol) が作成されたことを確認します。

```
primary# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
tank1                               100M  43.0G  24.5K  /tank1
tank1/myvol                          22.5K  43.1G  22.5K  -
```

4. tank1/myvol を仮想ディスクとしてエクスポートするサービスを構成します。

```
primary# ldm add-vdsdev /dev/zvol/rdisk/tank1/myvol zvol@primary-vds0
```

5. エクスポートしたディスクを別のドメイン (この例では ldg1) に追加します。

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

6. ほかのドメイン (この例では ldg1) で、ドメインを起動して、新しい仮想ディスクが認識されていることを確認します。devfsadm コマンドの実行が必要になる場合があります。

この例では、新しいディスクは /dev/rdisk/c2d2s0 として表示されます。

```
ldg1# newfs /dev/rdisk/c2d2s0
newfs: construct a new file system /dev/rdisk/c2d2s0: (y/n)? y
Warning: 4096 sector(s) in last cylinder unallocated
Warning: 4096 sector(s) in last cylinder unallocated
/dev/rdisk/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
```

---

**注** – ZFS ボリュームは、仮想ディスクスライスとして論理ドメインにエクスポートされます。そのため、format コマンドを使用したり、Solaris OS を zvol でサポートされた仮想ディスクにインストールしたりすることはできません。

---

## 仮想ディスクでの ZFS の使用

次の手順では、仮想ディスク上の ZFS をドメインから直接使用する方法について説明します。Solaris 10 OS の `zpool(1M)` および `zfs(1M)` コマンドを使用して、仮想ディスクに ZFS プール、ファイルシステム、およびボリュームを作成できます。ストレージバックエンドは異なりますが (物理ディスクではなく仮想ディスク)、ZFS の使用方法に変更はありません。

また、既存の ZFS ファイルシステムが存在する場合は、サービルドメインからこれをエクスポートして他のドメインで使用することができます。

この例では、サービルドメインは制御ドメインと同じで、`primary` という名前です。ゲストドメインは、例として `ldg1` という名前が付いています。各手順のプロンプトは、コマンドを実行するドメインを示します。

### ▼ 仮想ディスクで ZFS を使用する

1. ZFS プール (この例では `tank`) を作成してから、作成されたことを確認します。

```
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. ZFS ファイルシステム (この例では `tank/test`) を作成してから、作成されたことを確認します。

この例では、サービルドメインで次のコマンドを実行することにより、ファイルシステムがディスク `c2t42d0` 上に作成されます。

```
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. ZFS プール (この例では `tank`) をエクスポートします。

```
primary# zpool export tank
```

4. 物理ディスク `c2t42d0s2` を仮想ディスクとしてエクスポートするサービスを構成します。

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



5. エクスポートしたディスクを別のドメイン (この例では ldg1) に追加します。

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

6. ほかのドメイン (この例では ldg1) で、ドメインを起動して、新しい仮想ディスクが認識されていることを確認します。devfsadm コマンドの実行が必要になる場合があります。そのあと、ZFS プールをインポートします。

```
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
```

これで、ZFS プール (この例では tank/test) がインポートされ、ドメイン ldg1 で使用可能になりました。

## 起動ディスクとしての ZFS の使用

論理ドメインでは、仮想ディスクとしての大規模ファイルが構成された ZFS ファイルシステムを使用できます。

---

**注** – サービスドメインでは、ZFS ファイルシステムにより多くのメモリーが必要です。サービスドメインの構成時には、この点を考慮してください。

---

ZFS では次のことを実行できます。

- ファイルシステムをすばやく複製する
- 複製を使用して、ほかのドメインをプロビジョニングする
- ファイル上のディスクや ZFS ファイルシステム内のファイルにネットインストールを行う

## ▼ 起動ディスクとして ZFS を使用する

次の手順を使用して、論理ドメインの ZFS ディスクを作成できます。また、ほかのドメインの ZFS ディスクのスナップショットや複製を作成することもできます。

1. primary ドメインで、ZFS プール用の記憶領域として使用するディスク全体またはスライスを予約します。手順 2 では、ディスクのスライス 5 を使用します。
2. ZFS プール (たとえば `ldomspool`) を作成します。

