

# Dynamic Multi-Pathing 7.1 管理者ガイド - Linux

# Dynamic Multi-Pathing 管理者ガイド

最終更新: 2016-06-06

マニュアルのバージョン: 7.1 Rev 0

## 法的通知と登録商標

Copyright © 2016 Veritas Technologies LLC. All rights reserved.

Veritas、Veritas ロゴ、および NetBackup は、Veritas Technologies LLC または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。その他の会社名、製品名は各社の登録商標または商標です。

この製品には、サードパーティへの著作権を示す必要のあるサードパーティのソフトウェアが含まれる場合があります（「サードパーティプログラム」）。一部のサードパーティプログラムは、オープンソースまたはフリーウェアのライセンスの下で利用できます。このソフトウェアに付属の使用許諾契約によって、このようなオープンソースまたはフリーウェアのライセンスでお客様が有することのできる権利または義務は変更されないものとします。この Veritas 製品に伴うサードパーティの法的通知と登録商標の文書、または以下を参照してください。

<https://www.veritas.com/about/legal/license-agreements>

本書に記載の製品は、ライセンスに基づいて配布され、使用、コピー、配布、逆コンパイル、リバースエンジニアリングはそのライセンスによって制限されます。本書のいかなる部分も、Veritas Technologies LLC とそのライセンサーの書面による事前の許可なく、いかなる形式、方法であっても複製することはできません。

本書は「現状有姿のまま」提供され、商品性、特定目的への適合性、不侵害の黙示的な保証を含む、すべての明示的または黙示的な条件、表明、保証は、この免責が法的に無効であるとみなされないかぎり、免責されるものとします。VERITAS TECHNOLOGIES LLC は、本書の供給、性能、使用に関係する付随的または間接的損害に対して責任を負わないものとします。本書に記載の情報は、予告なく変更される場合があります。

ライセンスソフトウェアおよびマニュアルは、FAR 12.212 の規定によって商業用コンピュータソフトウェアと見なされ、Veritas 社によりオンプレミスで提供されるかホストされたサービスとして提供されるかに関わらず、FAR Section 52.227-19「Commercial Computer Software - Restricted Rights」および DFARS 227.7202「Commercial Computer Software and Commercial Computer Software Documentation」、その他の後継規制の規定により制限された権利の対象となります。使用許諾されたソフトウェアおよび文書の米国政府による修正、再生リリース、履行、表示または開示は、この契約の条件に従って行われます。

Veritas Technologies LLC  
500 E Middlefield Road  
Mountain View, CA 94043

<http://www.veritas.com>

## テクニカルサポート

テクニカルサポートはグローバルにサポートセンターを管理しています。すべてのサポートサービスは、サポート契約と現在のエンタープライズテクニカルサポートポリシーに応じて提供されます。サポート内容およびテクニカルサポートの利用方法に関する情報については、次の **Web** サイトにアクセスしてください。

<https://www.veritas.com/support>

Veritas Account 情報は、次の URL で管理できます。

<https://my.veritas.com>

現在のサポート契約についてご不明な点がある場合は、次に示すお住まいの地域のサポート契約管理チームに電子メールでお問い合わせください。

世界中 (日本以外)

[CustomerCare@veritas.com](mailto:CustomerCare@veritas.com)

日本

[CustomerCare\\_Japan@veritas.com](mailto:CustomerCare_Japan@veritas.com)

## マニュアル

マニュアルが最新版であることを確認してください。各マニュアルの 2 ページ目には最新更新日が記載されています。マニュアルのバージョンは各ガイドの 2 ページ目に記載されています。最新のマニュアルはベリタスの **Web** サイトで入手できます。

<https://sort.veritas.com/documents>

## マニュアルに関するご意見やご感想

ご意見、ご感想をお待ちしています。マニュアルに対する改善点の提案や誤植や抜けについての報告をお願いします。送信の際は、マニュアルの題名とバージョン、章、セクションのタイトルを明記してください。次の宛先にお送りください。

[doc.feedback@veritas.com](mailto:doc.feedback@veritas.com)

また、ベリタスのコミュニティサイトで、マニュアル情報を確認したり質問したりできます。

<http://www.veritas.com/community/>

## Veritas SORT (Services and Operations Readiness Tools)

Veritas SORT (Services and Operations Readiness Tools) は、時間のかかる特定の管理タスクを自動化および単純化するための情報とツールを提供する **Web** サイトです。製品に応じて、SORT はインストールとアップグレードの準備、データセンターのリスクの識別、効率性の改善に役立ちます。使用している製品に対して SORT が提供しているサービスおよびツールについては、次のデータシートを参照してください。

[https://sort.veritas.com/data/support/SORT\\_Data\\_Sheet.pdf](https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf)

