Logical Domains (LDoms) 1.0.3
管理ガイド

Sun Microsystems, Inc.
www.sun.com

Part No. 820-5003-10
2008 年 6 月, Revision A

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

原典: *Logical Domains (LDoms) 1.0.3 Administration Guide*

Part No: 820-4894-10

Revision A

目次

図目次

表目次

コード例

はじめに

『Logical Domains (LDoms) 1.0.3 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 1.0.3 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) 1.0.3 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、Logical Domains ソフトウェアで仮想ディスクを使用する方法について説明します。

第 6 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	*machine-name*%
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.
AaBbCc123	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。	**マシン名% su** Password:
AaBbCc123	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。	rm *filename* 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。	% **grep '^#define ** **XV_VERSION_STRING'**

* 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) 1.0.3 管理ガイド』および『Logical Domains (LDoms) 1.0.3 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) 1.0.3 リリースノート』	820-5009-10	HTML PDF	オンライン
Ldoms の Solaris マニュアルページ	Solaris 10 Reference Manual Collection: • `drd`(1M) マニュアルページ • `vntsd`(1M) マニュアルページ	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 『Logical Domains (LDoms) Manager 1.0.1 マニュアルページガイド』	なし 820-3453-10	SGML PDF	オンライン 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
セキュリティー	『Solaris Security Toolkit 4.2 管理マニュアル』	819-3789-10	HTML PDF	オンライン

用途	タイトル	Part No.	形式	場所
セキュリティー	『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
セキュリティー	『Solaris Security Toolkit 4.2 ご使用にあたって』	819-3796-10	HTML PDF	オンライン
セキュリティー	『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	`http://docs.sun.com`
サポート	`http://jp.sun.com/support`
トレーニング	`http://jp.sun.com/training`

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) 1.0.3 管理ガイド』、Part No. 820-5003-10

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) 1.0.3 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。

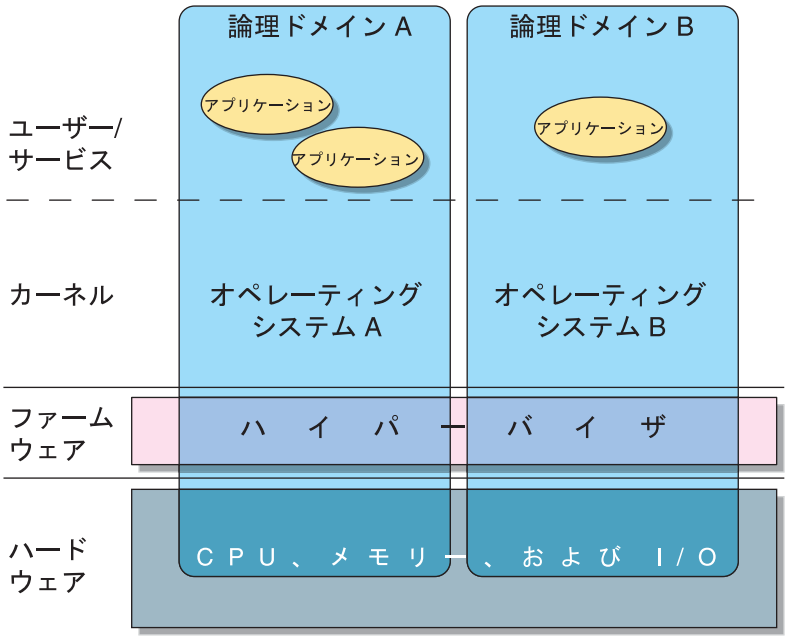
論理ドメインは、独自のオペレーティングシステム、リソース、および単一のコンピュータシステム内での識別情報を持つ個別の論理グループです。各論理ドメインは独立して作成、削除、再構成、および再起動することができ、そのときサーバーの電源の再投入は必要ありません。パフォーマンスおよびセキュリティー上の理由から、さまざまなアプリケーションソフトウェアを異なる論理ドメイン上で動作させて、アプリケーションの独立性を維持することができます。

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 つだけ存在できます。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 (for Bourne or K shell)
$ MANPATH=$MANPATH:/opt/SUNWldm/man; export MANPATH
% set PATH=($PATH /opt/SUNWldm/bin) (for C shell)
% set MANPATH=($MANPATH /opt/SUNWldm/man)
```

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 (110 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提供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、論理ドメインチャネル (LDC) と呼ばれるドメイン間通信チャネルを使用して、相互に、またはサービスの対象と通信します。Logical Domains 1.0.3 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提供します。

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 (121 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 1.0.3 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 1.0.3 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** ー 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。これは自動的には有効になりません。

- 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 アカウント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、セキュリティー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は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
`add-*`	`solaris.ldoms.write`
`bind-domain`	`solaris.ldoms.write`
`list`	`solaris.ldoms.read`
`list-*`	`solaris.ldoms.read`
`panic-domain`	`solaris.ldoms.write`
`remove-*`	`solaris.ldoms.write`
`set-*`	`solaris.ldoms.write`
`start-domain`	`solaris.ldoms.write`
`stop-domain`	`solaris.ldoms.write`
`unbind-domain`	`solaris.ldoms.write`

* 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、122 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 1.0.3 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) 1.0.3 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 1.0.3 ソフトウェアへのアップグレード

既存の LDoms 1.0.1 および 1.0.2 の設定は、LDoms 1.0.3 ソフトウェアでも機能するため、LDoms 1.0.1 または 1.0.2 ソフトウェアから LDoms 1.0.3 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 1.0.3 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 1.0.3 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 1.0.3 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. LDoms 1.0.3 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) 1.0.3 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 1.0.3 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) 1.0.3 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) 1.0.3 リリースノート』の「必須、推奨、およびオプションのソフトウェアと必須パッチ」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で必要なシステムファームウェアについては、『Logical Domains (LDoms) 1.0.3 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[sm] 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に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_3-04.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_3-04.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 1.0.3 ソフトウェアのディレクトリ構造

```
LDoms_Manager-1_0_3/
     Install/
          install-ldm
     Legal/
          Ldoms_1.0.3_Entitlement.txt
          Ldoms_1.0.3_SLA_Entitlement.txt
     Product/
          SUNWjass/
          SUNWldm.v/
          SUNWldmib.v
     README
```

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は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 1.0.3 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 1.0.3 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は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 1.0.3 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 1.0.3 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 1.0.3 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/819-7823`

1. 『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。
   a. 「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。
   b. 「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。
   c. 「`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。
2. 「`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

1. 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
```

2. `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
```

3. 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。
4. `/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
```

5. パスを反映するように `user.init` ファイルを編集します。

6. 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
```

7. 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
```