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

3. 最初のドメイン (この例では `ldg1`) に ZFS ファイルシステムを作成します。

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

4. このドメインのディスクとなるファイルを作成します。

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

5. そのファイルを、ドメインの作成時に使用するデバイスとして指定します。

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

6. ドメイン `ldg1` を起動し、`vdisk1` にネットインストールします。このファイルは完全なディスクとして機能し、パーティションを作成できます。つまり、`root`、`usr`、`home`、`dump`、および `swap` 用にパーティションを分割することができます。
7. インストールが完了したら、ファイルシステムのスナップショットを作成します。

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

---

**注** — ドメインを再起動する前にスナップショットを作成すると、スナップショットの一部、またはスナップショットから作成されたその他の複製には、ドメインの状態が保存されません。

---

8. スナップショットから追加の複製を作成して、ほかのドメイン (この例では ldg2 および ldg3) の起動ディスクとして使用します。

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

9. すべてが正常に作成されたことを確認します。

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
ldomspool                            1.07G  2.84G  28.5K  /ldomspool
ldomspool/ldg1                        1.03G  2.84G  1.00G  /ldomspool/ldg
1  ldomspool/ldg1@initial              23.0M      -  1.00G  -
ldomspool/ldg2                        23.2M  2.84G  1.00G  /ldomspool/ldg
2  ldomspool/ldg3                      21.0M  2.84G  1.00G  /ldomspool/ldg
3
```

---

**注** – 複製を作成するために必要な十分な領域が ZFS プールに確保されていることを確認してください。ZFS は書き込み時コピーを使用して、複製のブロックが変更されるときにのみプールの領域を使用します。ドメインの起動後でも、複製はディスクに必要なごくわずかな領域しか使用しません。これは、OS バイナリの大部分が最初のスナップショットのバイナリと同じであるためです。

---

---

## 論理ドメイン環境でのボリュームマネージャーの使用

この節では、次の項目について説明します。

- [102 ページの「ボリュームマネージャーでの仮想ディスクの使用」](#)
- [104 ページの「仮想ディスクでのボリュームマネージャーの使用」](#)

## ボリュームマネージャーでの仮想ディスクの使用

Zettabyte File System (ZFS)、Solaris™ ボリュームマネージャー (SVM)、または Veritas Volume Manager (VxVM) は、サービスマネインからゲストドメインに仮想ディスクとしてエクスポートできます。エクスポートされたボリュームは、1つのスライス (s0) を持つ仮想ディスクとしてゲストドメインに表示されます。

---

**注** – この節の残りの部分では、例として SVM ボリュームを使用します。ただし、説明は ZFS および VxVM ボリュームにも適用されます。

---

たとえば、サービスマネインが SVM ボリューム /dev/md/dsk/d0 を domain1 にエクスポートし、domain1 では仮想ディスクが /dev/dsk/c0d2\* として認識されている場合、domain1 には s0 デバイス、つまり /dev/dsk/c0d2s0 のみが存在します。

ゲストドメインの仮想ディスク (たとえば /dev/dsk/c0d2s0) は関連付けられたボリューム (たとえば /dev/md/dsk/d0) に直接割り当てられ、ゲストドメインからの仮想ディスクに格納されたデータは、メタデータを追加せずに関連付けられたボリュームに直接格納されます。そのためゲストドメインからの仮想ディスクに格納されたデータは、関連付けられたボリュームを介してサービスマネインから直接アクセスすることもできます。

例:

- SVM ボリューム d0 が primary ドメインから domain1 にエクスポートされる場合、domain1 の構成にはいくつかの手順が追加で必要になります。

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

- domain1 がバインドされて起動されると、エクスポートされたボリュームが /dev/dsk/c0d2s0 のように表示され、そのボリュームが使用可能になります。

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

- domain1 が停止してバインドが解除されると、domain1 からの仮想ディスクに格納されたデータは SVM ボリューム d0 を介して primary ドメインから直接アクセスできます。

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

---

注 – このような仮想ディスクは `format(1M)` コマンドでは認識できず、パーティションに分割できません。また、Solaris OS のインストールディスクとしても使用できません。この項目の詳細は、90 ページの「仮想ディスクで一部の `format(1M)` コマンドオプションが機能しない」を参照してください。

---

## SVM での仮想ディスクの使用

RAID またはミラー SVM ボリュームが別のドメインで仮想ディスクとして使用されている場合に、SVM ボリュームのいずれかのコンポーネントで障害が発生すると、`metareplace` コマンドまたはホットスペアを使用した SVM ボリュームの復旧は開始されません。`metastat` コマンドはそのボリュームを再同期化中と判断しますが、再同期化は進行していません。

たとえば、`/dev/md/dsk/d0` は別のドメインに仮想ディスクとしてエクスポートされた RAID SVM ボリュームで、`d0` にはいくつかのホットスペアデバイスが構成されているとします。`d0` のコンポーネントに障害が発生すると、SVM は障害の発生したコンポーネントをホットスペアに交換して、ふたたび SVM ボリュームとの同期をとります。ただし、再同期化は開始されません。ボリュームは再同期化中として報告されますが、再同期化は進行していません。

