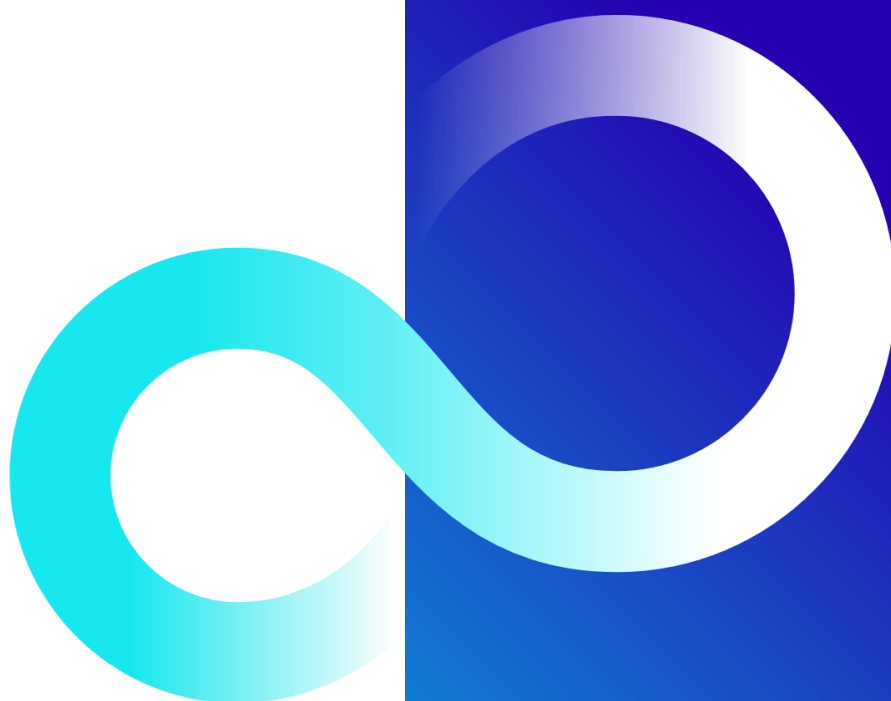


Oracle VM Server
for SPARC
Idmコマンド
クイックリファレンス



2019年12月

第3.6版

富士通株式会社

■ 使用条件

- 著作権・商標権・その他の知的財産権について

コンテンツ(文書・画像・音声等)は、著作権・商標権・その他の知的財産権で保護されています。

本コンテンツは、個人的に使用する範囲でプリントアウトまたはダウンロードできます。ただし、これ以外の利用(ご自分のページへの再利用や他のサーバへのアップロード等)については、当社または権利者の許諾が必要となります。

- 保障の制限

本コンテンツについて、当社は、その正確性、商品性、ご利用目的への適合性等に関して保証するものではなく、そのご利用により生じた損害について、当社は法律上のいかなる責任も負いかねます。本コンテンツは、予告なく変更・廃止されることがあります。

- 輸出または提供

本製品を輸出又は提供する場合は、外国為替及び外国貿易法及び米国輸出管理関連法規等の規制をご確認の上、必要な手続きをお取りください。

■ 商標について

- UNIX は、米国およびその他の国におけるオープン・グループの登録商標です。
- SPARC Enterprise、SPARC64、SPARC64 ロゴおよびすべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している、同社の米国およびその他の国における商標または登録商標です。
- Oracle と Java は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- その他各種製品名は、各社の製品名称、商標または登録商標です。

はじめに

本書の内容

- 本リファレンスは Oracle VM Server for SPARC 3.6 で提供している Idm コマンドのうち、主に使われるサブコマンド、オプションの使い方について紹介しています。
- 詳細については、Oracle 社のドキュメントを参照してください。
「Oracle VM Server for SPARC 3.6 Reference Manual」
https://docs.oracle.com/cd/E93612_01/html/E93621/index.html

留意事項

- Oracle VM Server for SPARC で提供している Idm コマンドすべてのサブコマンド、オプションを紹介していません。
すべての情報が必要な場合は、上記の Oracle 社のドキュメントをご参照ください。
- Oracle VM Server for SPARC のバージョンにより、使用できるサブコマンドやオプションが異なります。ご注意ください。

本書での表記

- ショートフォーマット(省略形)のあるコマンドは、省略形で記述
- [] 内は選択項目(省略可能)
- <> 内は必須項目
- <a | b> は a または b のどちらかを指定
- ★は SPARC M12/M10 専用サブコマンド・オプション
- 以下の用語は略称を用いて表記する場合があります。

略称	正式名称
Solaris	Oracle Solaris
Oracle VM	Oracle VM Server for SPARC

目次

1. domain (ドメインの設定)	1
add-domain (ドメインの作成)	1
set-domain (ドメインの設定変更)	2
remove-domain (ドメインの削除)	2
bind-domain (ドメインにリソースを割り当て)	3
unbind-domain (ドメインのリソース解除)	3
start-domain (ドメインの起動)	3
stop-domain (ドメインの停止)	4
panic-domain (ドメインの強制パニック)	4
2. vcpu (仮想 CPU の設定)	4
add-vcpu (CPU スレッドの追加)	4
set-vcpu (CPU スレッド数の設定)	5
remove-vcpu (CPU スレッドの削除)	5
3. core (コアの設定)	5
add-core (CPU コアの追加)	5
set-core (CPU コア数の設定)	5
remove-core (CPU コアの削除)	6
4. memory (メモリの設定)	6
add-memory (メモリ容量の追加)	6
set-memory (メモリ容量の設定)	7
remove-memory (メモリ容量の削除)	7
5. socket (CPU ソケットを指定したリソースの設定)	8
set-socket (ドメインに対して CPU ソケットを設定)	8
grow-socket (ドメインに対して CPU ソケットを設定してリソースを追加)	8
shrink-socket (ドメインに対して指定した CPU ソケットからリソースを削除)	9
list-socket (CPU ソケット情報の表示)	9
6. spconfig (構成情報の設定)	9
add-spconfig (ドメインの構成情報を保存)	9
set-spconfig (使用する構成情報を変更)	10
remove-spconfig (ドメインの構成情報を削除)	10

list-sponfig (ドメインの構成情報を一覧表示)	10
7. hvdump (ハイパーバイザダンプ機能の設定)	10
set-hvdump (ハイパーバイザダンプ機能の設定を変更)	10
list-hvdump (ハイパーバイザダンプ機能の設定内容を表示)	10
start-hvdump (ハイパーバイザのダンプを手動で採取)	11
8. vconscon (仮想コンソールサービスの設定)	11
add-vconscon (仮想コンソールサービスの作成)	11
set-vconscon (仮想コンソールサービスの設定変更)	11
remove-vconscon (仮想コンソールサービスの削除)	11
9. vswitch (仮想スイッチサービスの設定)	12
add-vswitch (仮想スイッチの作成)	12
set-vswitch (仮想スイッチの設定変更)	13
remove-vswitch (仮想スイッチの削除)	13
10. vdiskserver (仮想ディスクサービスの設定)	14
add-vdiskserver (仮想ディスクサービスの作成)	14
remove-vdiskserver (仮想ディスクサービスの削除)	14
11. vdiskserverdevice (仮想ディスクサービスのデバイスの設定)	14
add-vdiskserverdevice (仮想ディスクサービスにデバイスを追加)	14
set-vdiskserverdevice (仮想ディスクサービスに登録されたデバイスの設定変更)	15
remove-vdiskserverdevice (仮想ディスクサービスに登録されたデバイスの削除)	15
12. vdisk (仮想ディスクの設定)	15
add-vdisk (仮想ディスクの割り当て)	15
set-vdisk (仮想ディスクの設定変更)	16
remove-vdisk (仮想ディスクの削除)	16
13. vnet (仮想ネットワークの設定)	17
add-vnet (仮想ネットワークデバイスの割り当て)	17
set-vnet (仮想ネットワークデバイスの設定変更)	18
remove-vnet (仮想ネットワークデバイスの削除)	19
14. vf (SR-IOV 仮想デバイス(VF)の設定)	19
create-vf (SR-IOV 仮想デバイス(VF)の作成)	19
destroy-vf (SR-IOV 仮想デバイス(VF)の削除)	20

15.io（物理・仮想 I/O デバイスの設定）	20
add-io（物理・仮想 I/O デバイスの割り当て）	20
set-io（物理・仮想 I/O デバイスの設定変更）	21
remove-io（物理・仮想 I/O デバイスの割り当てを解除）	21
16.vconsole（仮想コンソールのオプションの設定）	22
set-vconsole（仮想コンソールの設定）	22
17.variable（環境変数の設定）	22
add-variable（環境変数の追加）	22
set-variable（環境変数の設定）	22
remove-variable（環境変数の削除）	23
18.migrate（ドメインの移行）	24
migrate-domain（ドメインの移行）	24
19.operation（操作の取消）	25
cancel-operation（操作の取消）	25
20.policy（CPU Dynamic Resource Management の設定）	25
add-policy（CPU Dynamic Resource Management ポリシーの設定）	25
set-policy（CPU Dynamic Resource Management ポリシーの設定変更）	26
remove-policy（CPU Dynamic Resource Management ポリシーの削除）	26
21.list（各情報のリスト表示）	27
list-domain（ドメインの構成情報の表示）	27
list-services（構成済みの仮想サービスの表示）	27
list-bindings（ドメインに割り当てられているリソースの表示）	27
list-devices（リソースの表示）	28
list-constraints（環境設定情報の表示）	28
list-variable（環境変数の一覧表示）	28
list-io（物理・仮想 I/O デバイスの表示）	29
list-permits（CPU コアアクティベーションの情報を一覧表示）	29
list-dependencies（仮想的なデバイス(vdisk, vnet, VF)の提供元と提供先の ドメインを表示）	29
list-netdev（ネットワークデバイスを一覧表示）	30
list-netstat（ネットワークデバイス統計情報を一覧表示）	30
list-history（Idm コマンドの実行履歴を表示）	30