8. 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。

9. 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。

10. 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は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、エントリは次のようになるはずです。

```
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) 1.0.3 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) 1.0.3 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 1.0.3 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に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 1.0.3 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=1.0.3,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 (ldom_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 ldom_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ァイルを削除する

- ローカルユーザーアカウントのすべてのプロファイルを削除します (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
- vconscon − 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信装置サービス (vcc) を、すべての論理ドメインコンソールの端末集配信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信装置サービス (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。47 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
```

注 – ドメイン再構成の一部として仮想スイッチデバイスを追加した場合は、必ず再構成のための再起動を行なってください。通常、この操作は制御ドメインの設定中に行います。詳細は、46 ページの「論理ドメインを使用するための再起動」を参照してください。

---

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。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ドメインを再起動する必要があります。

▼ 再起動する

1. 次の起動時に再構成を開始します。

```
primary# touch /reconfigure
```

注 – この再構成の手順は、ドメイン再構成の一部として仮想スイッチデバイスを追加した場合にのみ必要です。通常、この操作は制御ドメインの設定中に行います。

2. 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) のプロパティーを仮想スイッチ (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 vconscon 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) 1.0.3 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の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。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の名前です。ボリューム名は、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、ボリューム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、ボリューム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 backend vol1@primary-vds0
```

各表記の意味は次のとおりです。

- *backend* 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、このバックエンドに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。ボリューム名は、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、ボリューム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、ボリューム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、接続する既存のボリュームの名前です。
- 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 は除外されます。論理ドメインの仮想ディスクの形式は、c*N*d*N*s*N* です。c*N* は仮想コントローラ、d*N* は仮想ディスク番号、および s*N* 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 LDoms 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 (t*N*) 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 c*N*t*N*d*N*s*N* 形式の代わりに、仮想ディスクデバイス名は c*N*d*N*s*N* という形式になります。ここで、c*N* は仮想コントローラ、d*N* は仮想ディスク番号、および s*N* 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

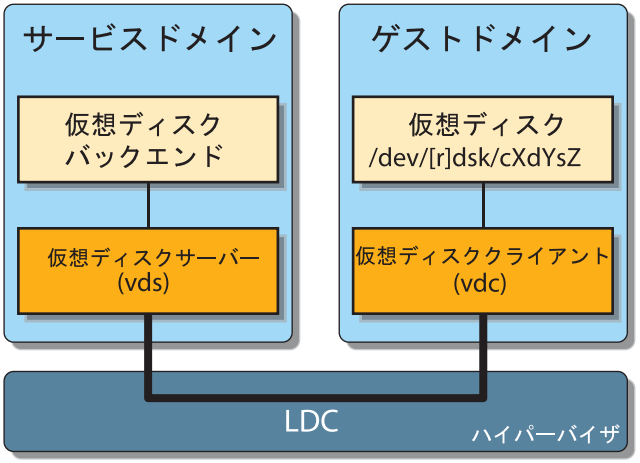
Logical Domains での仮想ディスクの使用

この章では、Logical Domains ソフトウェアで仮想ディスクを使用する方法について説明します。

仮想ディスクの概要

仮想ディスクには、2 つの構成要素があります。ドメインゲストに表示される仮想ディスク自体と、データの格納先であり仮想 I/O の終端である仮想ディスクバックエンドです。仮想ディスクバックエンドは、仮想ディスクサーバー (vds) ドライバによって、サービスドメインからエクスポートされます。vds ドライバは、論理ドメインチャネル (LDC) を使用して、ハイパーバイザを介してゲストドメインの仮想ディスククライアント (vdc) ドライバと通信します。最終的には、仮想ディスクはゲストドメイン内の /dev/[r]dsk/cXdYsZ デバイスとして表示されます。

図 5-1　Logical Domains での仮想ディスク



仮想ディスクバックエンドには、物理ディスク、物理ディスクスライス、ファイル、ボリューム管理フレームワークのボリュームを使用できます。ボリューム管理フレームワークとは、ZFS (Zettabyte File System)、Solaris™ ボリュームマネージャー (SVM)、Veritas Volume Manager (VxVM)、サービスドメインからアクセス可能なすべてのディスク擬似デバイスなどです。

仮想ディスクの管理

この節では、ゲストドメインへの仮想ディスクの追加、仮想ディスクオプションとタイムアウトオプションの変更、およびゲストドメインからの仮想ディスクの削除について説明します。仮想ディスクオプションの説明については、60 ページの「仮想ディスクバックエンドオプション」を参照してください。仮想ディスクのタイムアウトの説明については、71 ページの「仮想ディスクのタイムアウト」を参照してください。

▼ 仮想ディスクを追加する

1. 次のコマンドを使用して、仮想ディスクバックエンドをサービスドメインからエクスポートします。

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

2. 次のコマンドを使用して、ゲストドメインにバックエンドを割り当てます。

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

注 – バックエンドは、ゲストドメイン (*ldom*) がバインドされたときに、実際にサービスドメインからエクスポートされ、ゲストドメインに割り当てられます。

▼ 仮想ディスクバックエンドを複数回エクスポートする

仮想ディスクバックエンドは、同じ仮想ディスクまたは別の仮想ディスクサーバーのいずれかを介して複数回エクスポートできます。仮想ディスクバックエンドのエクスポートされたインスタンスは、それぞれ同じゲストドメインまたは別のゲストドメインのいずれかに割り当てることができます。

仮想ディスクバックエンドを複数回エクスポートする場合は、排他 (`excl`) オプションを指定してエクスポートしないでください。`excl` オプションを指定すると、バックエンドのエクスポートは 1 回のみ許可されます。`ro` オプションを指定すると、バックエンドは読み取り専用デバイスとして問題なく複数回エクスポートできます。

注意 – 仮想ディスクバックエンドが複数回エクスポートされる際は、ゲストドメインで動作中のアプリケーションおよびその仮想ディスクを使用中のアプリケーションが、同時の書き込みアクセスを調整および同期化して、データの一貫性を確保する役割を果たします。

次の例では、同じ仮想ディスクサービスを介して 2 つの異なるゲストドメインに同じ仮想ディスクを追加する方法について説明します。

1. 次のコマンドを使用して、サービスドメインから仮想ディスクバックエンドを 2 回エクスポートします。

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