# 目次

<b>第 1 章</b>	<b>DMP について</b> .....	8
	DMP (Dynamic Multi-Pathing) について .....	8
	DMP の動作方法 .....	9
	パスでの I/O を DMP で監視する方法 .....	13
	負荷分散 .....	14
	クラスタ環境における DMP .....	15
	マルチコントローラ ALUA のサポート .....	16
	ディスクアレイへの複数パス .....	16
	デバイス検出 .....	17
	ディスクデバイス .....	17
	DMP でのディスクデバイスの命名 .....	18
	オペレーティングシステムに基づく名前の付け方について .....	18
	エンクロージャに基づく名前の付け方について .....	19
<b>第 2 章</b>	<b>ネーティブデバイス管理のための DMP の設定</b> .....	24
	ネーティブデバイス管理のための DMP の設定について .....	24
	ネーティブマルチパス設定の表示 .....	26
	DMP への LVM ボリュームグループの移行 .....	26
	EMC PowerPath から DMP への移行 .....	27
	Hitachi Data Link Manager (HDLM) から DMP への移行 .....	28
	Linux Device Mapper Multipath から DMP への移行 .....	29
	Oracle Automatic Storage Management (ASM) での Dynamic Multi-Pathing (DMP) デバイスの使用 .....	30
	Oracle Automatic Storage Management (ASM) で使用可能にする ための Dynamic Multi-Pathing (DMP) デバイスの有効化 .....	31
	Oracle Automatic Storage Management (ASM) ディスクのリストから の Dynamic Multi-Pathing (DMP) デバイスの削除 .....	32
	オペレーティングシステムデバイス上の Oracle Automatic Storage Management (ASM) ディスクグループの Dynamic Multi-Pathing (DMP) デバイスへの移行 .....	33
	既存の LVM ボリュームグループへの DMP デバイスの追加または LVM ボリュームグループの新規作成 .....	36
	ネーティブデバイスの DMP サポートの削除 .....	38

## 第 3 章

<b>DMP の管理</b> .....	40
コントローラとストレージプロセッサに対する I/O の有効化と無効化について .....	40
DMP データベース情報の表示について .....	41
ディスクへのパスの表示 .....	41
DMP ノードのカスタム名の設定 .....	43
vxddmpadm ユーティリティを使った DMP の管理 .....	44
DMP ノードに関する情報の取得 .....	46
DMP ノードについての統合された情報の表示 .....	47
LUN グループのメンバーの表示 .....	48
DMP ノード、コントローラ、エンクロージャ、アレイポートによって制御されるパスの表示 .....	49
コントローラに関する情報の表示 .....	51
エンクロージャに関する情報の表示 .....	53
アレイポートに関する情報の表示 .....	53
ALUA アレイのためのユーザーフレンドリな CLI 出力 .....	54
サードパーティ製のドライバにより制御されるデバイスに関する情報の表示 .....	55
拡張デバイス属性の表示 .....	56
VxVM の制御下におけるデバイスの無効化と有効化 .....	58
I/O 統計情報の収集と表示 .....	59
エンクロージャへのパスに関する属性の設定 .....	65
デバイスまたはエンクロージャの冗長レベルの表示 .....	66
アクティブパスの最小数の指定 .....	67
I/O ポリシーの表示 .....	68
I/O ポリシーの指定 .....	68
パス、コントローラ、アレイポート、DMP ノードに対する I/O の無効化 .....	74
パス、コントローラ、アレイポート、DMP ノードに対する I/O の有効化 .....	75
エンクロージャ名の変更 .....	76
I/O エラーに対する応答の設定 .....	76
I/O 調整機構の設定 .....	78
サブパスフェールオーバーグループ (SFG) の設定 .....	79
LIPP (Low-Impact Path Probing) の設定 .....	80
リカバリオプション値の表示 .....	80
DMP パスリストアポリシーの設定 .....	81
DMP パスリストアスレッドの停止 .....	83
DMP パスリストアスレッドの状態の表示 .....	83
アレイポリシーモジュール (Array Policy Module) の設定 .....	83

<b>第 4 章</b>	<b>ディスクの管理</b> .....	85
	ディスク管理について .....	85
	新しく追加されたディスクデバイスの検出と設定 .....	85
	部分的なデバイス検出 .....	86
	ディスクの検出とディスクアレイの動的な追加について .....	87
	サードパーティドライバの共存について .....	89
	デバイス検出層の管理方法 .....	90
	ディスクデバイスの名前の付け方の変更 .....	102
	ディスクの名前の付け方の表示 .....	104
	永続的なデバイス名の再生成 .....	104
	サードパーティ製ドライバ制御のエンクロージャに対するデバイスの命 名の変更 .....	105
	エンクロージャに基づくディスク名と OS に基づくディスク名の関連付けの 検出 .....	106
<b>第 5 章</b>	<b>デバイスの動的再設定</b> .....	107
	オンラインの Dynamic Reconfiguration について .....	107
	Dynamic Reconfiguration ツールでの DMP の制御下にある LUN のオン ラインでの再設定 .....	108
	Dynamic Reconfiguration ツールでの既存のターゲット ID からの LUN の動的な削除 .....	108
	Dynamic Reconfiguration ツールでのターゲット ID への新しい LUN の動的な追加 .....	110
	Dynamic Reconfiguration ツールでの既存のターゲット ID からの LUN の動的な交換 .....	111
	DMP の制御下にある LUN のオンラインでの手動での再設定 .....	112
	既存のターゲット ID から LUN を動的に手動での削除 .....	113
	新しいターゲット ID に新しい LUN を動的に手動での追加 .....	115
	オペレーティングシステムのデバイスツリーがクリーンアップされていな い場合のターゲット ID 再利用の検出について .....	116
	LUN の追加または削除後のオペレーティングシステムデバイスツリー のスキャン .....	117
	LUN の削除後のオペレーティングシステムデバイスツリーの手動での クリーンアップ .....	118
	アレイ側からの LUN の特性の変更 .....	118
	Dynamic Reconfiguration ツールでのホストバスアダプタのオンラインでの 交換 .....	119
	アレイコントローラファームウェアのオンラインでのアップグレード .....	120