22. system (ドメイン構成の復元)	31
init-system (ドメインの構成の復元)	31
23. reconf (遅延再構成モードの設定)	31
start-reconf (遅延再構成状態の開始)	31
cancel-reconf (遅延再構成状態の取消)	31
24. logctl (Oracle VM のログに出力されるロギング機能の設定)	32
set-logctl (Oracle VM のログに出力されるロギング機能の設定)	32
list-logctl (Oracle VM のログに出力されるロギング機能の表示)	32
25. Idmconsole (コンソール接続可能なドメイン／コンソールグループの 表示と接続)	33
Idmconsole (コンソール接続可能なドメイン／コンソールグループの表示と接続)	33

1. domain (ドメインの設定)

add-domain (ドメインの作成)

省略形	create	
形式	ldm create -i <file 名>	
	ldm create [オプション] <ドメイン名 1, ドメイン名 2, ...>	
オプション	-i <file 名>	構成ファイル(xml)でドメインを作成 ドメインは指定しない
	mac-addr=<number>	ドメインの MAC アドレスを指定
	hostid=<number>	ドメインのホスト ID を指定
	failure-policy=<ignore stop reset panic>	マスタードメインの障害ポリシーを指定
	master=<master ドメイン名...>	マスタードメインを指定
	max-cores=<number unlimited>	ドメインに割り当てるコアの最大数を指定 • unlimited(デフォルト) 制限なくコア数を割り当て可能
	cpu-arch=<native migration-class1 sparc64-class1 generic>	ライブマイグレーション先のサーバの種類を指定 • native(デフォルト) 同じ CPU タイプのサーバ間でのみ移動でき、 CPU 性能を最大限に活用可能 • migration-class1 SPARC T4、SPARC T5、および SPARC M5 間 で移動でき、SPARC T4 の CPU 性能を活用可 能 • sparc64-class1 SPARC M12/M10 間で移動でき、SPARC64 X の CPU 性能を活用可能 • generic 任意の CPU タイプのサーバ間で移動できる が、すべての CPU がサポートする命令以外は 使用できないため、性能に影響がある可能性 あり
	rc-add-policy=[iov]	PCI Express Direct I/O を有効化
	★shutdown-group=<priority>	ドメインの停止グループ番号を指定
	★fj-software-limit-pagesize =<pagesize> *Oracle VM 3.5 の追加オプションです。 *詳細は、下記資料をご参照ください。 システム運用・管理ガイド	ドメインの最大ページサイズを指定 SPARC M12: 256 MB、2 GB、16 GB SPARC M10: 256 MB
使用例	ldm create ldom1	ドメイン「ldom1」を作成
	ldm create -i ldom1.xml	構成ファイル「ldom1.xml」でドメインを作成
	ldm create master=primary ldom1	ドメイン「ldom1」のマスタードメインに「primary」を 設定
関連サブ コマンド	set-domain , remove-domain	
	list-domain	

set-domain (ドメインの設定変更)

省略形	modify	
形式	ldm modify -i <file 名>	
	ldm modify [オプション] <ドメイン名 1, ドメイン名 2, ...>	
オプション	-i <file 名>	構成ファイル(xml)でドメインの設定を変更 ドメイン名は指定しない
	mac-addr=<number>	ドメインの MAC アドレスを指定
	hostid=<number>	ドメインのホスト ID を指定
	failure-policy=<ignore stop reset panic>	マスタードメインの障害ポリシーを指定
	master=<master ドメイン名...>	マスタードメインを設定
	max-cores=<number unlimited>	ドメインに割り当てるコアの最大数を指定 • unlimited(デフォルト) 制限なくコア数を割り当て可能
	cpu-arch=<native migration-class1 sparc64-class1 generic>	ライブマイグレーション先のサーバの種類を指定 • native(デフォルト) 同じ CPU タイプのサーバ間でのみ移動でき、 CPU 性能を最大限に活用可能 • migration-class1 SPARC T4、SPARC T5、および SPARC M5 間 で移動でき、SPARC T4 の CPU 性能を活用可 能 • sparc64-class1 SPARC M12/M10 間で移動でき SPARC64 X の CPU 性能を活用可能 • generic 任意の CPU タイプのサーバ間で移動できる が、すべての CPU がサポートする命令以外は 使用できないため、性能に影響がある可能性 あり
	rc-add-policy=[iov]	PCI Express Direct I/O を有効化
	★shutdown-group=<priority>	ドメインの停止グループ番号を指定
関連サブ コマンド	add-domain	
	list-domain	

remove-domain (ドメインの削除)

省略形	destroy	
形式	ldm destroy -a	
	ldm destroy <ドメイン名 1, ドメイン名 2, ...>	
オプション	-a	すべてのドメインを削除 ドメイン名は指定しない
使用例	ldm destroy ldom1 ldom2	ドメイン「ldom1」と「ldom2」を削除
	ldm destroy -a	すべてのドメインを削除
関連サブ コマンド	add-domain	
	list-domain	

bind-domain (ドメインにリソースを割り当て)

省略形	bind	
形式	ldm bind [オプション] <-i file 名 ドメイン名>	
オプション	-i <file 名>	構成ファイル(xml)でドメインを作成・バインド
	-f	無効なデバイスが検出されても、強制的にバインド
	-q	デバイスの検証をしないでバインド
使用例	ldm bind ldom1	ドメイン「ldom1」をバインド
	ldm bind -i ldom1.xml	構成ファイル「ldom1.xml」でドメインを作成・バインド
関連サブコマンド	unbind-domain	
	list-domain	

unbind-domain (ドメインのリソース解除)

省略形	unbind	
形式	ldm unbind <ドメイン名>	
使用例	ldm unbind ldom1	ドメイン「ldom1」をアンバインド
関連サブコマンド	bind-domain	
	list-domain	

start-domain (ドメインの起動)

省略形	start	
形式	ldm start <-a -i file 名 ドメイン名 1, ドメイン名 2, ...>	
	ldm start [-f] [-m] <ドメイン名 1, ドメイン名 2, ...>	
オプション	-a	すべてのドメインを起動 ドメイン名は指定しない
	-i <file 名>	構成ファイル(xml)でドメインを作成・起動 ドメイン名は指定しない
	-f *Oracle VM 3.5 の追加オプションです。	サービスドメインが起動されていない場合でも、ゲストドメインを起動
	-m *Oracle VM 3.5 の追加オプションです。	MAC アドレスの衝突チェックを実行
使用例	ldm start ldom1	ドメイン「ldom1」を起動
	ldm start -a	すべてのドメインを起動
	ldm start -i ldom1.xml	構成ファイル「ldom1.xml」でドメインを作成・起動
関連サブコマンド	stop-domain	
	list-domain	

stop-domain (ドメインの停止)

省略形	stop	
形式	ldm stop [-f -q] <-a ドメイン名 1, ドメイン名 2, ...>	
	ldm stop [-h -r -t sec] [-m msg] <-a ドメイン名 1, ドメイン名 2, ...>	
オプション	-f	強制的にドメインを停止
	-q	ドメインを即時停止 (uadmin コマンド)
	-h	OS を停止し、ドメインを停止 (shutdown コマンド)
	-r	ドメインの OS をリブート (shutdown コマンド)
	-t sec	「sec」で指定した秒数後にドメインを即時停止
	-m msg	停止またはリブートするドメインに送信するメッセージを指定
	-a	すべてのドメインを停止 ドメイン名は指定しない
使用例	ldm stop ldom1 ldom2	ドメイン「ldom1」と「ldom2」を停止
	ldm stop -a	すべてのドメインを停止
	ldm stop -f ldom1	ドメイン「ldom1」を強制的に停止
	ldm stop -r -m "We will reboot your domain." ldom1	ドメイン「ldom1」の OS をリブートし、メッセージ「We will reboot your domain.」を出力
関連サブ コマンド	start-domain	
	list-domain	

panic-domain (ドメインの強制パニック)

省略形	panic	
形式	ldm panic <ドメイン名>	
使用例	ldm panic ldom1	ドメイン「ldom1」を強制パニック
関連サブ コマンド	list-domain	

2. vcpu (仮想 CPU の設定)**add-vcpu** (CPU スレッドの追加)

省略形	-	
形式	ldm add-vcpu <number> <ドメイン名>	
使用例	ldm add-vcpu 8 primary	ドメイン「primary」に 8CPU スレッドを追加
	ldm add-vcpu 16 ldom1	ドメイン「ldom1」に 16CPU スレッドを追加
関連サブ コマンド	set-vcpu , remove-vcpu	
	list-domain , list-bindings , list-devices	

set-vcpu (CPU スレッド数の設定)

省略形	-	
形式	ldm set-vcpu <number> <ドメイン名>	
使用例	ldm set-vcpu 4 primary	ドメイン「primary」に 4CPU スレッドを設定
	ldm set-vcpu 16 ldom1	ドメイン「ldom1」に 16CPU スレッドを設定
関連サブ コマンド	add-vcpu , remove-vcpu	
	list-domain , list-bindings , list-devices	

remove-vcpu (CPU スレッドの削除)