```
# 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
```

このような状況で再同期化を完了するには、SVM ボリュームを仮想ディスクとして使用しているドメインを停止してバインドを解除する必要があります。そのあと、`metasync` コマンドを使用して、SVM ボリュームを再同期化できます。

```
# metasync d0
```

## VxVM のインストール時の仮想ディスクの使用

Veritas Volume Manager (VxVM) をシステムにインストールする場合は、仮想ディスクとしてエクスポートする物理ディスクまたはパーティションで **Dynamic Multipathing** (動的マルチパス化) (Veritas) が有効でないことを確認する必要があります。有効になっていると、このようなディスクを使用するドメインをバインドする間に `/var/adm/messages` にエラーが出力されます。

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

コマンド `vxdisk list` で出力されるマルチパス化情報を調べると、Veritas DMP が有効であるかどうかを確認できます。次に例を示します。

```
# 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
```

仮想ディスクとしてエクスポートするディスクまたはスライスで Veritas DMP が有効になっている場合は、`vxddmpadm` コマンドを使用して DMP を無効にしてください。次に例を示します。

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

## 仮想ディスクでのボリュームマネージャーの使用

この節では、論理ドメイン環境での次のような状況について説明します。

- [105 ページの「仮想ディスクでの ZFS の使用」](#)
- [105 ページの「仮想ディスクでの SVM の使用」](#)
- [105 ページの「仮想ディスクでの VxVM の使用」](#)

## 仮想ディスクでの ZFS の使用

仮想ディスクは ZFS とともに使用できます。ZFS ストレージプール (zpool) は、この zpool の一部であるすべてのストレージデバイスを認識する任意のドメインにインポートできます。ドメインが、これらのすべてのデバイスを仮想デバイスまたは実デバイスのどちらで認識するかは関係ありません。

## 仮想ディスクでの SVM の使用

仮想ディスクは、SVM ローカルディスクセットで使用できます。たとえば、仮想ディスクは、ローカルディスクセットの SVM メタデータベース (metadb) の格納またはローカルディスクセットでの SVM ボリュームの作成に使用できます。

現在、仮想ディスクはローカルディスクセットとのみ使用できますが、共有ディスクセット (metaset) とは使用できません。仮想ディスクは、SVM 共有ディスクセットに追加できません。仮想ディスクを SVM 共有ディスクに追加しようとすると、次のようなエラーが表示されて失敗します。

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

## 仮想ディスクでの VxVM の使用

現在、VxVM は仮想ディスクでは機能しません。VxVM ソフトウェアは仮想ディスクが構成されたドメインにインストールできますが、VxVM は使用可能な仮想ディスクを認識できません。

---

## 論理ドメイン環境での IPMP の構成

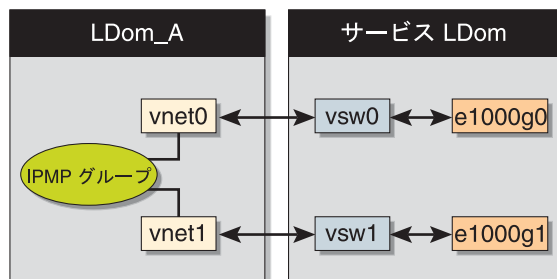
インターネットプロトコルネットワークマルチパス (IPMP) は、複数のネットワークインタフェースカード間の耐障害性と負荷分散を提供します。IPMP を使用すると、1 つ以上のインタフェースを IP マルチパスグループとして構成できます。IPMP を構成すると、システムは IPMP グループ内のインタフェースで障害が発生していないかを自動的に監視します。グループ内のインタフェースに障害が発生したり、保守のために削除されたりすると、IPMP は障害の発生したインタフェースの IP アドレスを自動的に移行して、フェイルオーバーを行います。論理ドメイン環境では、物理ネットワークインタフェースまたは仮想ネットワークインタフェースのいずれかで IPMP を使用したフェイルオーバーを構成できます。