<b>第 6 章</b>	<b>イベント監視</b> .....	122
	Dynamic Multi-Pathing (DMP) のイベントソースデーモン (vxesd) について .....	122
	ファブリック監視と予防的なエラー検出 .....	123
	Dynamic Multi-Pathing (DMP) の iSCSI および SAN ファイバーチャネル トポロジーの検出 .....	124
	DMP イベントログ .....	124
	Dynamic Multi-Pathing (DMP) のイベントソースデーモンの起動と停止 .....	125
<b>第 7 章</b>	<b>処理効率の監視とチューニング</b> .....	126
	テンプレートを使った DMP (Dynamic Multi-Pathing) のチューニングについて .....	126
	DMP チューニングテンプレート .....	127
	DMP チューニングテンプレートの例 .....	128
	設定属性テンプレートを使った DMP のホストのチューニング .....	131
	DMP 設定ファイルの管理 .....	133
	DMP のチューニングパラメータと属性のデフォルト値へのリセット .....	133
	テンプレートでサポートされる DMP のチューニングパラメータと属性 .....	133
	DMP チューニングパラメータ .....	134
<b>付録 A</b>	<b>DMP のトラブルシューティング</b> .....	141
	DMP からパスを除外または DMP にパスを追加するときのエラーのリカバリ .....	141
	アレイサポートのダウングレード .....	143
<b>付録 B</b>	<b>参照</b> .....	144
	Veritas コマンドに対するコマンド入力補完機能 .....	144
<b>用語集</b> .....		146
<b>索引</b> .....		153

# DMP について

この章では以下の項目について説明しています。

- [DMP \(Dynamic Multi-Pathing\) について](#)
- [DMP の動作方法](#)
- [マルチコントローラ ALUA のサポート](#)
- [ディスクアレイへの複数パス](#)
- [デバイス検出](#)
- [ディスクデバイス](#)
- [DMP でのディスクデバイスの命名](#)

## DMP (Dynamic Multi-Pathing) について

Dynamic Multi-Pathing (DMP) は、システム上で設定されているオペレーティングシステムのネイティブデバイスに対するマルチパス機能を提供します。DMP は DMP メタデバイス (既知の DMP ノード) を作成して、同じ物理 LUN へのデバイスパスをすべて示します。

DMP メタデバイスは OS ネイティブの論理ボリュームマネージャ (LVM) をサポートします。DMP メタデバイス上に LVM ボリュームとボリュームグループを作成できます。

Veritas Volume Manager (VxVM) ボリュームとディスクグループは、LVM ボリュームおよびボリュームグループと共存できますが、各デバイスは 1 つのタイプしかサポートできません。ディスクに VxVM ラベルが付いている場合、そのディスクは LVM で利用できません。同様に、ディスクが LVM によって使用中の場合、そのディスクは VxVM で利用できません。

## DMP の動作方法

DMP (Dynamic Multi-Pathing) では、パスフェールオーバー機能と負荷分散機能を使って、可用性、信頼性、パフォーマンスを向上します。これらの機能は、さまざまなベンダーのマルチポートディスクアレイに対応しています。

ディスクアレイは、複数のパスを介して、ホストシステムに接続することができます。ディスクへのさまざまなパスを検出するために、DMP では、対応している各アレイに特有の機構を使います。また、DMP では、DMP に対応していて同じホストシステムに接続されているアレイの様々なエンクロージャを識別します。

p.85 の「新しく追加されたディスクデバイスの検出と設定」を参照してください。

DMP で使われるマルチパスポリシーは、ディスクアレイの特性によって異なります。

DMP では、次の標準アレイタイプをサポートします。

表 1-1

アレイタイプ	説明
アクティブ/アクティブ (A/A)	複数のパスを同時に使って I/O を行うことができます。また DMP により、I/O 負荷が LUN への複数のパス上に均等に分散されるので I/O スループットが向上します。1 つのパスが失われた場合、DMP は自動的に、そのアレイに対して使える他のパスを介して I/O を行います。
非対称アクティブ/アクティブ (A/A-A)	A/A-A または非対称アクティブ/アクティブアレイは、パフォーマンスをほとんど低下させずにセカンダリストレージパスからアクセスできます。動作は、ALUA のアレイがサポートする SCSI コマンドをサポートしない点以外は ALUA と同じです。
非対称論理ユニットアクセス (ALUA)	DMP は ALUA のすべてのバリエーションをサポートします。