省略形	rm-vcpu	
形式	ldm rm-vcpu [オプション] <number> <ドメイン名>	
オプション	-f	CPU スレッドの削除を強制的に試行
使用例	ldm rm-vcpu 8 primary	ドメイン「primary」から 8CPU スレッドを削除
	ldm rm-vcpu -f 4 ldom1	ドメイン「ldom1」から 4CPU スレッドの削除を強制的に試行
関連サブ コマンド	set-vcpu , add-vcpu	
	list-domain , list-bindings , list-devices	

3. core (コアの設定)**add-core** (CPU コアの追加)

省略形	-	
形式	ldm add-core <number>	
	ldm add-core cid=<core-ID>[,core-ID][,...] <ドメイン名>	
オプション	cid=<core-ID>[, ...]	追加するコアの ID (物理位置) を指定
使用例	ldm add-core 8 primary	ドメイン「primary」に 8CPU コアを追加
	ldm add-core cid=60 ldom1	ドメイン「ldom1」に CPU コア (コア ID=60) を追加
関連サブ コマンド	set-core , remove-core	
	list-domain , list-bindings , list-devices	

set-core (CPU コア数の設定)

省略形	-	
形式	ldm set-core <number>	
	ldm set-core cid=<core-ID>[,core-ID][,...] <ドメイン名>	
オプション	cid=[core-ID][, ...]	割り当てるコアの ID (物理位置) を指定
使用例	ldm set-core 4 primary	ドメイン「primary」に 4CPU コアを設定
	ldm set-core cid=60 ldom1	ドメイン「ldom1」に CPU コア (コア ID=60) を設定
関連サブ コマンド	add-core , remove-core	
	list-domain , list-bindings , list-devices	

remove-core (CPU コアの削除)

省略形	rm-core	
形式	ldm rm-core <number>	
	ldm rm-core cid=<core-ID>[,core-ID][,...] <ドメイン名>	
オプション	-f	CPU コアの削除を強制的に試行
	cid=[core-ID][, ...]	削除するコアの ID (物理位置) を指定
使用例	ldm rm-core 4 primary	ドメイン「primary」から 4CPU コアを削除
	ldm rm-core cid=60 ldom1	ドメイン「ldom1」から CPU コア (CID=60) を削除
関連サブ コマンド	set-core , add-core	
	list-domain , list-bindings , list-devices	

4. memory (メモリの設定)**add-memory** (メモリ容量の追加)

省略形	add-mem	
形式	ldm add-mem [--auto-adj] <mem-size[unit]> <ドメイン名>	
	ldm add-mem mblock=<PA-start:mem-size[unit]>[,<PA-start:mem-size[unit]>][,...] <ドメイン名>	
オプション	--auto-adj	追加するメモリ容量を 256 MB 単位に切り上げ
	unit	メモリの追加単位を指定 • GB 単位の場合は、「G」または「g」 • MB 単位の場合は、「M」または「m」 • KB 単位の場合は、「K」または「k」
	mblock=<PA-start:mem-size[unit]>[,<PA-start:mem-size[unit]>][,...]	追加するメモリの物理アドレスを指定 • PA-start メモリの開始アドレス (16 進数) を指定 • mem-size メモリの容量 (256 MB 単位) を指定
使用例	ldm add-mem 1G primary	ドメイン「primary」に 1 GB のメモリを追加
	ldm add-mem 1024M ldom1	ドメイン「ldom1」に 1024 MB (1 GB) のメモリを追加
	ldm add-mem mblock=0x7e0000000000:32G ldom2	物理アドレス「0x7e0000000000」を指定してドメイン「ldom2」に 32 GB のメモリを追加
関連サブ コマンド	set-memory , remove-memory	
	list-domain , list-bindings , list-devices	

set-memory (メモリ容量の設定)

省略形	set-mem	
形式	ldm set-mem [--auto-adj] <number>[unit] <ドメイン名>	
	ldm set-mem mblock=<PA-start:mem-size[unit]> [,<PA-start:mem-size[unit]>][,...] <ドメイン名>	
オプション	--auto-adj	割り当てるメモリ容量を 256 MB 単位に切り上げ
	unit	メモリの設定単位を指定 - GB 単位の場合は、「G」または「g」 - MB 単位の場合は、「M」または「m」 - KB 単位の場合は、「K」または「k」
	mblock=<PA-start:mem-size[unit]> [,<PA-start:mem-size[unit]>][,...]	割り当てるメモリの物理アドレスを指定 - PA-start メモリの開始アドレス(16 進数)を指定 - mem-size メモリの容量(256 MB 単位)を指定
使用例	ldm set-mem 8G primary	ドメイン「primary」に 8 GB のメモリを設定
	ldm set-mem 6144M ldom1	ドメイン「ldom1」に 6144 MB(6 GB)のメモリを設定
	ldm set-mem mblock=0x7e000000000 :16G ldom2	物理アドレス「0x7e000000000」を指定してドメイン「ldom2」に 16 GB のメモリを設定
関連サブコマンド	add-memory , remove-memory	
	list-domain , list-bindings , list-devices	

remove-memory (メモリ容量の削除)

省略形	rm-mem	
形式	ldm rm-mem [--auto-adj] <number>[unit] <ドメイン名>	
	ldm rm-mem mblock=<PA-start:mem-size[unit]> [,<PA-start:mem-size[unit]>][,...] <ドメイン名>	
オプション	--auto-adj	削除するメモリ容量を 256 MB 単位に切り上げ
	unit	メモリの削除単位を指定 - GB 単位の場合は、「G」または「g」 - MB 単位の場合は、「M」または「m」 - KB 単位の場合は、「K」または「k」
	mblock=<PA-start:mem-size[unit]> [,<PA-start:mem-size[unit]>][,...]	削除するメモリの物理アドレスを指定 - PA-start メモリの開始アドレス(16 進数)を指定 - mem-size メモリの容量(256 MB 単位)を指定
使用例	ldm rm-mem 1G primary	ドメイン「primary」から 1 GB のメモリを削除
	ldm rm-mem 1024M ldom1	ドメイン「ldom1」から 1024 MB(1 GB)のメモリを削除
	ldm rm-mem mblock=0x7e000000000 :8G ldom2	物理アドレス「0x7e000000000」を指定してドメイン「ldom2」から 8 GB のメモリを削除
関連サブコマンド	set-memory , add-memory	
	list-domain , list-bindings , list-devices	

5. socket (CPU ソケットを指定したリソースの設定)

set-socket (ドメインに対して CPU ソケットを設定)

省略形	-	
形式	★ Idm set-socket [オプション] <socket_id=[id],[id]...> <ドメイン名>	
オプション	-f --force	強制的にソケット指定を実行 指定先の CPU ソケットに必要な空きリソースがない場合でも実行可能 ドメインの状態により割り当てられるリソース量は異なる - inactive 指定した CPU ソケットの空き容量すべて - bound 元々設定していたリソースと同じ容量 - active 元々設定していたリソースと同じ容量 ただし、CPU ソケットに必要な CPU やメモリのどちらか一方でも空きがない場合は実行不可
	--restore-degraded	ハード(CPU/メモリ)故障からの復旧時に使用 ドメインに割り当てられた CPU/メモリが縮退した場合、ハード交換後に実行することで、ドメインが使用する CPU コア・メモリの位置が元に戻る
使用例	Idm set-socket socket_id=2,3 Idom1	ドメイン「Idom1」に CPU ソケット「2」と「3」を設定
	Idm set-socket socket_id= Idom1	ドメイン「Idom1」の CPU ソケットの設定を削除
関連サブコマンド	grow-socket , shrink-socket , set-core , add-core	
	list-socket	

grow-socket (ドメインに対して CPU ソケットを設定してリソースを追加)

省略形	-	
形式	★ Idm grow-socket <オプション> <socket_id=[id]> <ドメイン名>	
オプション	vcpus=<number>	ドメインに追加する仮想 CPU の数を指定
	cores=<number>	ドメインに追加する CPU コア数を指定
	memory=<number>[GMK]	ドメインに追加するメモリの量と単位を指定 - GB 単位の場合は、「G」を指定 - MB 単位の場合は、「M」を指定 - KB 単位の場合は、「K」を指定
使用例	Idm grow-socket cores=1 socket_id=2 Idom1	ドメイン「Idom1」に CPU ソケット「2」の 1CPU コアを追加
関連サブコマンド	set-socket , shrink-socket , add-core	
	list-socket	

shrink-socket (ドメインに対して指定した CPU ソケットからリソースを削除)

省略形	-	
形式	★ Idm shrink-socket <オプション> <socket_id=[id]> <ドメイン名>	
オプション	vcpus=<number>	ドメインから削除する仮想 CPU の数を指定
	cores=<number>	ドメインから削除する CPU コアの数指定
	memory=<number>[GMK]	ドメインから削除するメモリの量と単位を指定 • GB 単位の場合は、「G」を指定 • MB 単位の場合は、「M」を指定 • KB 単位の場合は、「K」を指定
使用例	Idm shrink-socket cores=1 socket_id=2 Idm1	ドメイン「Idm1」から CPU ソケット「2」の 1CPU コアを削除
関連サブコマンド	set-socket , grow-socket , remove-core	
	list-socket	

list-socket (CPU ソケット情報の表示)

省略形	-	
形式	★ Idm list-socket [オプション] [socket_id=[id]] [ドメイン名]	
オプション	-l	物理 CPU セットとコア ID を一覧表示
	--free	空きのリソースのみ表示
	-o format	表示させるリソースの種類を指定 複数指定する場合はコンマ区切り • format raw, constraint, cpu, memory, physio
	-p	スクリプトなどで解析可能な形式で表示
使用例	Idm list-socket	CPU ソケット情報を表示
	Idm list-socket -o cpu,memory Idm1	ドメイン「Idm1」の CPU とメモリのリソースを表示
関連サブコマンド	set-socket , grow-socket , shrink-socket	