`add-vdsdev` サブコマンドは、次の警告を表示して、バックエンドが複数回エクスポートされていることを示しています。

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

2. 次のコマンドを使用して、エクスポートされたバックエンドを各ゲストドメインに割り当てます。

   ldom1 と ldom2 には、異なる *disk_name* を指定できます。

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

▼ 仮想ディスクオプションを変更する

- サービスドメインからバックエンドがエクスポートされたあとに、次のコマンドを使用して仮想ディスクオプションを変更できます。

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

▼ タイムアウトオプションを変更する

- 仮想ディスクがゲストドメインに割り当てられたあとに、次のコマンドを使用して仮想ディスクのタイムアウトを変更できます。

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

▼ 仮想ディスクを削除する

1. 次のコマンドを使用して、ゲストドメインから仮想ディスクを削除します。

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

2. 次のコマンドを使用して、サービスドメインからの対応するバックエンドのエクスポートを停止します。

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

仮想ディスクの表示

バックエンドが仮想ディスクとしてエクスポートされると、ゲストドメインにフルディスクまたは 1 つのスライスディスクとして表示可能になります。表示形式は、バックエンドの種類およびバックエンドのエクスポート時に使用したオプションによって異なります。

フルディスク

バックエンドをフルディスクとしてドメインにエクスポートすると、8 つのスライス (`s0` 〜 `s7`) を持つ通常のディスクとしてドメインに表示されます。このようなディスクは、`format`(1M) コマンドを使用して表示できます。ディスクのパーティションテーブルは、`fmthard`(1M) または `format`(1M) コマンドのいずれかを使用して変更できます。

また、フルディスクは OS インストールソフトウェアからも表示でき、OS のインストール先のディスクとして選択できます。

どのバックエンドも、フルディスクとしてエクスポートできます。ただし、1 つのスライスディスクとしてのみエクスポート可能な物理ディスクスライスは除きます。

1 つのスライスディスク

バックエンドを 1 つのスライスディスクとしてドメインにエクスポートすると、1 つのパーティション (`s0`) を持つディスクとしてドメインに表示されます。このようなディスクは、`format`(1M) コマンドでは表示できません。また、そのパーティションテーブルを変更することもできません。

1 つのスライスディスクは、OS インストールソフトウェアからは表示できず、OS をインストール可能なディスクデバイスとして選択することもできません。

どのバックエンドも、1 つのスライスディスクとしてエクスポートできます。ただし、フルディスクとしてのみエクスポートできる物理ディスクは除きます。

仮想ディスクバックエンドオプション

仮想ディスクのバックエンドをエクスポートする際には、さまざまなオプションを指定できます。これらのオプションは、`ldm add-vdsdev` コマンドの `options=` 引数にコンマ区切りのリストとして指定します。有効なオプションは、`ro`、`slice`、および `excl` です。

読み取り専用 (`ro`) オプション

読み取り専用 (`ro`) オプションは、バックエンドが読み取り専用デバイスとしてエクスポートされることを指定します。その場合、ゲストドメインに割り当てられるこの仮想ディスクに対しては読み取り操作のアクセスのみが可能で、仮想ディスクへの書き込み操作は失敗します。

排他 (`excl`) オプション

排他 (`excl`) オプションは、サービスドメインのバックエンドを仮想ディスクとして別のドメインにエクスポートするときに、仮想ディスクサーバーによって排他的に開かれる必要があることを指定します。バックエンドが排他的に開かれると、サービスドメインのほかのアプリケーションがこのバックエンドにアクセスすることはできません。これによって、サービスドメインで動作するアプリケーションが、ゲストドメインでも使用されているバックエンドを誤って使用することはなくなります。

注 – ドライバには `excl` オプションを受け入れないものもあるため、一部の仮想ディスクバックエンドを排他的に開くことが許可されません。`excl` オプションが物理ディスクおよびスライスで機能することはわかっていますが、このオプションはファイルでは機能しません。ディスクボリュームなどの擬似デバイスでは機能する場合と機能しない場合があります。バックエンドのドライバで排他的オープンが受け入れられない場合、バックエンドの `excl` オプションは無視され、バックエンドは排他的に開かれません。

`excl` オプションによって、サービスドメインで動作中のアプリケーションが、ゲストドメインにエクスポートされるバックエンドにアクセスできなくなるため、次の場合は `excl` オプションを設定しないでください。

- ゲストドメインの動作中に `format`(1M)、`luxadm`(1M) などのコマンドを使用して物理ディスクを管理できるようにする場合は、これらの物理ディスクをエクスポートする際に `excl` オプションを指定しないでください。

- RAID、ミラー化ボリュームなどの SVM ボリュームをエクスポートする場合は、`excl` オプションを設定しないでください。このようにしないと、RAID またはミラー化ボリュームのコンポーネントに障害が発生した場合に、SVM で一部の復旧処理の開始が妨げられる可能性があります。詳細は、81 ページの「SVM での仮想ディスクの使用」を参照してください。
- Veritas Volume Manager (VxVM) がサービスドメインにインストールされていて、Veritas Dynamic Multipathing (VxDMP) が物理ディスクに対して有効な場合は、`excl` オプションを指定せずに物理ディスクをエクスポートする必要があります。このようにしないと、仮想ディスクサーバー (`vds`) が物理ディスクデバイスを開くことができないため、エクスポートは失敗します。詳細は、82 ページの「VxVM のインストール時の仮想ディスクの使用」を参照してください。
- 同じ仮想ディスクバックエンドを同じ仮想ディスクサービスから複数回エクスポートする場合の詳細は、57 ページの「仮想ディスクバックエンドを複数回エクスポートする」を参照してください。

デフォルトでは、バックエンドは排他的ではない状態で開かれます。このため、バックエンドが別のドメインにエクスポートされている間でも、サービスドメインで動作中のアプリケーションはこのバックエンドを使用できます。これは、Solaris 10 5/08 OS リリースから導入された新しい動作です。Solaris 10 5/08 OS より前のリリースでは、ディスクバックエンドは常に排他的に開かれ、バックエンドを排他的でない状態で開くことはできませんでした。

スライス (`slice`) オプション

通常、バックエンドは、その種類に応じてフルディスクまたは 1 つのスライスディスクのいずれかとしてエクスポートされます。`slice` オプションを指定すると、バックエンドは強制的に 1 つのスライスディスクとしてエクスポートされます。

このオプションは、バックエンドの raw コンテンツをエクスポートする場合に便利です。たとえば、データを格納済みの ZFS または SVM ボリュームがある場合に、ゲストドメインでこのデータにアクセスするには、`slice` オプションを使用して ZFS または SVM ボリュームをエクスポートする必要があります。

このオプションの詳細は、「仮想ディスクバックエンド」を参照してください。

仮想ディスクバックエンド

仮想ディスクバックエンドは、仮想ディスクのデータの格納場所です。バックエンドには、ディスク、ディスクスライス、ファイル、またはボリューム (ZFS、SVM、VxVM など) を使用できます。バックエンドは、バックエンドをサービスドメインからエクスポートする際に `slice` オプションを設定するかどうかに応じて、フルディ

スクまたは 1 つのスライスディスクのいずれかとしてゲストドメインに表示されます。デフォルトでは、仮想ディスクバックエンドは読み取りおよび書き込み可能なフルディスクとして排他的でない状態でエクスポートされます。

物理ディスクまたはディスクの LUN

物理ディスクまたは論理ユニット番号 (LUN) は、常にフルディスクとしてエクスポートされます。この場合、仮想ディスクドライバ (vds および vdc) は仮想ディスクからの入出力を転送し、物理ディスクまたは LUN へのパススルーとして動作します。

slice オプションを設定せずにそのディスクのスライス 2 (s2) に対応するデバイスをエクスポートすると、物理ディスクまたは LUN はサービスドメインからエクスポートされます。slice オプションを指定してディスクのスライス 2 をエクスポートすると、ディスク全体ではなくこのスライスのみがエクスポートされます。

▼ 物理ディスクを仮想ディスクとしてエクスポートする

1. たとえば、物理ディスク c1t48d0 を仮想ディスクとしてエクスポートするには、次のようにそのディスクのスライス 2 (c1t48d0s2) をサービスドメインからエクスポートする必要があります。

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

2. たとえば、サービスドメインから、ディスク (pdisk) をゲストドメイン ldg1 に割り当てます。

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

3. ゲストドメインが起動されて Solaris OS が実行されたら、ディスク (c0d1 など) を表示して、そのディスクがアクセス可能であり、フルディスク (8 つのスライスを持つ通常のディスク) であることを確認できます。