アレイタイプ	説明
<p>アクティブ/パッシブ (A/P)</p>	<p>通常の操作中に 1 つのコントローラ(アクセスポートまたはストレージプロセッサ)上のプライマリ(アクティブ)パス経由で LUN(論理ユニット番号。ハードウェアを使って作成される実際のディスクまたは論理ディスク)へのアクセスが可能です。</p> <p>非明示的フェールオーバーモード(auto-trespass モード)では、プライマリパスに障害が発生した場合、別のコントローラ上のセカンダリ(パッシブ)パスに I/O をスケジューリングすることによって、A/P アレイが自動的にフェールオーバーします。このパッシブポートは、アクティブポートに障害が発生するまで I/O には使われません。A/P アレイでは、プライマリパスで I/O 障害が発生すると、単一の LUN でパスのフェールオーバーが実行されます。</p> <p>このアレイモードは、複数のプライマリパスを 1 つのコントローラに持つことで、同時 I/O と負荷分散をサポートします。この機能は、複数のポートを持つコントローラにより、またはアレイとコントローラ間に SAN スイッチを挿入することによって、提供されます。セカンダリ(パッシブ)パスへのフェールオーバーは、すべてのアクティブなプライマリパスに障害が発生した場合にのみ実行されます。</p>
<p>明示的フェールオーバーモードまたは非 auto-trespass モードのアクティブ/パッシブ (A/PF)</p>	<p>LUN のセカンダリパスへのフェールオーバーを実行するには、該当するコマンドをアレイに発行する必要があります。</p> <p>このアレイモードは、複数のプライマリパスを 1 つのコントローラに持つことで、同時 I/O と負荷分散をサポートします。この機能は、複数のポートを持つコントローラにより、またはアレイとコントローラ間に SAN スイッチを挿入することによって、提供されます。セカンダリ(パッシブ)パスへのフェールオーバーは、すべてのアクティブなプライマリパスに障害が発生した場合にのみ実行されます。</p>

アレイタイプ	説明
LUN グループフェールオーバーが設定されたアクティブ/パッシブ (A/PG)	<p>LUN グループフェールオーバーが設定されたアクティブ/パッシブアレイ (A/PG アレイ) の場合、1 つのコントローラを介して接続されている LUN のグループは単一のフェールオーバーエンティティとして扱われず、A/P アレイの場合と異なり、フェールオーバーは個々の LUN レベルではなくコントローラレベルで実行されます。プライマリコントローラとセカンダリコントローラは、それぞれ別の LUN グループに接続されます。プライマリコントローラの LUN グループ内の LUN の 1 つに障害が発生した場合、そのグループ内のすべての LUN に対して、セカンダリコントローラへのフェールオーバーが実行されます。</p> <p>このアレイモードは、複数のプライマリパスを 1 つのコントローラに持つことで、同時 I/O と負荷分散をサポートします。この機能は、複数のポートを持つコントローラにより、またはアレイとコントローラ間に SAN スイッチを挿入することによって、提供されます。セカンダリ (パッシブ) パスへのフェールオーバーは、すべてのアクティブなプライマリパスに障害が発生した場合にのみ実行されます。</p>

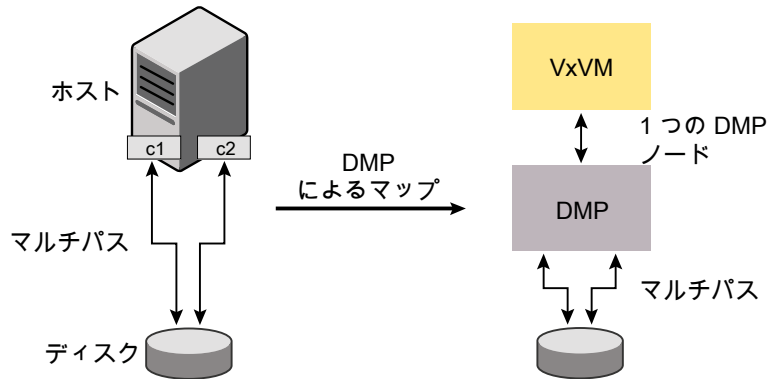
アレイポリシーモジュール (APM) では、DMP がサポートする標準タイプ以外のアレイタイプを DMP に定義できます。

Dynamic Multi-Pathing は、DMP メタノード (DMP ノード) を使って、システムに接続されているディスクデバイスにアクセスします。DMP に対応しているアレイ内のディスクの場合は、DMP により各ディスクに接続するパスセットに 1 つのノードがマップされます。さらに、DMP によりそのディスクアレイに適合するマルチパスポリシーがノードに関連付けられます。

DMP に対応していないアレイ内のディスクの場合は、DMP によりディスクに接続するパスそれぞれに、個別のノードがマップされます。ノードの raw デバイスおよびブロックデバイスは、ディレクトリ /dev/vx/rdmp および /dev/vx/dmp にそれぞれ作成されます。

図 1-1 では、DMP によりサポートされているディスクアレイ内のディスクにどのようにノードが設定されるかについて説明しています。

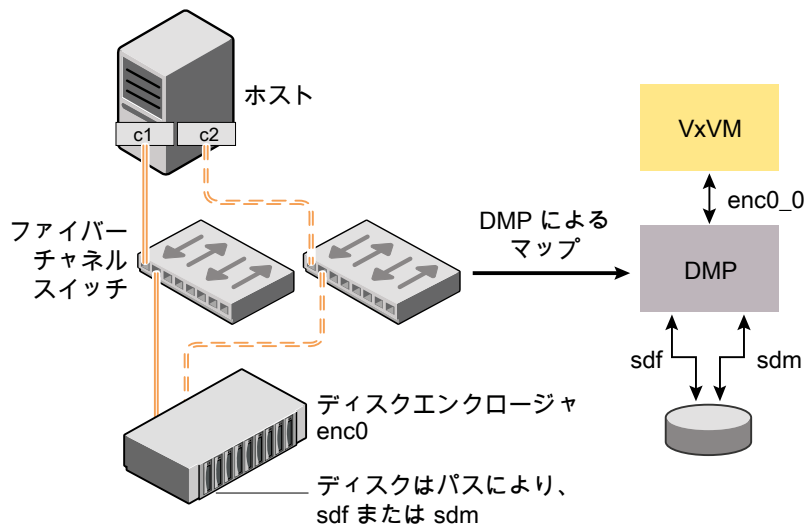
図 1-1 DMP で、ディスクに対する複数の物理パスを 1 つのノードで表す方法