6. spconfig (構成情報の設定)**add-spconfig** (ドメインの構成情報を保存)

省略形	add-config	
形式	Idm add-config [オプション] <config 名>	
使用例	Idm add-config config_initial	構成情報を「config_initial」の名前で保存
関連サブコマンド	set-spconfig , remove-spconfig	
	list-spconfig	

set-spconfig (使用する構成情報を変更)

省略形	set-config	
形式	ldm set-config <config 名>	
使用例	ldm set-config config_initial	使用する構成情報に「config_initial」を指定
関連サブコマンド	add-spconfig , remove-spconfig list-spconfig	

remove-spconfig (ドメインの構成情報を削除)

省略形	rm-config	
形式	ldm rm-config [オプション] <config 名>	
使用例	ldm rm-config config_initial	構成情報「config_initial」を削除
関連サブコマンド	set-spconfig , add-spconfig list-spconfig	

list-spconfig (ドメインの構成情報を一覧表示)

省略形	ls-config	
形式	ldm ls-config [オプション]	
使用例	ldm ls-config	構成情報を一覧表示
関連サブコマンド	add-spconfig , remove-spconfig , set-spconfig	

7. hvdump (ハイパーバイザダンプ機能の設定)**set-hvdump** (ハイパーバイザダンプ機能の設定を変更)

省略形	-	
形式	★ ldm set-hvdump [オプション]	
オプション	hvdump=<on off>	ハイパーバイザダンプ機能の有効(on)/無効(off)を設定 デフォルトは「on」
	hvdump-reboot=<on off>	ダンプ採取後の自動リブートの有効(on)/無効(off)を設定 デフォルトは「off」
使用例	ldm set-hvdump hvdump=off	ハイパーバイザダンプ機能の無効化
	ldm set-hvdump hvdump-reboot=on	ダンプ採取後の自動リブートの有効化
関連サブコマンド	start-hvdump list-hvdump	

list-hvdump (ハイパーバイザダンプ機能の設定内容を表示)

省略形	ls-hvdump	
形式	★ ldm ls-hvdump	
使用例	ldm ls-hvdump	ハイパーバイザダンプ機能の設定内容を表示
関連サブコマンド	set-hvdump , start-hvdump	

start-hvdump (ハイパーバイザのダンプを手動で採取)

省略形	-	
形式	★ Idm start-hvdump	
使用例	Idm start-hvdump	ダンプを手動で採取
関連サブコマンド	set-hvdump , list-hvdump	

8. vconscon (仮想コンソールサービスの設定)**add-vconscon** (仮想コンソールサービスの作成)

省略形	add-vcc	
形式	Idm add-vcc port-range=<x>-<y> <仮想コンソールサービス名> <ドメイン名>	
オプション	x	コンソールポートの開始番号を指定
	y	コンソールポートの終了番号を指定
使用例	Idm add-vcc port-range=5000-5100 primary-vcc0 primary	ドメイン「primary」にコンソールポート「5000～5100」の仮想コンソールサービス「primary-vcc0」を作成
関連サブコマンド	set-vconscon , remove-vconscon , set-vconsole list-domain , list-bindings	

set-vconscon (仮想コンソールサービスの設定変更)

省略形	set-vcc	
形式	Idm set-vcc port-range=<x>-<y> <仮想コンソールサービス名>	
オプション	x	コンソールポートの開始番号を指定
	y	コンソールポートの終了番号を指定
使用例	Idm set-vcc port-range=5000-5200 primary-vcc0	仮想コンソールサービス「primary-vcc0」のコンソールポートを「5000～5200」に変更
関連サブコマンド	add-vconscon , remove-vconscon , set-vconsole list-domain , list-bindings	

remove-vconscon (仮想コンソールサービスの削除)

省略形	rm-vcc	
形式	Idm rm-vcc [オプション] <仮想コンソールサービス名>	
オプション	-f	強制的に仮想コンソールサービスを削除
使用例	Idm rm-vcc primary-vcc0	仮想コンソールサービス「primary-vcc0」を削除
	Idm rm-vcc -f primary-vcc0	仮想コンソールサービス「primary-vcc0」を強制的に削除
関連サブコマンド	add-vconscon list-domain , list-bindings	

9. vswitch（仮想スイッチサービスの設定）

add-vswitch（仮想スイッチの作成）

省略形	add-vsw	
形式	ldm add-vsw [オプション] <仮想スイッチ名> <ドメイン名>	
オプション	-q	ネットワークデバイスを検証しない
	pvid=<pvid_id>	VLAN ID をタグなしモードで指定
	vid=<vid_id1><,...>	VLAN ID をタグ付きモードで指定
	linkprop=phys-state	リンクベースの IPMP を使用する場合に設定
	mac-addr=<number>	仮想スイッチの MAC アドレスを指定
	net-dev=<ネットワークインターフェース名>	仮想スイッチの元となる物理デバイスを指定
	mtu=<size>	仮想スイッチデバイスの最大転送単位を指定
	id=<switch-id>	仮想スイッチデバイスの ID を指定
	inter-vnet-link=<on off auto>	同じ仮想スイッチの vnet 間の直接通信を On/Off • auto 同一 vsw 内の vnet が 8 個までは直接通信を On、vnet の数が 8 個を超えると Off
使用例	ldm add-vsw net-dev=net1 pvid=10 vid=20,30 primary-vsw0 primary	ドメイン「primary」に以下の設定で仮想スイッチ「primary-vsw0」を作成 • タグなしモードで VLAN ID「10」を指定 • タグ付きモードで VLAN ID「20」と「30」を指定 • 物理デバイス「net1」を指定
	ldm add-vsw net-dev=net1 linkprop=phys-state primary-vsw0 primary	ドメイン「primary」に以下の設定で仮想スイッチ「primary-vsw0」を作成 • リンクベースの IPMP を使用 • 物理デバイス「net1」を指定
関連サブコマンド	set-vswitch , remove-vswitch , add-vnet	
	list-domain , list-bindings , list-devices	

set-vswitch（仮想スイッチの設定変更）

省略形	set-vsw	
形式	ldm set-vsw [オプション] <仮想スイッチ名>	
オプション	-q	ネットワークデバイスを検証しない
	pvid=[pvid_id]	タグなしモードの VLAN ID を設定変更
	vid=<+ ->[vid_id1,...]	タグ付きモードの VLAN ID を設定変更 • + タグ付きモードの VLAN ID を追加 • - タグ付きモードの VLAN ID を削除
	mac-addr=[number]	仮想スイッチの MAC アドレスを変更
	net-dev=[ネットワークインターフェース名]	仮想スイッチの元となる物理デバイスを変更
	linkprop=[phys-state]	リンクベースの IPMP を使用するように変更
	mtu=[size]	最大転送単位 (MTU) を変更
	inter-vnet-link=<on off auto>	同じ仮想スイッチの vnet 間の直接通信を On/Off • auto 同一 vsw 内の vnet が 8 個までは直接通信を On、vnet の数が 8 個を超えると Off
使用例	ldm set-vsw net-dev=net1 pvid=10 vid=20,30 primary-vsw0	ドメイン「primary」の仮想スイッチ「primary-vsw0」に以下を設定 • タグなしモードの VLAN ID を「10」に設定 • タグ付きモードの VLAN ID を「20」と「30」に設定 • 物理デバイスを「net1」に設定
	ldm set-vsw linkprop=phys-state primary-vsw0	ドメイン「primary」の仮想スイッチ「primary-vsw0」でリンクベースの IPMP を使用するように設定
関連サブコマンド	add-vswitch , add-vnet	
	list-domain , list-bindings , list-devices	

remove-vswitch（仮想スイッチの削除）

省略形	rm-vsw	
形式	ldm rm-vsw [オプション] <仮想スイッチ名>	
オプション	-f	強制的に仮想スイッチを削除
使用例	ldm rm-vsw primary-vsw0	仮想スイッチ「primary-vsw0」を削除
	ldm rm-vsw -f primary-vsw0	仮想スイッチ「primary-vsw0」を強制的に削除
関連サブコマンド	add-vswitch , remove-vnet	
	list-domain , list-bindings , list-devices	

10. vdiskserver (仮想ディスクサービスの設定)

add-vdiskserver (仮想ディスクサービスの作成)

省略形	add-vds	
形式	ldm add-vds <仮想ディスクサービス名> <ドメイン名>	
使用例	ldm add-vds primary-vds0 primary	ドメイン「primary」に仮想ディスクサービス「primary-vds0」を作成
関連サブコマンド	remove-vdiskserver , add-vdiskserverdevice	
	list-domain , list-bindings , list-devices	

remove-vdiskserver (仮想ディスクサービスの削除)

省略形	rm-vds	
形式	ldm rm-vds <仮想ディスクサービス名>	
オプション	-f	強制的に仮想ディスクサービスを削除
使用例	ldm rm-vds primary-vds0	仮想ディスクサービス「primary-vds0」を削除
	ldm rm-vds -f primary-vds0	仮想ディスクサービス「primary-vds0」を強制的に削除
関連サブコマンド	add-vdiskserver , remove-vdiskserverdevice	
	list-domain , list-bindings , list-devices	

11. vdiskserverdevice (仮想ディスクサービスのデバイスの設定)

add-vdiskserverdevice (仮想ディスクサービスにデバイスを追加)