```
ldg1# ls -1 /dev/dsk/c0d1s*
/dev/dsk/c0d1s0
/dev/dsk/c0d1s1
/dev/dsk/c0d1s2
/dev/dsk/c0d1s3
/dev/dsk/c0d1s4
```

```
/dev/dsk/c0d1s5
/dev/dsk/c0d1s6
/dev/dsk/c0d1s7
```

物理ディスクスライス

物理ディスクスライスは、常に 1 つのスライスディスクとしてエクスポートされます。この場合、仮想ディスクドライバ (vds および vdc) は仮想ディスクから入出力を転送し、物理ディスクスライスへのパススルーとして動作します。

物理ディスクスライスは、対応するスライスデバイスをエクスポートすることで、サービスドメインからエクスポートされます。デバイスがスライス 2 と異なる場合は、slice オプションの指定の有無にかかわらず、自動的に 1 つのスライスディスクとしてエクスポートされます。デバイスがディスクのスライス 2 である場合は、slice オプションを設定して、スライス 2 のみを 1 つのスライスディスクとしてエクスポートする必要があります。このようにしないと、ディスク全体がフルディスクとしてエクスポートされます。

▼ 物理ディスクスライスを仮想ディスクとしてエクスポートする

1. **たとえば、物理ディスク c1t57d0 のスライス 0 を仮想ディスクとしてエクスポートするには、そのスライス (c1t57d0s0) に対応するデバイスをサービスドメインから次のようにエクスポートする必要があります。**

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

   スライスは常に 1 つのスライスディスクとしてエクスポートされるため、slice オプションを指定する必要はありません。

2. **たとえば、サービスドメインから、ディスク (pslice) をゲストドメイン ldg1 に割り当てます。**

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

3. **ゲストドメインが起動されて Solaris OS が実行されたら、ディスク (c0d1 など) を表示して、そのディスクがアクセス可能であり、1 つのスライスディスク (s0) であることを確認できます。**

```
ldg1# ls -1 /dev/dsk/c0d13s*
/dev/dsk/c0d13s0
```

▼ スライス 2 をエクスポートする

- スライス 2 (ディスク `c1t57d0s2` など) をエクスポートするには、`slice` オプションを指定する必要があります。このようにしないと、ディスク全体がエクスポートされます。

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

ファイルおよびボリューム

ファイルまたはボリューム (たとえば ZFS または SVM からの) は、`slice` オプションの指定の有無に応じて、フルディスクまたは 1 つのスライスディスクのいずれかとしてエクスポートされます。

フルディスクとしてエクスポートされるファイルまたはボリューム

`slice` オプションを設定しない場合、ファイルまたはボリュームはフルディスクとしてエクスポートされます。この場合、仮想ディスクドライバ (`vds` および `vdc`) は仮想ディスクから入出力を転送し、仮想ディスクのパーティション分割を管理します。最終的には、このファイルまたはボリュームは、仮想ディスクのすべてのスライスのデータ、およびパーティション分割とディスク構造の管理に使用されるメタデータを含むディスクイメージになります。

空のファイルまたはボリュームをフルディスクとしてエクスポートすると、未フォーマットのディスク、つまり、パーティションのないディスクとしてゲストドメインに表示されます。このため、ゲストドメインで `format`(1M) コマンドを実行して、使用可能なパーティションを定義し、有効なディスクラベルを書き込む必要があります。ディスクが未フォーマットの間、この仮想ディスクへの入出力はすべて失敗します。

注 – Solaris 10 5/08 OS より前のリリースでは、空のファイルが仮想ディスクとしてエクスポートされると、システムによってデフォルトのディスクラベルが書き込まれ、デフォルトのパーティションが作成されていました。Solaris 10 5/08 OS リリースではこの処理は行われなくなったため、ゲストドメインで `format`(1M) を実行してパーティションを作成する必要があります。

▼ ファイルをフルディスクとしてエクスポートする

1. サービスドメインから、ファイル (fdisk0 など) を作成して仮想ディスクとして使用します。

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

ファイルのサイズによって、仮想ディスクのサイズが定義されます。この例では、100M バイトの空のファイルを作成して、100M バイトの仮想ディスクを取得しています。

2. サービスドメインから、ファイルを仮想ディスクとしてエクスポートします。

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

この例では、slice オプションを設定していないため、ファイルはフルディスクとしてエクスポートされます。

3. たとえば、サービスドメインから、ディスク (fdisk) をゲストドメイン ldg1 に割り当てます。

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

4. ゲストドメインが起動されて Solaris OS が実行されたら、ディスク (c0d5 など) を表示して、そのディスクがアクセス可能で、フルディスク (8 つのスライスを持つ通常のディスク) であることを確認できます。