DMP では、ディスクが属するアレイを識別することができるディスクデバイス名前の付け方を導入しました。

図 1-2 には、エンクロージャ内の 1 つのディスクに 2 つのパス (sdf と sdm) が存在し、VxVM により 1 つの DMP ノード (enc0\_0) を使ってディスクにアクセスされる例が示されています。

図 1-2 SAN 環境における、ディスクエンクロージャに対するマルチパスの例



p.19 の「エンクロージャに基づく名前の付け方について」を参照してください。

p.102 の「ディスクデバイスの名前の付け方の変更」を参照してください。

p.85 の「新しく追加されたディスクデバイスの検出と設定」を参照してください。

## パスでの I/O を DMP で監視する方法

リリース 5.0 より前の VxVM には、エラー処理を実行するカーネルデーモン(`error`)と、パシストアアクティビティを実行するカーネルデーモン(`restored`)がありました。

リリース 5.0 からは、DMP が、エラー処理、パシストア、統計情報収集、SCSI 要求コールバックなどのタスクに使うカーネルスレッドのプールを保守します。`restored` の名前は、下位互換性のために引き続き維持されています。

1 つのカーネルスレッドは、パスで I/O エラーが発生すると、そのパスに対応する HBA の精査を開始することで応答します。続いて、別のスレッドが HBA からの応答に従って適切な処理を行います。適用した処理は、そのパスでの I/O 要求を再試行することも、そのパスを破棄して代替パスで I/O をスケジューリング設定し直すこともできます。

リストアカーネルタスクは定期的(デフォルトでは 5 分間隔)に起動して、パスの健全性を調べ、リストアされたパスで I/O を再開します。パスによっては断続的にエラーが起きることがあるため、パスが一定期間(デフォルトでは 5 分)健全であり続けた場合にのみ、このパスで I/O を再開します。DMP ではパスのチェックに異なるポリシーを設定できます。

p.81 の「DMP パシストアポリシーの設定」を参照してください。

統計情報収集タスクは、各 I/O 要求の開始時間と終了時間、各パスでの I/O エラー数と再試行回数を記録します。この情報を使って、I/O 要求によって SCSI ドライバのフラッドが起きないように、DMP を設定できます。この機能を I/O 調整と呼びます。

I/O 要求がミラーボリュームに関連する場合、VxVM は、`FAILFAST` フラグを指定します。この場合、DMP はエラーになった I/O 要求をそのパス上で再試行する代わりに、エラーが起きたというマークをパス上のディスクに付けます。

p.13 の「パスフェールオーバー機構」を参照してください。

p.14 の「I/O 調整」を参照してください。

## パスフェールオーバー機構

複数パスを持つディスクアレイで DMP を使うと、システムの可用性が向上します。ディスクアレイへのパスの 1 つが失われた場合、DMP では、管理者が介入しなくても、I/O 要求に対して次に使用可能なパスが自動的に選択されます。

また、接続が修復または復元されたり、OS が正しくデバイスを認識している場合にシステムが完全に起動した後にデバイスの追加や削除が行われると、DMP に通知されます。

必要に応じて、パスでの I/O エラーに対する DMP の応答は、個々のアレイへのパスごとに調整できます。DMP では、I/O 要求が成功することなく一定時間が経過した場合、またはパスでの一定回数の再試行が失敗した場合に、I/O 要求を時間切れにするように設定できます。

p.76 の「[I/O エラーに対する応答の設定](#)」を参照してください。

## サブパスフェールオーバーグループ (SFG)

サブパスフェールオーバーグループ (SFG) は、まとめて失敗およびリストアできるパスのグループを表します。SFG のパスで I/O エラーが発生した場合、DMP は SFG のパス以外に、その他のパスでプロアクティブなパスのプローブを行います。この動作により、パスのフェールオーバーのパフォーマンスが大幅に向上し、結果として I/O パフォーマンスが向上します。サブパスフェールオーバーグループを形成するために DMP が現在従っている基準は、ホストからアレイまで同じエンドポイントを持つパスを、1 つの論理的なストレージフェールオーバーグループにまとめることです。

p.79 の「[サブパスフェールオーバーグループ \(SFG\) の設定](#)」を参照してください。

## LIPP (Low-Impact Path Probing)

DMP のリストアデーモンは、LUN パスを定期的にプローブし続けます。この動作は、パスで I/O が発生しない場合でも、DMP がパスを最新の状態に保つのに役立ちます。パスの状態がリストアデーモンによって更新されている間に行われるプローブの数を最適化するため、LIPP (Low-Impact Path Probing) はリストアデーモンにロジックを追加します。この最適化は、論理的なサブパスのフェールオーバーグループを使うことで実現されます。LIPP のロジックが導入された DMP は、サブパスフェールオーバーグループ (SFG) 内のすべてのパスをプローブする代わりに、SFG 内の限られた数のパスのみをプローブします。これらのプローブの結果に基づいて、DMP はその SFG 内のすべてのパスの状態を判断します。