省略形	add-vdsdev	
形式	ldm add-vdsdev [オプション] <バックエンドデバイス名> <ボリューム名>@<仮想ディスクサービス名>	
オプション	-f	強制的に仮想ディスクサービスにデバイスを追加
	-q	バックエンドを検証しない
	options=<ro slice excl>	仮想ディスクのオプションを指定 • ro 読み取り専用の仮想ディスク • slice 単一スライスの仮想ディスク • excl 排他的なアクセス(1つのドメインのみで使用)
使用例	ldm add-vdsdev /IMG/vdisk0.img vol0@primary-vds0	仮想ディスクサービス「primary-vds0」にデバイス「/IMG/vdisk0.img」をボリューム名「vol0」で追加
	ldm add-vdsdev -f options=ro /dev/dsk/c3t1d0s2 vol1@primary-vds0	仮想ディスクサービス「primary-vds0」にデバイス「/dev/dsk/c3t1d0s2」をボリューム名「vol1」で読み取り専用として強制的に追加
関連サブコマンド	add-vdiskserver , set-vdiskserverdevice , remove-vdiskserverdevice , add-vdisk list-domain , list-bindings , list-devices	

set-vdiskserverdevice (仮想ディスクサービスに登録されたデバイスの設定変更)

省略形	set-vdsdev	
形式	ldm set-vdsdev [オプション] <ボリューム名>@<仮想ディスクサービス名>	
オプション	-f	強制的に仮想ディスクサービスのデバイス設定を変更
	options=<ro slice excl>	仮想ディスクのオプションを変更 • ro 読み取り専用の仮想ディスク • slice 単一スライスの仮想ディスク • excl 排他的なアクセス(1つのドメインのみで使用)
使用例	ldm set-vdsdev -f options=ro vol0@primary-vds0	仮想ディスクサービス「primary-vds0」のデバイス「vol0」を強制的に読み取り専用に変更
関連サブコマンド	add-vdiskserverdevice	
	list-domain , list-bindings , list-devices	

remove-vdiskserverdevice (仮想ディスクサービスに登録されたデバイスの削除)

省略形	rm-vdsdev	
形式	ldm rm-vdsdev [オプション] <ボリューム名>@<仮想ディスクサービス名>	
オプション	-f	強制的に仮想ディスクサービスからデバイスを削除
使用例	ldm rm-vdsdev vol0@primary-vds0	仮想ディスクサービス「primary-vds0」からデバイス「vol0」を削除
	ldm rm-vdsdev -f vol0@primary-vds0	仮想ディスクサービス「primary-vds0」からデバイス「vol0」を強制的に削除
関連サブコマンド	add-vdiskserverdevice , remove-vdiskserver , remove-vdisk	
	list-domain , list-bindings , list-devices	

12. vdisk (仮想ディスクの設定)**add-vdisk** (仮想ディスクの割り当て)

省略形	-	
形式	ldm add-vdisk [オプション] <仮想ディスク名> <ボリューム名>@<仮想ディスクサービス名> <ドメイン名>	
オプション	timeout=[seconds]	仮想ディスクへの接続をリトライする時間を指定
	id=<disk-id>	仮想ディスクデバイスの ID を設定 デフォルトでは、ID を自動生成
使用例	ldm add-vdisk vdisk0 vol0@primary-vds0 ldom1	ドメイン「ldom1」に仮想ディスクサービス「primary-vds0」のボリューム「vol0」を仮想ディスク「vdisk0」として割り当て
関連サブコマンド	remove-vdisk , add-vdiskserverdevice , set-vdisk	
	list-domain , list-bindings , list-devices	

set-vdisk（仮想ディスクの設定変更）

省略形	-	
形式	ldm set-vdisk [オプション] <仮想ディスク名> <ドメイン名>	
オプション	timeout=[seconds]	仮想ディスクへの接続をリトライする時間を変更
	volume=<ボリューム名>@<仮想ディスクサーバ名>	仮想ディスクに割り当てるボリュームを変更
使用例	ldm set-vdisk volume=vol1@primary-vds0 vdisk0 ldom1	ドメイン「ldom1」の仮想ディスク「vdisk0」を仮想ディスクサービス「primary-vds0」のボリューム「vol1」に変更
関連サブコマンド	add-vdisk	
	list-domain , list-bindings , list-devices	

remove-vdisk（仮想ディスクの削除）

省略形	rm-vdisk	
形式	ldm rm-vdisk [オプション] <仮想ディスク名> <ドメイン名>	
オプション	-f	強制的に仮想ディスクを削除
使用例	ldm rm-vdisk vdisk0 ldom1	ドメイン「ldom1」から仮想ディスク「vdisk0」を削除
	ldm rm-vdisk -f vdisk0 ldom1	ドメイン「ldom1」から仮想ディスク「vdisk0」を強制的に削除
関連サブコマンド	add-vdisk , remove-vdiskserverdevice	
	list-domain , list-bindings , list-devices	

13. vnet（仮想ネットワークの設定）

add-vnet（仮想ネットワークデバイスの割り当て）

省略形	-	
形式	ldm add-vnet [オプション] <仮想ネットワークデバイス名> <仮想スイッチ名> <ドメイン名>	
オプション	mac-addr=<number>	仮想ネットワークデバイスの MAC アドレスを指定
	alt-mac-addr=<auto number1> [,auto number2] [,...]	代替 MAC アドレスを指定 指定した代替 MAC アドレスの数だけ、Solaris ゾーンの仮想 NIC(vnic)を作成可能 auto を指定すると自動で MAC アドレスを生成 auto と number は混在可能
	auto-alt-mac-addr=<number> *Oracle VM 3.4 の追加オプションです。 *詳細は、下記資料をご参照ください。 ネットワーク設計・構築ガイド～on Oracle VM Server for SPARC～	代替 MAC アドレスの数を指定 指定した代替 MAC アドレスの数だけ、Solaris ゾーンの仮想 NIC(vnic)を作成可能 MAC アドレスは自動で生成
	pvid=<pvid_id>	VLAN ID をタグなしモードで指定
	vid=<vid_id>[,...]	VLAN ID をタグ付きモードで指定
	linkprop=phys-state	リンクベースの IPMP を使用する場合に指定
	id=<network-id>	仮想ネットワークデバイスの ID を指定 デフォルトでは、ID を自動生成
	mtu=<size>	最大転送単位(MTU)を指定
使用例	ldm add-vnet vnet0 primary-vsw0 ldom1	ドメイン「ldom1」に仮想スイッチ「primary-vsw0」から仮想ネットワークデバイス「vnet0」を割り当て
	ldm add-vnet pvid=10 vid=20,30 vnet0 primary-vsw0 ldom1	ドメイン「ldom1」に仮想スイッチ「primary-vsw0」から仮想ネットワークデバイス「vnet0」を以下の設定で割り当て • タグなしモードで VLAN ID「10」を指定 • タグ付きモードで VLAN ID「20」と「30」を指定
	ldm add-vnet linkprop=phys-state vnet0 primary-vsw0 ldom1	ドメイン「ldom1」に仮想スイッチ「primary-vsw0」からリンクベースの IPMP を使用する仮想ネットワークデバイス「vnet0」を割り当て
	ldm add-vnet auto-alt-mac-addr=2 vnet0 primary-vsw0 ldom1	ドメイン「ldom1」に仮想スイッチ「primary-vsw0」から 2 つの代替 MAC アドレスを設定した仮想ネットワークデバイス「vnet0」を割り当て
関連サブコマンド	set-vnet , remove-vnet , add-vswitch	
	list-domain , list-bindings , list-devices	

set-vnet（仮想ネットワークデバイスの設定変更）

省略形	-	
形式	Idm set-vnet [オプション] <仮想ネットワークデバイス名> <ドメイン名>	
オプション	mac-addr=<number>	仮想ネットワークデバイスの MAC アドレスを変更
	alt-mac-addr=<+ -> <auto number1> [,auto number2] [,...]	代替 MAC アドレスを変更 指定した代替 MAC アドレスの数だけ、Solaris ゾーンの仮想 NIC(vnic)を作成可能 auto を指定すると自動で MAC アドレスを生成 auto と number は混在可能 • MAC アドレスを追加する場合は、「+」を指定 • MAC アドレスを削除する場合は、「-」を指定
	auto-alt-mac-addr=[+]<number> *Oracle VM 3.4 の追加オプションです。 *詳細は、下記資料をご参照ください。 ネットワーク設計・構築ガイド～on Oracle VM Server for SPARC～	代替 MAC アドレスを変更 指定した代替 MAC アドレスの数だけ、Solaris ゾーンの仮想 NIC(vnic)を作成可能 MAC アドレスは自動で生成 • MAC アドレスを追加する場合は、「+」を指定
	vswitch=<仮想スイッチ名>	vnet に割り当てる仮想スイッチを変更
	pvid=<pvid_id>	タグなしモードの VLAN ID を変更
	linkprop=phys-state	リンクベースの IPMP を使用するように変更
	vid=<+ -><vid_id>[,...]	タグ付きモードの VLAN ID を変更 • タグ付きモードの VLAN ID を追加する場合は、「+」を指定 • タグ付きモードの VLAN ID を削除する場合は、「-」を指定
	mtu=<size>	最大転送単位(MTU)を変更
使用例	Idm set-vnet vswitch=primary-vsw1 vnet0 ldom1	ドメイン「ldom1」の仮想ネットワークデバイス「vnet0」の仮想スイッチを「primary-vsw1」に変更
	Idm set-vnet pvid=10 vid=20,30 vswitch=primary-vsw1 vnet0 ldom1	ドメイン「ldom1」の仮想ネットワークデバイス「vnet0」を以下の設定に変更 • 仮想スイッチを「primary-vsw1」に変更 • タグなしモードで VLAN ID「10」を設定 • タグ付きモードで VLAN ID「20」と「30」を設定
	Idm set-vnet linkprop=phys-state vswitch=primary-vsw1 vnet0 ldom1	ドメイン「ldom1」の仮想ネットワークデバイス「vnet0」を以下の設定に変更 • 仮想スイッチを「primary-vsw1」に変更 • リンクベースの IPMP を使用するように変更
	Idm set-vnet alt-mac-addr=auto,auto vnet0 ldom1	ドメイン「ldom1」の仮想ネットワークデバイス「vnet0」に代替 MAC アドレスを 2 つ自動生成
	Idm set-vnet auto-alt-mac-addr=+2 vnet0 ldom1	ドメイン「ldom1」の仮想ネットワークデバイス「vnet0」に代替 MAC アドレスを 2 つ追加
関連サブコマンド	add-vnet	
	list-domain , list-bindings , list-devices	