## 論理ドメインの IPMP グループへの仮想ネットワークワークデバイスの構成

IPMP グループに仮想ネットワークワークデバイスを構成することで、論理ドメインに耐障害性を持たせるように構成できます。アクティブ/スタンバイ構成で、仮想ネットワークワークデバイスを使用して IPMP グループを設定する場合は、グループでプロンプベースの検出を使用するようにグループを設定します。Logical Domains 1.0.2 ソフトウェアでは、現在、仮想ネットワークワークデバイスに対するリンクベースの検出とフェイルオーバーはサポートされていません。

次の図に、サービスドメインで個別の仮想スイッチインスタンス (vsw0 および vsw1) に接続された 2 つの仮想ネットワーク (vnet0 および vnet1) を示します。これらは、同様に、2 つの異なる物理インタフェース (e1000g0 および e1000g1) を使用します。物理インタフェースに障害が発生した場合、LDom\_A の IP 層が、プロンプベースの検出を使用して対応する vnet の障害と接続の損失を検出し、vnet の二次デバイスに自動的にフェイルオーバーします。

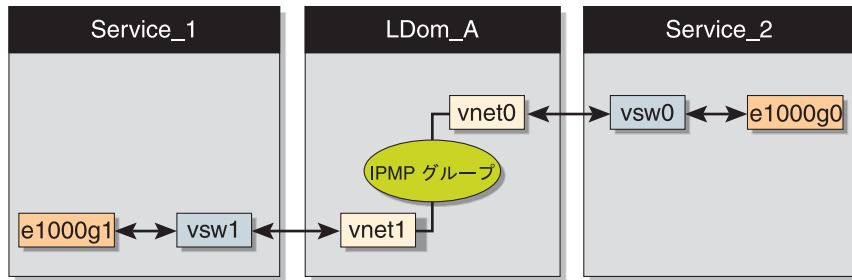
図 5-1 個別の仮想スイッチインスタンスに接続された 2 つの仮想ネットワーク



次の図に示すように、各仮想ネットワークデバイス (vnet0 および vnet1) を異なるサービスドメインの仮想スイッチインスタンスに接続すると、論理ドメインでの信頼性をさらに高めることができます。仮想スイッチインスタンス (vsw0 および vsw1) が構成された 2 つのサービスドメイン (Service\_1 および Service\_2) は、分割 PCI 構成を使用して設定できます。この場合、ネットワークハードウェアの障害に加えて、LDom\_A が仮想ネットワークの障害を検出し、サービスドメインがクラッシュまたは停止したあとでフェイルオーバーを引き起こすことができます。



図 5-2 異なるサービスドメインに接続された各仮想ネットワークデバイス



IPMP グループの構成と使用法の詳細は、Solaris 10 の『Solaris のシステム管理 (IP サービス)』を参照してください。

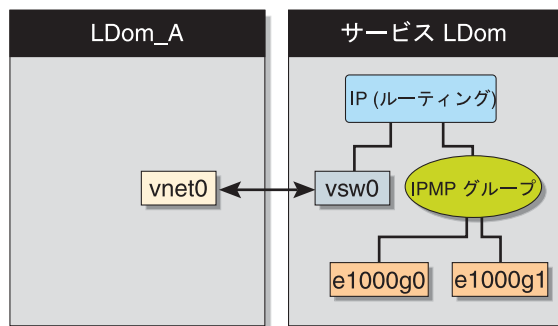
## サービスドメインでの IPMP の構成と使用

サービスドメインの物理インタフェースを IPMP グループとして構成すると、論理ドメイン環境でのネットワーク障害の検出と復旧を設定することもできます。これを行うには、サービスドメインの仮想スイッチをネットワークデバイスとして構成し、サービスドメイン自体を IP ルーターとして動作するように構成します。IP ルーティングの設定については、Solaris 10 の『Solaris のシステム管理 (IP サービス)』を参照してください。

いったん仮想スイッチが構成されると、仮想ネットワークから発生し外部のマシンに送信される予定のすべてのパケットは、物理デバイスを経由して直接送信されるのではなく、IP 層に送信されます。物理インタフェースに障害が発生した場合、IP 層は障害を検出し、自動的に二次インタフェースを使用してパケットをふたたび経路指定します。

物理インタフェースは直接 IPMP グループに構成されているため、グループは、リンクベースまたはプローブベースのいずれかの検出用に設定できます。次の図に、IPMP グループの一部として構成された 2 つのネットワークインタフェース (`e1000g0` および `e1000g1`) を示します。仮想スイッチインスタンス (`vsw0`) は、IP 層にパケットを送信するネットワークデバイスとして `plumb` されています。

図 5-3 IPMP グループの一部として構成された 2 つのネットワークインタフェース



# 用語集

---

この一覧は、Logical Domains 1.0.2 マニュアルで使用される用語、略語、および頭字語を定義したものです。

---

## A

**ALOM CMT** Advanced Lights Out Manager Chip MultiThreading (Advanced Lights Out Manager チップマルチスレッディング)。システムコントローラ上で動作し、CMT サーバーを監視および制御できます。

---

## B

**bge** Broadcom BCM57xx デバイスの Broadcom ギガビット Ethernet ドライバ  
**BSM** Basic Security module (基本セキュリティーモジュール)

---