p.80 の「[LIPP \(Low-Impact Path Probing\) の設定](#)」を参照してください。

## I/O 調整

I/O 調整を有効にし、応答動作が低下したパスでの未処理の I/O 要求数が増加した場合、未処理の I/O 要求数が一定値に達したとき、またはそのパスで最後に I/O 要求が成功してから一定時間が経過したときに、新しい I/O 要求をそのパスに送らないように DMP を設定できます。調整がパスに適用されると、そのパスでの新しい I/O 要求は、別の使用可能なパス上にスケジュール設定されます。この調整は、パスにエラーがないと HBA から報告があった場合、またはパスでの未処理の I/O 要求が成功した場合に、パスから削除されます。

p.78 の「[I/O 調整機構の設定](#)」を参照してください。

## 負荷分散

デフォルトでは、DMP は最小キュー I/O ポリシーを使ってすべてのアレイタイプのパス全体に負荷を分散します。負荷分散が行われると、使用可能なパスすべての総帯域幅を使って、I/O スループットが最大化されます。I/O は未処理の I/O が最小のパスを使って送信されます。

アクティブ/パッシブ (A/P) のディスクアレイ場合は、I/O はプライマリパスで送信されます。すべてのプライマリパスに障害が発生した場合、I/O は使用可能なセカンダリパスでの送信に切り替えられます。あるコントローラから別のコントローラへ連続して LUN 制御が移動し I/O 処理が極端に遅くなると、A/P ディスクアレイに対するプライマリ/セカンダリパスの負荷分散は、複数 I/O の同時処理をサポートしていないかぎり行われません。

その他のアレイでは、負荷分散は現在アクティブなパスのすべてで実行されます。

エンクロージャやディスクアレイへのパスに適用する I/O ポリシーは変更できます。この操作はサーバーに影響しない、ダウンタイムを必要としないオンライン操作です。

## クラスタ環境における DMP

A/P (アクティブ/パッシブ) タイプのディスクアレイを複数のホストで共有するクラスタ環境では、クラスタ内のすべてのノードが同一の物理ストレージコントローラポート経由でディスクにアクセスする必要があります。ディスクへのアクセスに複数のパスを同時に使うと、I/O パフォーマンスが大幅に低下します (ピンポン効果とも呼ばれます)。単一のクラスタノードでパスフェールオーバーが発生した場合でも、すべてのノードが継続して同一の物理パスを共有できるように、クラスタ全体が調整されます。

VxVM 4.1 より前のリリースでは、クラスタ化と DMP 機能は、A/P アレイでパスがリストアされたときに自動フェールバックを処理できず、明示的フェールオーバーモードアレイのフェールバックもサポートしていませんでした。フェールバックは、パス障害が修復された後に、各クラスタノードで `vxctl enable` コマンドを実行することにより、手動で実行する必要がありました。リリース 4.1 からは、フェールバックは、マスターノードによって調整され、クラスタ全体で自動的に実行されるようになりました。明示的フェールオーバーモードアレイの自動フェールバックも、適切な下位コマンドを実行することにより処理できます。

---

**メモ:** A/P アレイの自動フェールバックのサポートには、システムへの適切な ASL (Array Support Library) のインストールが必要です。APM (Array Policy Module) も必要になることがあります。

---

p.87 の「[ディスクの検出とディスクアレイの動的な追加について](#)」を参照してください。

アクティブ/アクティブタイプのディスクアレイの場合、すべてのディスクは、接続されたすべての物理パスを通じて同時にアクセスできます。クラスタ環境では、ノードが同じ物理パスでディスクにアクセスする必要はありません。

p.90 の「[デバイス検出層の管理方法](#)」を参照してください。

p.83 の「[アレイポリシーモジュール \(Array Policy Module\) の設定](#)」を参照してください。

## 共有ディスクグループでのコントローラの有効化と無効化について

VxVM (Veritas Volume Manager) 5.0 より前のリリースでは、共有 Veritas Volume Manager ディスクグループの一部であるディスクに接続されているパスまたはコントロー

ラを有効または無効にすることはできませんでした。VxVM 5.0 以降では、クラスタ内の共有 DMP ノードで、これらの操作をサポートします。

## マルチコントローラ ALUA のサポート

マルチコントローラ ALUA のサポートにより、次が可能になります。

- 複数のストレージコントローラを含む ALUA アレイDMP は ALUA 標準に準拠するストレージアレイをすでにサポートしていましたが、そのサポートは従来のデュアルストレージコントローラモデルに基づいていました。
- PATH-TYPE[M] 列にレガシーの PRIMARY 状態または SECONDARY 状態を表示する代わりに ALUA Asymmetric Access State (AAS) を表示するユーザーフレンドリな CLI 出力。ALUA アレイに対しては、DMP 管理インターフェースは次のように ALUA 状態を表示します。
  - アクティブ/最適化 (Active/Optimized)
  - アクティブ/非最適化 (Active/Non-optimized)
  - スタンバイ (Standby)
  - 利用不能 (Unavailable)
  - 進行中の移行 (TransitionInProgress)
  - オフライン (Offline)

---

**メモ:** `dmp_display_alua_states` チューニングパラメータのデフォルト値は `on` です。レガシーのパスタイプである PRIMARY または SECONDARY を表示するように表示モードを変更するには、`dmp_display_alua_states` チューニングパラメータを `off` にします。