remove-vnet（仮想ネットワークデバイスの削除）

省略形	rm-vnet	
形式	ldm rm-vnet [オプション] <仮想ネットワークデバイス名> <ドメイン名>	
オプション	-f	強制的に仮想ネットワークデバイスを削除
使用例	ldm rm-vnet vnet0 ldom1	ドメイン「ldom1」の仮想ネットワークデバイス「vnet0」を削除
	ldm rm-vnet -f vnet0 ldom1	ドメイン「ldom1」の仮想ネットワークデバイス「vnet0」を強制的に削除
関連サブコマンド	add-vnet , remove-vswitch	
	list-domain , list-bindings , list-devices	

14. vf（SR-IOV 仮想デバイス(VF)の設定）**create-vf**（SR-IOV 仮想デバイス(VF)の作成）

省略形	-	
形式	ldm create-vf [オプション] <物理ポートのデバイス名 (PF 名) >	
オプション	-n number	作成する VF の数を指定
	-n max	指定した PF から最大数の VF を作成
	mac-addr=<number>	VF の MAC アドレスを指定
	pvid=<pvid_id>	VLAN ID をタグなしモードで指定
	vid=<vid_id1><,...>	VLAN ID をタグ付きモードで指定
	mac-addr=<number>	VF の MAC アドレスを指定
	alt-mac-addr=<auto number1> [,auto number2] [,...]	代替 MAC アドレスを指定 指定した代替 MAC アドレスの数だけ、Solaris ゾーンの仮想 NIC(vnic)を作成可能 auto を指定すると自動で MAC アドレスを生成 auto と number は混在可能
	mtu=<size>	最大転送単位 (MTU) を指定
	port-wwn=<value> node-wwn=<value>	ポート WWN とノード WWN を指定
	port-wwn=<value> node-wwn=<value>	ポート WWN とノード WWN を指定
使用例	ldm create-vf /SYS/PCI1/IOVFC.PF0	PF「/SYS/PCI1/IOVFC.PF0」から VF を作成
	ldm create-vf -n 2 /SYS/PCI1/IOVFC.PF0	PF「/SYS/PCI1/IOVFC.PF0」から VF を 2 つ作成
	ldm create-vf port-wwn=10:00:00:14:4f:fc:01:00 node-wwn=20:00:00:14:4f:fc:01:00 /SYS/PCI1/IOVFC.PF0	PF「/SYS/PCI1/IOVFC.PF0」からポート WWN「10:00:00:14:4f:fc:01:00」とノード WWN「20:00:00:14:4f:fc:01:00」を指定して VF を作成
関連サブコマンド	destroy-vf , set-io , add-io	
	list-io , list-dependencies , list-netdev	

destroy-vf (SR-IOV 仮想デバイス(VF)の削除)

省略形	-	
形式	ldm destroy-vf [オプション] <物理ポートのデバイス名 (PF 名)>	
オプション	-n number	削除する VF の数を指定
	-n max	指定した PF から最大数の VF を削除
使用例	ldm destroy-vf /SYS/PCI1 /IOVFC.PF0.VF0	VF「/SYS/PCI1/IOVFC.PF0.VF0」を削除
関連サブコマンド	create-vf , remove-io	
	list-io , list-dependencies , list-netdev	

15.io (物理・仮想 I/O デバイスの設定)**add-io** (物理・仮想 I/O デバイスの割り当て)

省略形	-	
形式	ldm add-io [オプション] <PCIe バス (ルートコンプレックス) 名> <ドメイン名>	
	ldm add-io <PCIe バス PCI Express Direct I/O デバイス名 SR-IOV 仮想デバイス名> <ドメイン名>	
オプション	iov=on off	PCIe バスの PCI Express Direct I/O および SR-IOV 機能の有効 (on) / 無効 (off) を指定 デフォルトは「off」
使用例	ldm add-io pci@8600 ldom1	ドメイン「ldom1」に PCIe バス「pci@8600」を割り当て
	ldm add-io iov=on pci@8700 ldom2	ドメイン「ldom2」に PCI Express Direct I/O および SR-IOV 機能を有効にして PCIe バス「pci@8700」を割り当て
	ldm add-io /SYS/PCI0/IOVFC.PF0.VF0 ldom1	ドメイン「ldom1」に VF「/SYS/PCI0/IOVFC.PF0.VF0」を割り当て
関連サブコマンド	remove-io , create-vf	
	list-io	

set-io (物理・仮想 I/O デバイスの設定変更)

省略形	-	
形式	ldm set-io iov=<on off> <PCIe バス (ルートコンプレックス) 名>	
	ldm set-io [オプション] <VF 名>	
オプション	iov=on off	PCIe バスの PCI Express Direct I/O および SR-IOV 機能の有効(on)/無効(off)を変更 デフォルトは「off」
	pvid=<pvid_id>	タグなしモードの VLAN ID を変更
	vid=<vid_id1><,...>	タグ付きモードの VLAN ID を変更
	mac-addr=<number>	SR-IOV 仮想デバイス(VF)の MAC アドレスを変更
	alt-mac-addr=<auto number1> [,auto number2] [,...]	代替 MAC アドレスを変更 指定した代替 MAC アドレスの数だけ、Solaris ゾーンの仮想 NIC(vnic)を作成可能 auto を指定すると自動で MAC アドレスを生成 auto と number は混在可能
	mtu=<size>	最大転送単位(MTU)を変更
	port-wwn=<value> node-wwn=<value>	ポート WWN とノード WWN を変更
使用例	ldm set-io iov=off pci@8700	PCIe バス「pci@8700」の PCI Express Direct I/O および SR-IOV 機能を無効化
	ldm set-io port-wwn=10:00:00:14:4f:fc:00:00 node-wwn=20:00:00:14:4f:fc:00:00 /SYS/PCI0/IOVFC.PF0.VF0	VF「/SYS/PCI0/IOVFC.PF0.VF0」にポート WWN 「10:00:00:14:4f:fc:00:00」とノード WWN 「20:00:00:14:4f:fc:00:00」を設定
関連サブコマンド	create-vf	
	list-io	

remove-io (物理・仮想 I/O デバイスの割り当てを解除)

省略形	rm-io	
形式	ldm rm-io [オプション] <PCIe バス PCIe Direct I/O デバイス名 SR-IOV 仮想デバイス名> <ドメイン名>	
オプション	-n	コマンドが成功するかどうかを事前に確認
使用例	ldm rm-io -n pci@8600 ldom1	ドメイン「ldom1」から PCIe バス「pci@8600」の割り当てを解除
	ldm rm-io /SYS/PCI0/IOVFC.PF0.VF0 ldom1	ドメイン「ldom1」から VF「/SYS/PCI0/IOVFC.PF0.VF0」の割り当てを解除
関連サブコマンド	add-io , destroy-vf	
	list-io	

16. vconsole（仮想コンソールのオプションの設定）

set-vconsole（仮想コンソールの設定）

省略形	set-vcons	
形式	ldm set-vcons [オプション] <ドメイン名>	
オプション	port=[port-number]	ドメインのコンソールに使用するポート番号を指定
	group=group	コンソールグループ名を指定（同一のポートで、複数のドメインのコンソールに接続）
	service=vcc-server	仮想コンソールサービスを指定
	log=<on off>	仮想コンソールのログギングの有効（on）／無効（off）を指定 デフォルトは「on」
使用例	ldm set-vcons port=5010 ldom1	ドメイン「ldom1」仮想コンソールポート「5010」を設定
	ldm set-vcons service=primary-vcc1 ldom2	ドメイン「ldom2」の仮想コンソールサービスに「primary-vcc1」を設定
関連サブコマンド	add-vconscon , set-vconscon	
	list-domain , list-bindings	

17. variable（環境変数の設定）

add-variable（環境変数の追加）

省略形	add-var	
形式	ldm add-var <オプション> <ドメイン名>	
オプション	variable 名 1=<value1> [variable 名 2=<value2> [...]]	環境変数を指定
使用例	ldm add-var auto-boot?=true ldom1	ドメイン「ldom1」の環境変数「auto-boot」を有効化
	ldm add-var boot-device=vdisk0 ldom1	ドメイン「ldom1」の環境変数「boot-device」に「vdisk0」を設定
関連サブコマンド	set-variable , remove-variable	
	list-domain , list-bindings	

set-variable（環境変数の設定）

省略形	set-var	
形式	ldm set-var <オプション> <ドメイン名>	
オプション	variable 名 1=<value1> [...]	環境変数を指定
使用例	ldm set-var auto-boot?=false boot-device=vdisk0 ldom1	ドメイン「ldom1」の環境変数「auto-boot」を無効化し、「boot-device」を「vdisk0」に設定
関連サブコマンド	add-variable	
	list-domain , list-bindings	

remove-variable（環境変数の削除）

省略形	rm-var	
形式	ldm rm-var <オプション> <ドメイン名>	
オプション	<variable 名 1> [variable 名 2] [...]	環境変数を指定
使用例	ldm rm-var auto-boot? ldom1	ドメイン「ldom1」の環境変数「auto-boot」を削除
関連サブ コマンド	add-variable	
	list-domain , list-bindings	