```
ldg1# ls -1 /dev/dsk/c0d5s*
/dev/dsk/c0d5s0
/dev/dsk/c0d5s1
/dev/dsk/c0d5s2
/dev/dsk/c0d5s3
/dev/dsk/c0d5s4
/dev/dsk/c0d5s5
/dev/dsk/c0d5s6
/dev/dsk/c0d5s7
```

1 つのスライスディスクとしてエクスポートされるファイルまたはボリューム

slice オプションを設定すると、ファイルまたはボリュームは 1 つのスライスディスクとしてエクスポートされます。この場合、仮想ディスクには 1 つのパーティション (s0) のみが含まれ、このパーティションが直接ファイルまたはボリュームバック

エンドにマップされます。ファイルまたはボリュームには仮想ディスクに書き込まれるデータのみが含まれ、パーティション情報やディスク構造などの追加データは含まれません。

ファイルまたはボリュームが 1 つのスライスディスクとしてエクスポートされると、システムは擬似的なディスクのパーティション分割のシミュレーションを行います。これにより、そのファイルまたはボリュームはディスクスライスとして表示されます。ディスクのパーティション分割のシミュレーションが行われるため、そのディスクに対してパーティションは作成しないでください。

▼ ZFS ボリュームを 1 つのスライスディスクとしてエクスポートする

1. サービスドメインから、ZFS ボリューム (`zdisk0` など) を作成して、1 つのスライスディスクとして使用します。

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

ボリュームのサイズによって、仮想ディスクのサイズが定義されます。この例では、100M バイトのボリュームを作成して、100M バイトの仮想ディスクを取得しています。

2. サービスドメインから、その ZFS ボリュームに対応するデバイスをエクスポートします。このボリュームが 1 つのスライスディスクとしてエクスポートされるように `slice` オプションを設定します。

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

3. たとえば、サービスドメインから、ボリューム (`zdisk0`) をゲストドメイン `ldg1` に割り当てます。

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

4. ゲストドメインが起動されて Solaris OS が実行されたら、ディスク (`c0d9` など) を表示して、そのディスクがアクセス可能で、1 つのスライスディスク (`s0`) であることを確認できます。

```
ldg1# ls -1 /dev/dsk/c0d9s*
/dev/dsk/c0d9s0
```

ボリュームのエクスポートおよび下位互換性

Solaris 10 5/08 OS より前のリリースでは、`slice` オプションがなく、ボリュームは 1 つのスライスディスクとしてエクスポートされていました。ボリュームを仮想ディスクとしてエクスポートする構成である場合に、そのシステムを Solaris 10 5/08 OS にアップグレードすると、ボリュームは 1 つのスライスディスクではなくフルディスクとしてエクスポートされるようになります。アップグレード前の動作を保持して、ボリュームを 1 つのスライスディスクとしてエクスポートするには、次のいずれかを実行する必要があります。

- LDoms 1.0.3 ソフトウェアで `ldm set-vdsdev` コマンドを使用して、1 つのスライスディスクとしてエクスポートするすべてのボリュームに `slice` オプションを設定します。このコマンドの詳細は、`ldm` マニュアルページまたは『Logical Domains (LDoms) Manager 1.0.3 Man Page Guide』を参照してください。
- 次の行を、サービスドメインの `/etc/system` ファイルに追加します。

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

注 – この調整可能なオプションを設定すると、すべてのボリュームが強制的に 1 つのスライスディスクとしてエクスポートされ、ボリュームをフルディスクとしてエクスポートできなくなります。

各種のバックエンドのエクスポート方法の概要

バックエンド	スライスオプションなし	スライスオプションを設定
ディスク (ディスクスライス 2)	フルディスク*	1 つのスライスディスクd
ディスクスライス (スライス 2 以外)	1 つのスライスディスク†	1 つのスライスディスク
ファイル	フルディスク	1 つのスライスディスク
ボリューム (ZFS、SVM、VxVM など)	フルディスク	1 つのスライスディスク

* ディスク全体をエクスポートします。

† スライスは常に 1 つのスライスディスクとしてエクスポートされます。

d スライス 2 のみをエクスポートします。

ガイドライン

ループバックファイル (lofi) ドライバの使用

ループバックファイル (lofi) ドライバを使用すると、ファイルを仮想ディスクとしてエクスポートできます。ただし、これを行うと別のドライバ層が追加され、仮想ディスクのパフォーマンスに影響を及ぼします。代わりに、フルディスクまたは 1 つのスライスディスクとしてファイルを直接エクスポートすることができます。64 ページの「ファイルおよびボリューム」を参照してください。

ディスクスライスの直接的または間接的なエクスポート

仮想ディスクとしてスライスを直接的に、または SVM ボリュームを介すなどして間接的にエクスポートするには、prtvtoc(1M) コマンドを使用して、スライスが物理ディスクの最初のブロック (ブロック 0) で開始されていないことを確認します。

物理ディスクの最初のブロックから始まるディスクスライスを直接的または間接的にエクスポートする場合は、物理ディスクのパーティションテーブルを上書きして、そのディスクのすべてのパーティションにアクセスできないようにすることもできます。

CD、DVD および ISO イメージ

コンパクトディスク (CD) またはデジタル多用途ディスク (DVD) のエクスポートは、通常のディスクと同じ方法で実行できます。CD または DVD をゲストドメインにエクスポートするには、CD または DVD デバイスのスライス 2 をフルディスクとして、つまり `slice` オプションを指定しないでエクスポートします。

注 – CD または DVD ドライブ自体をエクスポートすることはできません。エクスポートできるのは、CD または DVD ドライブ内の CD または DVD のみです。このため、CD または DVD はエクスポート前にドライブ内に存在している必要があります。また、CD または DVD をエクスポートできるようにするには、その CD または DVD がサービスドメインで使用中になっていない必要があります。特に、ボリューム管理ファイルシステムの `volfs`(7FS) サービスが CD または DVD を使用していてはいけません。`volfs` によるデバイスの使用を解除する方法については、70 ページの「CD または DVD をサービスドメインからゲストドメインにエクスポートする」を参照してください。

ファイルまたはボリュームに CD または DVD の ISO (国際標準化機構) イメージが格納されている場合に、そのファイルまたはボリュームをフルディスクとしてエクスポートすると、ゲストドメインで CD または DVD として表示されます。

CD、DVD、または ISO イメージをエクスポートすると、自動的にゲストドメインで読み取り専用デバイスとして表示されます。ただし、ゲストドメインから CD の制御操作を実行することはできません。つまり、ゲストドメインから CD の起動、停止、または取り出しは実行できません。エクスポートされた CD、DVD、または ISO イメージを起動可能な場合は、対応する仮想ディスクでゲストドメインを起動できます。

たとえば、Solaris OS インストール DVD をエクスポートした場合は、その DVD に対応する仮想ディスク上のゲストドメインを起動し、その DVD からゲストドメインをインストールすることができます。これを行うには、ゲストドメインで `ok` プロンプトが表示されたときに次のコマンドを使用します。

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

n は、エクスポートされた DVD を表す仮想ディスクのインデックスです。

注 – Solaris OS インストール DVD をエクスポートし、その DVD に対応する仮想ディスク上でゲストドメインを起動してゲストドメインをインストールする場合、インストール中に DVD を変更することはできません。このため、異なる CD または DVD を要求するインストール手順は省略する必要がある場合があります。または、要求されたメディアにアクセスするための代替パスを指定する必要があります。

▼ CD または DVD をサービスドメインからゲストドメインにエクスポートする

1. CD または DVD ドライブに CD または DVD を挿入します。
2. サービスドメインから、ボリューム管理デーモンの `vold`(1M) が動作中でオンラインかどうかを確認します。

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

3. 次のいずれかを実行します。
   - ボリューム管理デーモンが動作中またはオンラインでない場合は、手順 5 に進みます。
   - 手順 2 の例に示すように、ボリューム管理デーモンが動作中でオンラインの場合は、次の手順を実行します。

   a. `/etc/vold.conf` ファイルを編集して、次の文字列で始まる行をコメントアウトします。

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

   詳細は、`vold.conf`(1M) マニュアルページを参照してください。

   b. サービスドメインから、ボリューム管理ファイルシステムサービスを再起動します。

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

4. サービスドメインから、CD-ROM デバイスのディスクパスを検出します。

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

5. サービスドメインから、CD または DVD ディスクデバイスをフルディスクとしてエクスポートします。

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

6. サービスドメインから、エクスポートされた CD または DVD をゲストドメイン (この例では `ldg1`) に割り当てます。

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

CD または DVD の複数回のエクスポート

CD または DVD は複数回エクスポートし、異なるゲストドメインに割り当てることができます。詳細は、57 ページの「仮想ディスクバックエンドを複数回エクスポートする」を参照してください。

仮想ディスクのタイムアウト

デフォルトでは、仮想ディスクバックエンドへのアクセスを提供するサービスドメインが停止すると、ゲストドメインから対応する仮想ディスクへのすべての入出力がブロックされます。サービスドメインが動作していて、仮想ディスクバックエンドへの入出力要求が処理されている場合、入出力は自動的に再開されます。

ただし、ファイルシステムまたはアプリケーションが入出力処理をブロックしない場合がありますが、そのような場合でもサービスドメインの停止状態が長すぎる場合は失敗し、エラーが報告されます。現在は、仮想ディスクごとに接続タイムアウト時間を設定することが可能になり、ゲストドメインの仮想ディスククライアントとサービスドメインの仮想ディスクサーバー間の接続確立に使用できます。タイムアウト時間に達した場合、サービスドメインが停止し、仮想ディスククライアントと仮想ディスクサーバー間の接続が再確立されていない間中、保留中の入出力および新規の入出力は失敗します。

このタイムアウトは、次のいずれかを実行すると設定できます。

■ ldm add-vdisk コマンドを使用します。

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

■ ldm set-vdisk コマンドを使用します。

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

タイムアウトは秒単位で指定します。タイムアウトを 0 に設定すると、タイムアウトは無効になり、サービスドメインの停止中は入出力がブロックされます (デフォルトの設定および動作)。

また、ゲストドメインの /etc/system ファイルに次の行を追加すると、タイムアウトを設定できます。

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

注 – この調整可能なオプションを設定すると、ldm CLI を使用して設定されたタイムアウトが上書きされます。また、この調整可能なオプションはゲストドメインのすべての仮想ディスクのタイムアウトを設定します。

仮想ディスクおよび SCSI

物理 SCSI ディスクまたは LUN をフルディスクとしてエクスポートする場合、対応する仮想ディスクでは、ユーザー SCSI コマンドインタフェース uscsi(7D) および多重ホストディスク制御操作 mhd(7I) がサポートされます。バックエンドとしてファイルまたはボリュームを含む仮想ディスクなど、その他の仮想ディスクでは、これらのインタフェースはサポートされません。

そのため、SCSI コマンド (SVM metaset、Solaris Cluster shared devices など) を使用するアプリケーションまたは製品機能は、バックエンドとして物理 SCSI ディスクを含む仮想ディスクのみを使用するゲストドメインで使用できます。

注 – SCSI 操作は、仮想ディスクバックエンドとして使用される物理 SCSI ディスクまたは LUN を管理するサービスドメインによって効果的に実行されます。特に、サービスドメインは SCSI の予約を行います。このため、サービスドメインおよびゲストドメインで動作するアプリケーションは、同じ物理 SCSI ディスクに対して SCSI コマンドを発行するべきではありません。そうでないと、ディスクが予期しない状態になる可能性があります。

仮想ディスクおよび format(1M) コマンド

format(1M) コマンドは、フルディスクとしてエクスポートされる仮想ディスクを使用するゲストドメインで機能します。1 つのスライスディスクは、format(1M) コマンドでは表示されません。また、このようなディスクのパーティション分割を変更することはできません。

バックエンドが SCSI ディスクである仮想ディスクでは、すべての format(1M) サブコマンドがサポートされています。バックエンドが SCSI ディスクでない仮想ディスクでは、一部の format(1M) サブコマンド (repair、defect など) がサポートされていません。この場合、format(1M) の動作は、統合開発環境 (IDE) ディスクの動作に類似しています。

注 – 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。『Logical Domains (LDoms) 1.0.3 リリースノート』のバグ ID 6363316 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 ZFS (Zettabyte File System) の使用について、次の項目で説明します。

- 73 ページの「ZFS ボリュームでの仮想ディスクの作成」
- 75 ページの「仮想ディスクでの ZFS の使用」
- 77 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 options=slice /dev/zvol/dsk/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 vzdisk 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/rdsk/c2d2s0 として表示されます。

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

ldg1# mount /dev/dsk/c2d2s0 /mnt
```