---

## ディスクアレイへの複数パス

ディスクデバイスにアクセスする複数のポートを備えたディスクアレイもあります。これらのポートと、ホストバスアダプタ (HBA) コントローラとアレイにローカルなデータバスまたは I/O プロセッサを接続して、ディスクデバイスにアクセスする複数のハードウェアパスを作成できます。このようなディスクアレイをマルチパス化されたディスクアレイと呼びます。このタイプのディスクアレイは、多様な設定でホストシステムに接続できます (たとえば、シングルホスト上の異なるコントローラに接続された複数ポート構成、ホスト上の 1 つのコントローラを介したポートのチェーン構成、異なるホストに同時に接続されたポート構成など)。

p.9 の「[DMP の動作方法](#)」を参照してください。

## デバイス検出

デバイス検出は、ホストに接続されているディスクを検出するプロセスを示すために使う用語です。この機能は DMP にとって重要です。DMP では多くのベンダーにより増加し続けるディスクアレイをサポートする必要があるためです。ホストに接続されているデバイスを検出する機能とともに、デバイス検出サービスでは、新しいディスクアレイのサポートを追加できます。デバイス検出はデバイス検出層 (DDL) と呼ばれる機能を使います。

DDL により、再ブートすることなく新しいディスクアレイのサポートを追加できます。

## ディスクデバイス

デバイス名 (`devname` またはディスクアクセス名とも呼ばれる) には、ディスクデバイスの、オペレーティングシステムに認識される名前を定義します。

これらのデバイスは、場合によっては異なりますが、通常は `/dev` ディレクトリに配置されます。特定のベンダーのハードウェア固有のデバイスは、独自のパス名規則に従う場合があります。

VxVM は、オペレーティングシステムによって提供されるディスクパーティション設定規則をサポートしています。デバイス名の構文は、`hdx[N]` または `sdx[N]` です。この構文の中の `x` は、オペレーティングシステムから見た EIDE (`hd`) ディスクまたは SCSI (`sd`) ディスクの順序を示す英字で、`N` (オプション) はパーティション番号 (1 から 15 の範囲) です。たとえば `sda7` というデバイス名は、最初の SCSI ディスク上のパーティション 7 を表します。パーティション番号が省略されている場合、そのデバイス名はディスク全体を指します。

特定のベンダーのハードウェア固有のデバイスは、パス名が異なる場合があります。たとえば、COMPAQ SMART コントローラと SMARTII コントローラでは、それぞれ `/dev/ida/cXdXpX` と `/dev/cciss/cXdXpX` という形式のデバイス名を使います。

Dynamic Multi-Pathing (DMP) はデバイス名を使ってメタデバイスを `/dev/vx/[r]dmp` ディレクトリに作成します。DMP は、このメタデバイス (DMP ノード) を使って、1 つ以上の物理パス (異なるコントローラ経由の場合もあり) を介してアクセスできるディスクを表します。使えるアクセスパスの数は、ディスクが単体ディスクであるか、またはシステムに接続されているマルチポートディスクアレイに属するかにより異なります。

`vxdisk` ユーティリティを使って、DMP メタデバイスが包括するパスを表示したり、各パスの状態 (たとえば、有効状態か無効状態か) を表示できます。

p.9 の「[DMP の動作方法](#)」を参照してください。

デバイス名は、エンクロージャに基づく名前としてリマップすることもできます。

p.18 の「[DMP でのディスクデバイスの命名](#)」を参照してください。

## DMP でのディスクデバイスの命名

ディスクのデバイス名は DMP に指定する名前の付け方によって割り当てられます。デバイス名の形式は、ディスクのカテゴリが異なると変わります。

p.88 の「[ディスクカテゴリ](#)」を参照してください。

デバイス名は次の名前の付け方の 1 つを利用できます。

- オペレーティングシステムに基づく名前の付け方  
p.18 の「[オペレーティングシステムに基づく名前の付け方について](#)」を参照してください。

- エンクロージャに基づく名前の付け方  
p.19 の「[エンクロージャに基づく名前の付け方について](#)」を参照してください。

デバイス名が 31 文字より長いデバイスは、常にエンクロージャに基づく名前を使います。

デフォルトでは、DMP はエンクロージャに基づく名前の付け方を使います。必要な場合、ディスクデバイスの名前の付け方を変更できます。

p.102 の「[ディスクデバイスの名前の付け方の変更](#)」を参照してください。

## オペレーティングシステムに基づく名前の付け方について

OS に基づく名前の付け方では、すべてのディスクデバイスは `hd $x$ [ $N$ ]` 形式または `sd $x$ [ $N$ ]` 形式で名前が設定されます。 $x$  はオペレーティングシステムから見た EIDE (`hd`) ディスクまたは SCSI (`sd`) ディスクの順序を示す英字で、 $N$  (オプション) はパーティション番号 (1 から 15 の範囲) です。

DMP はディスクに、複数のパスからの DMP メタデバイスの名前 (ディスクアクセス名) を割り当てます。DMP は名前をアルファベット順にソートし、最初の名前を選択します。たとえば、`sdd` ではなく `sd $c$`  を選択します。この動作によって、デバイスとその下位にあるストレージの関連付けが容易になります。

CVM クラスタが対称である場合、クラスタ内の各ノードは同じディスクセットにアクセスします。この名前の付け方によって、対称クラスタのノード間で名前の付け方の一貫性が保たれます。