18. migrate (ドメインの移行)

migrate-domain (ドメインの移行)

省略形	migrate	
形式	ldm migrate [オプション] <移行対象ドメイン名> [<移行先ユーザー名>@<移行先ホスト名>[:移行先ドメイン名]]	
オプション	-f	強制的にドメインを移行
	-n	移行の予行演習を実行
	-c	信頼できる SSL 証明書を使用してドメインを移行 -p オプションとの併用不可 「ユーザー名の指定 ([<移行先ユーザー名>@])」との併用不可
	-p <passwd_file>	移行先ホストの root パスワードが記述されたファイルを指定し、パスワードを自動読取
	-s <config 名> *Oracle VM 3.5 の追加オプションです。	移行後、指定した config 名で構成情報を保存 移行元と移行先で同じ config 名で保存される
	mblockmap=<PA:PA>[,<PA:PA>,...] *Oracle VM 3.5 の追加オプションです。	物理メモリアドレスを指定してドメインを移行
	cidmap=<core-ID:core-ID>[,<core-ID:core-ID>,...] *Oracle VM 3.5 の追加オプションです。	CPU コア ID を指定してドメインを移行
	<移行先ホスト名>	移行先のホスト名を指定 IP アドレスでの指定も可能
	[:移行先ドメイン名]	ドメイン名の変更 移行元のドメイン名を移行先で変更
使用例	ldm migrate -n -p passfile ldom1 root@M10-1-02	ユーザー名「root」でホスト「M10-1-02」へのドメイン「ldom1」移行の予行演習を実行 移行先パスワードはファイル名「passfile」で自動読取を指定
	ldm migrate ldom1 192.168.10.20 :ldom2	ドメイン「ldom1」をドメイン名「ldom2」に変更し、ホスト「192.168.10.20」に移行
	ldm migrate -c ldom1 cidmap=8:16,9:17 mblockmap=0x400000000:c00000000 M10-1-02	信頼できる SSL 証明書を使用し、以下の設定でドメイン「ldom1」をホスト「M10-1-02」へ移行 • CPU コア ID 移行元: 8 と 9、移行先: 16 と 17 • 物理メモリアドレス 移行元: 0x400000000、移行先: c00000000
関連サブコマンド	cancel-operation	
	list-domain	

19. operation（操作の取消）

cancel-operation（操作の取消）

省略形	cancel-op	
形式	ldm cancel-op [オプション] <ドメイン名>	
オプション	reconf	遅延再構成の取消
	migration	マイグレーション操作の取消
	memdr	メモリ動的再構成削除の取消
使用例	ldm cancel-op reconf primary	ドメイン「primary」の遅延再構成を取消
	ldm cancel-op migration ldom1	ドメイン「ldom1」のマイグレーション操作を取消
関連サブコマンド	migrate-domain , start-reconf	
	list-domain	

20. policy（CPU Dynamic Resource Management の設定）

add-policy（CPU Dynamic Resource Management ポリシーの設定）

省略形	-	
形式	ldm add-policy [オプション] name=<ポリシー名> <ドメイン名>...	
オプション	enable=yes no	ポリシーの有効(yes)／無効(no)を指定
	priority=<value>	複数ポリシーの実行優先順位を指定 数値が小さいほど優先度高
	attack=<value>	仮想 CPU を追加する際の単位を指定
	decay=<value>	仮想 CPU を削除する際の単位を指定
	sample-rate=<value>	負荷をサンプリングする周期を秒単位で指定
	tod-begin=<hh>:<mm>[:<ss>]	ポリシーが有効となる時間(hh:時、mm:分、ss:秒)を指定
	tod-end=<hh>:<mm>[:<ss>]	ポリシーが無効となる時間(hh:時、mm:分、ss:秒)を指定
	util-lower=<percent>	CPU 使用率の下限を%で指定
	util-upper=<percent>	CPU 使用率の上限を%で指定
	vcpu-min=<value>	仮想 CPU の最小値を指定
	vcpu-max=<value>	仮想 CPU の最大値を指定
使用例	ldm add-policy enable=yes priority=1 attack=8 sample-rate=8 name=policy_1 primary	以下の設定でドメイン「primary」にポリシー「policy_1」を設定 - 優先度「1」 - 仮想 CPU の追加単位を「8」 - 負荷のサンプリング周期を「8」
関連サブコマンド	set-policy , remove-policy	
	list-domain , list-bindings	

set-policy (CPU Dynamic Resource Management ポリシーの設定変更)

省略形	-	
形式	ldm set-policy [オプション] name=<ポリシー名> <ドメイン名>...	
オプション	enable=<yes no>	ポリシーの有効(yes)／無効(no)を変更
	priority=<value>	複数ポリシーの実行優先順位を変更 数値が小さいほど優先度高
	attack=<value>	仮想 CPU を追加する際の単位を変更
	decay=<value>	仮想 CPU を削除する際の単位を変更
	sample-rate=<value>	負荷をサンプリングする周期(秒単位)を変更
	tod-begin=<hh>:<mm>[:<ss>]	ポリシーが有効となる時間(hh:時、mm:分、ss:秒)を変更
	tod-end=<hh>:<mm>[:<ss>]	ポリシーが無効となる時間(hh:時、mm:分、ss:秒)を変更
	util-lower=<percent>	CPU 使用率の下限(%)を変更
	util-upper=<percent>	CPU 使用率の上限(%)を変更
	vcpu-min=<value>	仮想 CPU の最小値を変更
	vcpu-max=<value>	仮想 CPU の最大値を変更
使用例	ldm set-policy enable=no name=policy_1 primary	ドメイン「primary」のポリシー「policy_1」を無効化
	ldm set-policy enable=yes priority=1 tod-begin=09:00 tod-end=15:30 name=policy_2 ldom1	ドメイン「ldom1」のポリシー「policy_2」を以下の設定に変更 - 優先度「1」 - ポリシーが有効となる時間「09:00」 - ポリシーが無効となる時間「15:00」
関連サブ コマンド	add-policy	
	list-domain , list-bindings	

remove-policy (CPU Dynamic Resource Management ポリシーの削除)

省略形	rm-policy	
形式	ldm rm-policy [name=]<ポリシー名> ... <ドメイン名>	
使用例	ldm rm-policy name=policy_1 primary	ドメイン「primary」からポリシー「policy_1」を削除
	ldm rm-policy policy_2 ldom1	ドメイン「ldom1」からポリシー「policy_2」を削除
関連サブ コマンド	add-policy	
	list-domain , list-bindings	

21. list (各情報のリスト表示)

list-domain (ドメインの構成情報の表示)

省略形	ls-dom	
形式	ldm ls-dom [オプション] [ドメイン名]	
オプション	-e	仮想サービスや仮想デバイスを含む情報を表示
	-l	ドメイン構成の詳細を表示
	★-S	CPU やメモリなどのリソースのステータスを表示 -e, -l とともに使用 (正常時は「ok」、障害時は「fail」を表示)
	-o <format: console,core,cpu,crypto,disk, domain,memory,network, physio, resmgmt,serial,status>	特定の情報 (CPU やメモリなど) のみを表示
	-p	スクリプトなどで解析可能な形式で表示
使用例	ldm ls-dom	ドメインの構成情報を表示
	ldm ls-dom -S -o cpu	CPU リソースのステータスを表示

list-services (構成済みの仮想サービスの表示)

省略形	ls-services	
形式	ldm ls-services [オプション] [ドメイン名]	
オプション	-e	自動設定のサービスやデバイスも含めて表示
	-p	スクリプトなどで解析可能な形式で表示
使用例	ldm ls-services	構成済みの仮想サービスを表示
	ldm ls-services -p primary	ドメイン「primary」の構成済みの仮想サービスをスクリプトなどで解析可能な形式で表示

list-bindings (ドメインに割り当てられているリソースの表示)

省略形	ls-bindings	
形式	ldm ls-bindings [オプション] [ドメイン名]	
オプション	-e	自動設定のサービスやデバイスも含めて表示
	-o network net *Oracle VM 3.4 の追加オプションです。	MAC アドレスなどのネットワーク情報を表示
	-p	スクリプトなどで解析可能な形式で表示
使用例	ldm ls-bindings	ドメインに割り当てられているリソースを表示
	ldm ls-bindings -p ldom1	ドメイン「ldom1」に割り当てられているリソースをスクリプトなどで解析可能な形式で表示

list-devices (リソースの表示)

省略形	ls-devices	
形式	ldm ls-devices [オプション]	
オプション	-a	すべてのリソースを表示 -a を指定しない場合、使用していないリソースのみを表示
	-p	スクリプトなどで解析可能な形式で表示
	★-S	リソースのステータスを表示(正常時は「ok」、障害時は「fail」を表示)
	core	CPU コアの情報を表示
	cpu	CPU スレッドの情報を表示
	memory	メモリの情報を表示
	io	物理 I/O デバイス情報を表示
使用例	ldm ls-devices	使用していないリソースを表示
	ldm ls-devices -a	すべてのリソースを表示
	ldm ls-devices memory	メモリの情報を表示

list-constraints (環境設定情報の表示)