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

注 – この例では、ZFS ボリュームは 1 つのスライスディスクとしてエクスポートされています。ZFS ボリュームは、フルディスクとしてエクスポートすることもできます。仮想ディスクをパーティション分割する場合、または仮想ディスクに Solaris OS をインストールする場合は、ZFS ボリュームをフルディスクとしてエクスポートします。

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、サービス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。

▼ 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、サービス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/rdsk/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、複製はディスクで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。

ボリュームマネージャーでの仮想ディスクの使用

ZFS (Zettabyte File System)、Solaris ボリュームマネージャー (SVM)、または Veritas Volume Manager (VxVM) は、サービスドメ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。ボリュームは、1 つのスライスディスク (`slice` オプションが `ldm add-vdsdev` コマンドで指定されている場合) またはフルディスクのいずれかとしてエクスポートできます。

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。

次の例に、ボリュームを 1 つのスライスディスクとしてエクスポートする方法を示します。たとえば、サービスドメインが SVM ボリューム `/dev/md/dsk/d0` を `domain1` に 1 つのスライスディスクとしてエクスポートし、`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ドメ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 options=slice /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/rdsk/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 – 1 つのスライスディスクは 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、59 ページの「仮想ディスクの表示」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る場合は、排他 (excl) オプションを設定せずにエクスポートする必要があります。このようにしないと、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したときに、metareplace コマンドまたはホットスペアを使用した SVM ボリュームの復旧が開始されません。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 excl オプションを使用して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/c10t600C0FF00000000000015153295A4B100d0s1  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