## C

**CLI** Command-Line Interface (コマンド行インタフェース)  
**CMT** Chip MultiThreading (チップマルチスレッディング)  
**config** システムコントローラに保存されている論理ドメイン構成の名前  
**CPU** Central Processing Unit (中央演算処理装置)

CWQ Control Word Queue の略で、Sun UltraSPARC T2 ベースのプラットフォーム用の暗号化装置

---

## D

- DHCP Dynamic Host Configuration Protocol (動的ホスト構成プロトコル)
- DMP Dynamic MultiPathing (Veritas)
- DR Dynamic Reconfiguration (動的再構成)
- drd(1M) Logical Domains Manager 用の動的再構成デーモン (Solaris 10 OS)
- DS Domain Service module (ドメインサービスモジュール)(Solaris 10 OS)
- 

## E

- e1000g ネットワークインタフェースコントローラの Intel PRO/1000 ギガビットファミリ用のドライバ
- EFI Extensible Firmware Interface (拡張ファームウェアインタフェース)
- ETM Encoding Table Management (エンコーディングテーブル管理) モジュール (Solaris 10 OS)
- 

## F

- FC\_AL Fiber Channel Arbitrated Loop (ファイバチャネル調停ループ)
- FMA Fault Management Architecture (障害管理アーキテクチャー)
- fmd(1M) 障害管理デーモン (Solaris 10 OS)
- FTP File Transfer Protocol (ファイル転送プロトコル)

---

## G

GLDv3 Generic LAN Driver version 3 (汎用 LAN ドライバ version 3)

---

## H

HDD Hard Disk Drive (ハードディスクドライブ)

---

## I

I/O ドメイン 物理 I/O デバイスに対する直接の所有権と直接のアクセス権を持ち、仮想デバイスの形式ではほかの論理ドメインとこれらのデバイスを共有するドメイン

IB InfiniBand

io 内部ディスクおよび PCI-Express (PCI-E) コントローラと、それらに接続されたアダプタやデバイスなどの I/O デバイス

ioctl input/output control call (I/O 制御コール)

IP Internet Protocol (インターネットプロトコル)

IPMP Internet Protocol Network Multipathing (インターネットプロトコルネットワークマルチパス)

---

## K

kaio Kernel Asynchronous Input/Output (カーネル非同期 I/O)

KB KiloByte (K バイト)

KU Kernel Update (カーネル更新)

---

## L

LAN	Local-Area Network (ローカルエリアネットワーク)
LDAP	Lightweight Directory Access Protocol
LDC	Logical Domain Channel (論理ドメインチャネル)
ldm(1M)	Logical Domain Manager ユーティリティー
ldmd	Logical Domains Manager デーモン
Logical Domains (LDoms) Manager	論理ドメインを作成および管理したり、リソースをドメインに割り当てたりするための CLI を提供する

---

## M

MAC	Media Access Control address (メディアアクセス制御アドレス) の略で、LDoms によって自動的に割り当てることも、手動で割り当てることも可能
MAU	Modular Arithmetic Unit の略で、Sun UltraSPARC T1 ベースのプラットフォーム用の暗号化装置
MB	MegaByte (M バイト)
MD	サーバーデータベース内のマシン記述