省略形	ls-constraints	
形式	ldm ls-constraints [-e] [-p] [ドメイン名]	
	ldm ls-constraints [-x] [ドメイン名]	
オプション	-e	自動設定のサービスやデバイスも含めて表示
	-p	スクリプトなどで解析可能な形式で表示
	-x	環境設定情報を xml 形式で出力
使用例	ldm ls-constraints	環境設定情報を表示
	ldm ls-constraints -x ldom1	ドメイン「ldom1」の環境設定情報を xml 形式で表示
関連サブコマンド	add-domain , init-system	

list-variable (環境変数の一覧表示)

省略形	list-var	
形式	ldm list-var [オプション] <ドメイン名>	
オプション	variable 名 1 variable 名 2 ...	一覧表示する環境変数を指定する
使用例	ldm list-var auto-boot? boot-device ldom1	ドメイン「ldom1」の環境変数(自動起動とブートデバイス)を表示
関連サブコマンド	add-variable , set-variable , remove-variable	

list-io（物理・仮想 I/O デバイスの表示）

省略形	ls-io	
形式	ldm ls-io [オプション] <PCIe バス PCIe Direct I/O デバイス名 SR-IOV 物理デバイス名>	
オプション	-l	PCIe バスと PCIe Direct I/O デバイスの関係などの詳細を表示
	-p	スクリプトなどで解析可能な形式で表示
使用例	ldm ls-io	物理・仮想 I/O デバイスを表示
	ldm ls-io -l	物理・仮想 I/O デバイスの詳細を表示
関連サブコマンド	add-io , set-io , remove-io	

list-permits（CPU コアアクティベーションの情報を一覧表示）

省略形	ls-permits	
形式	★ ldm ls-permits	
使用例	ldm ls-permits	CPU コアアクティベーションの情報を一覧表示
関連サブコマンド	list-domain , list-bindings	

list-dependencies（仮想的なデバイス(vdisk, vnet, VF)の提供元と提供先のドメインを表示）

省略形	ls-dependencies	
形式	ldm ls-dependencies [オプション] [ドメイン名]	
オプション	-l	仮想デバイス名を表示
	-r	提供元と提供先の順に詳細を表示
	-p	スクリプトなどで解析可能な形式で表示
使用例	ldm ls-dependencies	仮想的なデバイスの提供元と提供先のドメインを表示
	ldm ls-dependencies -l -r primary	ドメイン「primary」の仮想的なデバイスを提供元と提供先の順に仮想デバイス名を表示
関連サブコマンド	add-vdisk , set-vdisk , remove-vdisk , add-vnet , set-vnet , remove-vnet , add-io , set-io , remove-io	

list-netdev (ネットワークデバイスを一覧表示)

省略形	ls-netdev	
形式	ldm ls-netdev [オプション] [ドメイン名]	
オプション	-l	IP アドレス、MAC アドレスなどの詳細を一覧表示
	-b	物理ネットワークデバイスのみ表示
	-o <net-device>	指定したネットワークデバイスのみ表示 -o を指定する場合はドメイン名も指定すること
	-p	スクリプトなどで解析可能な形式で表示
使用例	ldm ls-netdev	ネットワークデバイスを一覧表示
	ldm ls-netdev -l -o net0 primary	ドメイン「primary」のネットワークデバイス「net0」の詳細を一覧表示
関連サブコマンド	add-vswitch , set-vswitch , remove-vswitch , add-vnet , set-vnet , remove-vnet	

list-netstat (ネットワークデバイス統計情報を一覧表示)

省略形	ls-netstat	
形式	ldm ls-netstat [オプション] [ドメイン名]	
オプション	-u <R K M G>	出力の単位を設定 • B(バイト) 単位の場合は、「R」を指定 • KB 単位の場合は、「K」を指定 • MB 単位の場合は、「M」を指定 • GB 単位の場合は、「G」を指定
	-o <net-device>	指定したネットワークデバイスのみ表示 -o を指定する場合はドメイン名も指定すること
	-t <interval>	統計する間隔を指定(デフォルトは 1 秒) -t を指定する場合はドメイン名も指定すること
	-c <count>	統計する回数(1 以上)を指定 -c を指定する場合は-t とドメイン名も指定すること
	-p	スクリプトなどで解析可能な形式で表示
使用例	ldm ls-netstat	ネットワークデバイス統計情報を一覧表示
	ldm ls-netstat -o net0 -t 1 -c 3 primary	ドメイン「primary」のネットワークデバイス「net0」の統計情報を以下の条件で一覧表示 • 統計間隔「1 秒」 • 統計回数「3 回」
関連サブコマンド	add-vswitch , set-vswitch , remove-vswitch , add-vnet , set-vnet , remove-vnet	

list-history (ldm コマンドの実行履歴を表示)

*Oracle VM 3.5 の追加コマンドです。

直近 10 回のコマンド実行結果を表示します。set-logctl コマンドで回数を変更できます。

省略形	ls-history	
形式	ldm ls-history	
使用例	ldm ls-history	ldm コマンドの実行履歴を表示
関連サブコマンド	set-logctl	
	list-logctl	

22. system (ドメイン構成の復元)

init-system (ドメインの構成の復元)

省略形	-	
形式	ldm init-system [オプション] -i <xml ファイル名>	
オプション	-i	ドメインの復元に使用する xml ファイルを指定
	-f	ドメインの構成チェックを省略
	-r	ドメイン復元後、自動でシステムを再起動
	-s	仮想サービス (vds, vcc, vsw) のみを復元
使用例	ldm init-system -r -i /OVM/alldomains.xml	xml ファイル「/OVM/alldomains.xml」でドメインを復元し、復元後に自動でシステムを再起動
関連サブコマンド	list-constraints , list-domain , list-bindings	

23. reconf (遅延再構成モードの設定)

start-reconf (遅延再構成状態の開始)

省略形	-	
形式	ldm start-reconf <ドメイン名>	
使用例	ldm start-reconf primary	ドメイン「primary」を遅延再構成状態にする
関連サブコマンド	cancel-reconf , cancel-operation	
	list-domain , list-bindings	

cancel-reconf (遅延再構成状態の取消)

省略形	-	
形式	ldm cancel-reconf <ドメイン名>	
使用例	ldm cancel-reconf primary	ドメイン「primary」の遅延再構成状態を取消
関連サブコマンド	start-reconf	
	list-domain , list-bindings	

24. logctl (Oracle VM のログに出力されるロギング機能の設定)

set-logctl (Oracle VM のログに出力されるロギング機能の設定)

*Oracle VM 3.5 の追加コマンドです。

省略形	-	
形式	ldm set-logctl [オプション]	
オプション	history=<number>	ldm history で表示される履歴の数を指定 デフォルトは「10」
	notice=[on off]	警告メッセージをログに記録する (on) / 記録しない (off) を指定 デフォルトは「on」
	info=[on off]	情報メッセージをログに記録する (on) / 記録しない (off) を指定 デフォルトは「on」
	cmd=[on off resp]	CLS/XML コマンド行をログに記録する (on) / 記録しない (off) を指定 コマンドとコマンド応答をログに記録する場合は「resp」を指定 デフォルトは「off」
	debug=[on off]	デバッグメッセージをログに記録する (on) / 記録しない (off) を指定 デフォルトは「off」
	defaults	ロギング機能の設定をデフォルト値に戻す history をデフォルト値に戻す場合は「ldm set-logctl history=10 (または空白)」を実行する
使用例	ldm set-logctl history=20	ldm history で表示する履歴の数を「20」に設定
	ldm set-logctl cmd=resp	CLS/XML コマンドとコマンド応答をログに記録
	ldm set-logctl defaults	ロギング機能の設定をデフォルト値に戻す
関連サブコマンド	list-history , list-logctl	

list-logctl (Oracle VM のログに出力されるロギング機能の表示)

*Oracle VM 3.5 の追加コマンドです。

省略形	ls-logctl	
形式	ldm ls-logctl [-a] [-d] [ログの種類]	
オプション	-a	設定値をすべて表示 ログの種類オプションとの併用不可
	-d	デフォルト値を表示
	ログの種類	表示させるログの種類を指定 (fatal, warning, notice, info, cmd, debug) -a オプションとの併用不可
使用例	ldm ls-logctl -a	ロギング機能の設定値をすべて表示
関連サブコマンド	set-logctl	
	list-history	

25. **Idmconsole** (コンソール接続可能なドメイン/コンソールグループの表示と接続)

Idmconsole (コンソール接続可能なドメイン/コンソールグループの表示と接続)

*Oracle VM 3.6 の追加コマンドです。

ドメイン名またはコンソールグループ名による接続が可能です。

省略形	-	
形式	Idmconsole [オプション]	
	Idmconsole <ドメイン名>	
オプション	-l	コンソール接続可能なドメイン/コンソールグループの表示
使用例	Idmconsole	コンソール接続可能なドメイン/コンソールグループを表示
	Idmconsole Idom1	ドメイン「Idom1」にコンソール接続
関連サブコマンド	start-domain	
	list-domain	

改版履歴

改版年月	版数	改版内容
2013 年 2 月	第 1.0 版	新規作成 Oracle VM Server for SPARC 3.0 用に作成
2014 年 1 月	第 1.1 版	Oracle VM Server for SPARC 3.1 用に更新
2016 年 3 月	第 1.2 版	全コマンドの見直し、修正
2016 年 8 月	第 3.4 版	Oracle VM Server for SPARC 3.4 用に更新
2019 年 12 月	第 3.6 版	Oracle VM Server for SPARC 3.6 用に更新 レイアウトデザインを更新