システムに Veritas Volume Manager (VxVM) がインストールされていて、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 Veritas Dynamic Multipathing (DMP) が有効な場合は、`excl` オプションを設定せずにそのディスクまたはパーティションをエクスポートする必要があります。そうしない場合、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
```

また、`excl` オプションを設定して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、`vxdmpadm` コマンドを使用して DMP を無効にすることもできます。次に例を示します。

```
# vxdmpadm -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。

- 83 ページの「仮想ディスクでの ZFS の使用」
- 83 ページの「仮想ディスクでの SVM の使用」
- 83 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 メタデバイス状態データベース metadb(1M) 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。

バックエンドが SCSI ディスクであるすべての仮想ディスクは、SVM 共有ディスクセット metaset(1M) で使用できます。バックエンドが SCSI ディスクでない仮想ディスクは、SVM 共有ディスクセットに追加できません。バックエンドが SCSI ディスクでない仮想ディスクを SVM 共有ディスクセット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

ゲストドメインでの VxVM サポートについては、Symantec 社の VxVM ドキュメントを参照してください。

第6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 *backend* 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、95 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 6-1　　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 6-1　　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] [crypto|mau] [memory] [io]

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

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

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

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

    reconf
        remove-reconf <ldom>

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

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

コード例 6-1 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>
          set-vdisk [timeout=<seconds>] [volume=<volume_name>@<service_name>]
<disk_name> <ldom>
          remove-vdisk [-f] <disk_name> <ldom>

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

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

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

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

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

      vswitch      ( vsw )
```

コード例 6-1　　ldm のすべてのサブコマンドの構文の使用法 (続き)

```
          add-vswitch [mac-addr=<num>] [net-dev=<device>] [mode=<mode>]
<vswitch_name> <ldom>
          set-vswitch [mac-addr=<num>] [net-dev=<device>] [mode=<mode>]
<vswitch_name>
          remove-vswitch [-f] <vswitch_name>

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

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

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 6-2 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 1.0.3)
   Hypervisor control protocol v 1.0

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

▼ 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 6-3　　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           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 (-l)

- 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 6-4　　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                       routed

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

コード例 6-4　　すべてのドメインの長形式のリスト (続き)

```
    vds1             myvol-d                          /disk/d

VDPCS
    NAME
    vdpcs0
    vdpcs1

------------------------------------------------------------------------------
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
    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 6-5　　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
    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
```

コード例 6-5　すべてのドメインの拡張リスト (続き)

```
    NAME             SERVICE                     DESC
    hvctl            primary@primary             hvctl
    vldcc0           primary@primary             ds

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

VCPU
    VID    PID    UTIL STRAND
    0      1              100%

MEMORY
    RA               PA               SIZE
    0x4000000        0x34000000       512M

VLDCC
 NAME             SERVICE                     DESC
 vldcc0           primary@primary             ds

NETWORK
    NAME         SERVICE                     DEVICE       MAC
    mynet-b      vsw0@primary                network@0    08:00:20:ab:9a:12
    mynet-a      vsw0@primary                network@1    08:00:20:ab:9a:11

DISK
    NAME             VOLUME                      DEVICE     SERVER
    mydisk-a         myvol-a@vds0                disk@0     primary
    mydisk-b         myvol-b@vds0                disk@1     primary

VDPCC
    NAME             SERVICE
    myvdpcc-a        vdpcs0@primary
    myvdpcc-b        vdpcs0@primary

VCONS
    NAME             SERVICE                     PORT
    mygroup          vcc0@primary                5000
```

▼ 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 6-6　　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 6-7　　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 6-8　　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 6-9　　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
```

コード例 6-9　　ドメインのバインドのリスト (続き)

```
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                          MAC
        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                          MAC
        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

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 6-10　　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 6-11　　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
    8       100
    9       100
    10      100
    11      100
    12      100
    13      100
    14      100
    15      100
    16      100
    17      100
    18      100
    19      100
```

コード例 6-11　すべてのサーバーリソースのリスト (続き)

```
    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

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 6-12　　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 6-13　　1 つのドメインの制約のリスト

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

VCPU
    COUNT
    1

MEMORY
    SIZE
    512M

NETWORK
```

コード例 6-13　1 つのドメインの制約のリスト (続き)

```
    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 6-14　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>
      <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>
```

コード例 6-14　　ドメインの XML 形式の制約 (続き)

```
          <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 6-15　　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
|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
```

コード例 6-15　　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)

```
VARIABLES
|boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
|nvramrc=devalias vnet0 /virtual-devices@100/channel-devices@200/network@0
|use-nvramrc?=true
VNET|name=vnet0|dev=network@0|service=primary-vsw0|mac-addr=01:14:4f:fa:0f:55
VDISK|name=vdisk0|vol=vol0@primary-vds0
```

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。

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て使用すると、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

次の例では、ゲストドメイン ldg1 には net-a および net-c の 2 つの仮想ネットワークデバイスが含まれています。net-c に対応する、ldg1 での Solaris OS ネットワークインタフェース名を確認するには、次の手順を実行します。

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。

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 (107 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、106 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) プロパティー `ldmd/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。

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* <= 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 6-16 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。

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

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。

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 <= 0x7e816000 <= (0x78000000 + 1073741824 - 1)、つまり、*pa* <= PA <= (*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 6-16　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
```

コード例 6-16　論理ドメイン構成の解析可能な長形式のリスト (続き)

```
VCONS|type=SP
DOMAIN|name=ldg1|state=active|flags=normal|cons=5000|ncpu=2|mem=
805306368|util=29|uptime=903|softstate=Solaris running
VCPU
|vid=0|pid=4|util=29|strand=100
|vid=1|pid=5|util=29|strand=100
MEMORY
|ra=0x8000000|pa=0x48000000|size=805306368
VARIABLES
|auto-boot?=true
|boot-device=/virtual-devices@100/channel-devices@200/disk@0
VNET|name=net|dev=network@0|service=primary-vsw0@primary|mac-addr=
00:14:4f:f9:8f:e6
VDISK|name=vdisk-1|vol=disk-ldg1@primary-vds0|dev=disk@0|server=primary
VCONS|group=group1|service=primary-vcc0@primary|port=5000
DOMAIN|name=ldg2|state=active|flags=normal|cons=5001|ncpu=3|mem=
1073741824|util=35|uptime=775|softstate=Solaris running
VCPU
|vid=0|pid=6|util=35|strand=100
|vid=1|pid=7|util=34|strand=100
|vid=2|pid=8|util=35|strand=100
MEMORY
|ra=0x8000000|pa=0x78000000|size=1073741824
VARIABLES
|auto-boot?=true
|boot-device=/virtual-devices@100/channel-devices@200/disk@0
VNET|name=net|dev=network@0|service=primary-vsw0@primary|mac-addr=
00:14:4f:f9:8f:e7
VDISK|name=vdisk-2|vol=disk-ldg2@primary-vds0|dev=disk@0|server=primary
VCONS|group=group2|service=primary-vcc0@primary|port=5000
```