mem、memory	メモリー単位 – バイト単位でのデフォルトのサイズ。G バイト (G)、K バイト (K)、または M バイト (M) を指定することもできます。ゲストドメインに割り当てることができる、サーバーの仮想化されたメモリーです。
MIB	Management Information Base (管理情報ベース)
MMF	MultiMode Fiber (マルチモードファイバ)
MMU	Memory Management Unit (メモリー管理ユニット)
mtu	Maximum Transmission Unit (最大転送単位)

---

## N

NAT	Network Address Translation (ネットワークアドレス変換)
ndpsldcc	Netra Data Plane Software Logical Domain Channel Client。「vdpsc」も参照してください。
ndpsldcs	Netra Data Plane Software Logical Domain Channel Service。「vdpcs」も参照してください。
NDPSS	Netra Data Plane Software Suite
NFS	Network File System (ネットワークファイルシステム)
NIS	Network Information Service (ネットワーク情報サービス)
NIU	Network Interface Unit (ネットワークインタフェースユニット)(Sun SPARC Enterprise T5120 および T5220 サーバー)
NTS	Network Terminal Server (ネットワーク端末サーバー)
NVRAM	Non-Volatile Random-Access Memory (非揮発性ランダムアクセスメモリー)
nxge	Sun x8 Express 1/10G Ethernet アダプタ用のドライバ

---

## O

OS	Operating System (オペレーティングシステム)
----	---------------------------------

---

## P

PA	Physical Address (物理アドレス)
PCI	Peripheral Component Interconnect バス
PCI-E	PCI Express バス
PCI-X	PCI 拡張バス
PICL	Platform Information and Control Library (プラットフォーム情報とコントロールライブラリ)

picld(1M) PICL デーモン  
PRI PRiority (優先度)

---

## R

RA Real Address (実アドレス)  
RAID Redundant Array of Inexpensive Disks  
RBAC Role-Based Access Control (役割に基づくアクセス制御)  
RPC Remote Procedure Call (遠隔手続き呼び出し)

---

## S

SC System Controller (システムコントローラ) の略で、システムプロセッサと同じ  
SCSI Small Computer System Interface  
SMA System Management Agent (システム管理エージェント)  
SMF Solaris 10 OS のサービス管理機能  
SNMP Simple Network Management Protocol (簡易ネットワーク管理プロトコル)  
SP System Processor (システムプロセッサ) の略で、システムコントローラと同じ  
SSH Secure Shell  
ssh(1) Secure Shell コマンド  
sshd(1M) Secure Shell デーモン  
SunVTS Sun Validation Test Suite  
SVM Solaris Volume Manager (Solaris ボリュームマネージャー)

---

## T

TCP Transmission Control Protocol (伝送制御プロトコル)



---

## U

- UDP User Diagram Protocol (ユーザーダイアグラムプロトコル)
- USB Universal Serial Bus (ユニバーサルシリアルバス)
- UTP Unshielded Twisted Pair (シールドなし・より対線)

---