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、114 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 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ー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の 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. l (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: l
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。

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
```

4. 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
```

5. 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
```

6. 制御ドメインを停止します。

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

7. 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とき、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

Logical Domains Manager によって作成された論理ドメイン構成がインスタンス化されると、ドメイン化は有効になります。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ー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ー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 6-1 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 6-1 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:** システムは、リセットして OpenBoot ok プロンプトに進むか、または次のプロンプトに進みます。 `r)eboot, o)k prompt, or h)alt?` **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。
	有効	はい	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。
`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。

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 1.0.3 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、次回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。次回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、次回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 1.0.3 Man Page Guide』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。122 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。**

サポートされるネットワークアダプタ

論理ドメイン環境のサービスドメイン内で動作する仮想スイッチサービスは、GLDv3 準拠のネットワークアダプタと直接対話できます。GLDv3 に準拠していないネットワークアダプタは、これらのシステムで使用できますが、仮想スイッチと直接

対話することはできません。GLDv3 に準拠していないネットワークアダプタを使用する方法については、125 ページの「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」を参照してください。

▼ ネットワークアダプタが GLDv3 準拠かどうかを判別する

1. Solaris OS `dladm`(1M) コマンドを使用します。ここでは、ネットワークデバイス名として `bge0` を指定します。

```
# dladm show-link bge0
bge0            type: non-vlan   mtu: 1500      device: bge0
```

2. 出力結果の `type:` を確認します。
   - GLDv3 に準拠しているドライバの種類は、`non-vlan` または `vlan` です。
   - GLDv3 に準拠していないドライバの種類は、`legacy` です。

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. **物理デバイスを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。

論理ドメイン環境での IPMP の構成

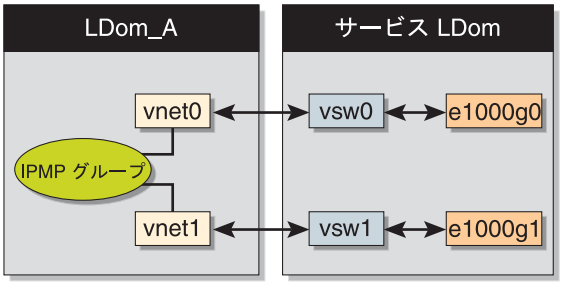
インターネットプロトコルネットワークマルチパス (IPMP) は、複数のネットワークインタフェースカード間の耐障害性と負荷分散を提供します。IPMP を使用すると、1 つ以上のインタフェースを IP マルチパスグループとして構成できます。IPMP を構成すると、システムは IPMP グループ内のインタフェースで障害が発生していないかを自動的に監視します。グループ内のインタフェースに障害が発生したり、保守のために削除されたりすると、IPMP は障害の発生したインタフェースの IP アドレスを自動的に移行して、フェイルオーバーを行います。論理ドメイン環境では、物理ネットワークインタフェースまたは仮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

IPMP グループに仮想ネット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、グループでプローブ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 1.0.3 ソフトウェアでは、現在、仮想ネット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。

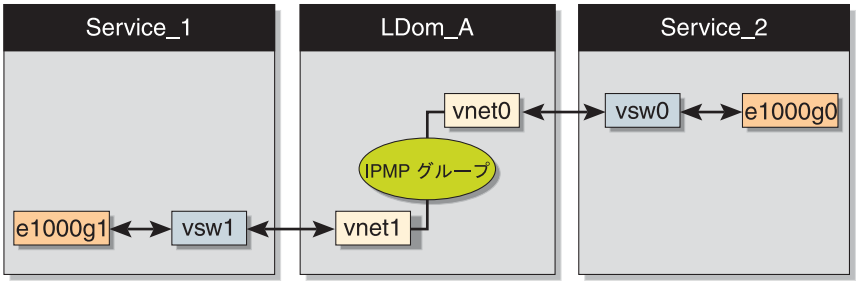
次の図に、サービスドメインで個別の仮想スイッチインスタンス (vsw0 および vsw1) に接続された 2 つの仮想ネットワーク (vnet0 および vnet1) を示します。これらは、同様に、2 つの異なる物理インタフェース (e1000g0 および e1000g1) を使用します。物理インタフェースに障害が発生した場合、LDom_A の IP 層が、プローブベースの検出を使用して対応する vnet の障害と接続の損失を検出し、vnet の二次デバイスに自動的にフェールオーバーします。

図 6-1　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 6-2　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。

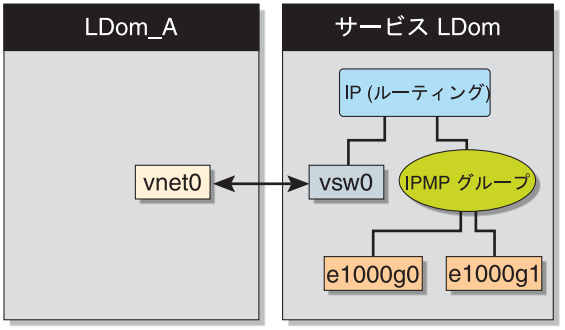
サービスドメインでの IPMP の構成と使用

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 6-3　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 1.0.3 のドキュメントで使用される用語、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。

auditreduce(1M) 監査証跡ファイルの監査レコードのマージおよび選択

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)

bsmconv(1M) BSM の有効化

bsmunconv(1M) BSM の無効化

C

CD Compact Disc (コンパクトディスク)

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)

DVD Digital Versatile Disc (デジタル多用途ディスク)

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)

`fmthard`(1M) ハードディスクのラベルの生成

`format`(1M) ディスクのパーティション分割および保守ユーティリティー

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

IDE Integrated Development Environment (統合開発環境)

`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)

ISO International Organization for Standardization (国際標準化機構)

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
lofi	ループバックファイル
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
LUN	Logical Unit Number (論理ユニット番号)

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。

metadb(1M)　SVM メタデバイス状態データベースの複製の作成および削除

metaset(1M)　ディスクセットの構成

mhd(7I)　多重ホストディスク制御操作

MMF　MultiMode Fiber (マルチモードファイバ)

MIB　Management Information Base (管理情報ベース)

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。「vdpcc」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

`praudit`(1M) 監査証跡ファイルの内容の出力

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

svcadm(1M) サービスインスタンスの操作

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)

uscsi(7D) ユーザー SCSI コマンドインタフェース

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。

vdpcc　NDPS 環境における仮想データプレーンチャネルクライアント

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)

volfs(7FS)　ボリューム管理ファイルシステム

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)

VxDMP　Veritas Dynamic MultiPathing

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

ZVOL ZFS ボリュームエミュレーションドライバ

か

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

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