## V

- vBSC** Virtual Blade System Controller (仮想ブレードシステムコントローラ)
- vcc、vconscon 特定のポート範囲をゲストドメインに割り当てる仮想コンソール端末集配信装置サービス
- vcons、vconsole システムレベルのメッセージにアクセスするための仮想コンソール。接続は、特定のポートで制御ドメイン上の vconscon サービスに接続することによって実現します。
- vcpu **Virtual Central Processing Unit** (仮想中央演算処理装置)。サーバーの各コアは、仮想 CPU として表現されます。たとえば、8 コアの Sun Fire T2000 サーバーには、論理ドメイン間で割り当てることができる 32 の仮想 CPU があります。
- vdc **Virtual Disk Client** (仮想ディスククライアント)
- vdisk 仮想ディスク。さまざまな種類の物理デバイス、ボリューム、またはファイルで構成される総称的なブロック型デバイスです。
- vdppc NDPS 環境における仮想データプレーンチャンネルクライアント
- vdpcs NDPS 環境における仮想データプレーンチャンネルサービス
- vds、vdiskserver 仮想ディスクサーバー。これを使用すると、論理ドメインに仮想ディスクをインポートできます。
- vdsdev、  
vdiskserverdevice 仮想ディスクサーバーデバイス。仮想ディスクサーバーによってエクスポートされます。このデバイスには、ディスク全体、ディスクのスライス、ファイル、またはディスクボリュームを指定できます。
- vnet 仮想ネットワークデバイス。仮想 Ethernet デバイスを実装し、仮想ネットワークスイッチ (vswitch) を使用するシステム内のほかの vnet デバイスと通信します。

---

vntsd(1M)	論理ドメインコンソール用の仮想ネットワーク端末サーバーデーモン (Solaris 10 OS)
vsw、vswitch	仮想ネットワークデバイスを外部ネットワークに接続し、仮想ネットワークデバイス間でのパケットの切り替えも行う仮想ネットワークスイッチ
VTOC	Volume Table Of Contents (ボリューム構成テーブル)
VxVM	Veritas Volume Manager

---

## W

WAN	Wide-Area Network (広域ネットワーク)
-----	------------------------------

---

## X

XFP	eXtreme Fast Path
XML	eXtensible Markup Language

---

## Z

ZFS	Zettabyte File System (Solaris 10 OS)
zpool(1M)	ZFS ストレージプール

---

## か

監査	Solaris OS BSM を使用して、セキュリティの変更元を識別すること
----	--

---

## き

**強化** セキュリティーを向上するために Solaris OS の構成を変更すること

---

## け

**ゲストドメイン** I/O ドメインおよびサービスドメインのサービスを使用し、制御ドメインによって管理されます。

---

## さ

**サービスドメイン** 仮想スイッチ、仮想コンソールコネクタ、仮想ディスクサーバーなどのデバイスをほかの論理ドメインに提供する論理ドメイン

**最小化** 最低限必要な数のコア Solaris OS パッケージをインストールすること

---

## し

**承認** Solaris OS RBAC を使用して承認を設定すること

---

## せ

**制御ドメイン** ほかの論理ドメインおよびサービスを作成および管理するドメイン

**制約** Logical Domains Manager に対する制約とは、特定のドメインに割り当てられる 1 つ以上のリソースのこと。使用可能なリソースに応じて、ドメインに追加するように要求したすべてのリソースを受け取るか、まったく受け取らないかのいずれかです。

---

## て

**適合性** システムの構成が事前に定義されたセキュリティープロファイルに適合しているかどうかを確認すること

---

## は

**ハイパーバイザ** オペレーティングシステムとハードウェア層の間に配置されるファームウェア層

---

## ろ

**論理ドメイン** 1つのコンピュータシステム内で、独自のオペレーティングシステム、リソース、および識別情報を持つ個別の論理グループ