デフォルトでは、OS ベースの名前は永続的ではありません。したがって、オペレーティングシステムから認識されるようにシステム構成がデバイス名を変更すると、OS ベースの名前は再生成されます。OS に基づく名前が再ブート後も変更されないようにするには、名前の付け方の `persistence` 属性を設定します。

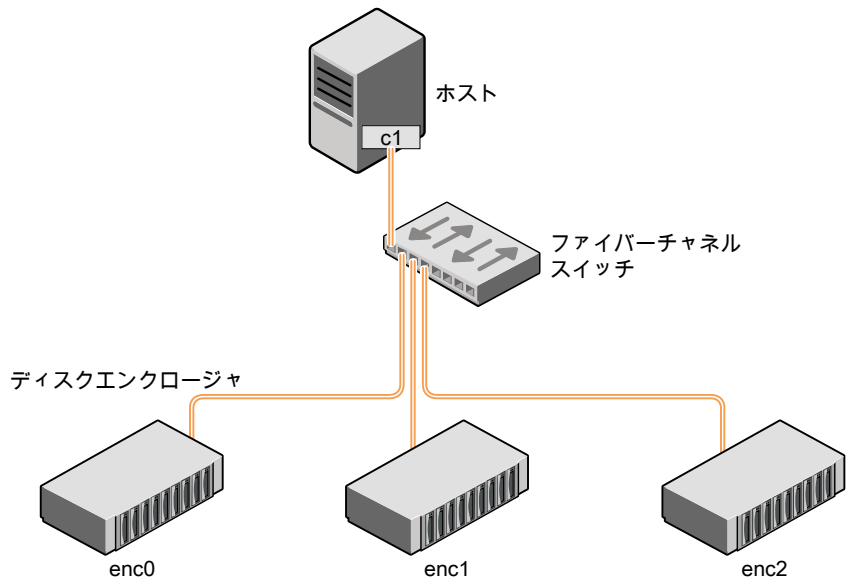
p.102 の「[ディスクデバイスの名前の付け方の変更](#)」を参照してください。

## エンクロージャに基づく名前の付け方について

オペレーティングシステムに基づくデバイスの名前の付け方の代わりに、エンクロージャに基づく名前の付け方を使うことができます。ファイバーチャネルスイッチを使うストレージエリアネットワーク(SAN)では、オペレーティングシステムからのディスクの配置情報で、ディスクの物理的位置が正しく示されない場合があります。エンクロージャに基づく名前の付け方では、DMP はエンクロージャに個々の物理エンティティとしてアクセスできます。個々のエンクロージャに、データの冗長性のあるコピーを設定することで、1 つ以上のエンクロージャの障害に対抗できます。

図 1-3 に、ホストコントローラがファイバーチャネルスイッチを使って複数のエンクロージャに接続されている通常の SAN 環境を示します。

図 1-3 ファイバーチャネルスイッチで接続されているディスクエンクロージャの設定例



このような設定では、エンクロージャに基づく命名を使って、エンクロージャ内の各ディスクを示すことができます。たとえば、エンクロージャ enc0 内のディスクのデバイス名は enc0\_0、enc0\_1 のように設定されています。この規則の主な利点は、大規模な SAN 設定でディスクの物理的位置を迅速に特定できることです。

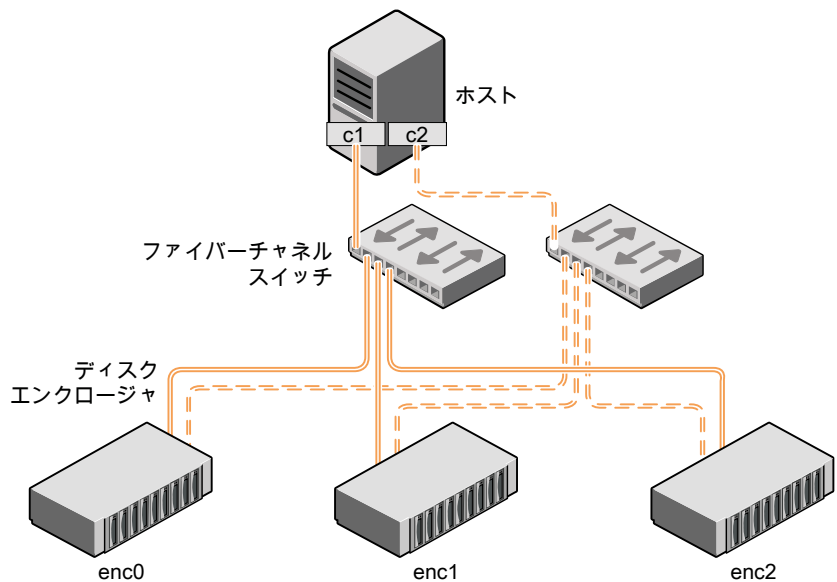
ほとんどのディスクアレイでは、ハードウェアベースのストレージ管理を使って、複数の物理ディスク 1 つの LUN としてオペレーティングシステムに提示できます。このような場合、VxVM でもコンポーネントディスクではなく、1 つの論理ディスクデバイスを認識します。このため、エンクロージャ内のディスクにリファレンスを作成する場合、ディスクは物理ディスクまたは LUN になります。

エンクロージャに基づく命名によるもう 1 つの重要な利点は、VxVM がデータの冗長なコピーを同じエンクロージャに配置するのを回避できることです。各エンクロージャは独立した障害のあるドメインと認識される可能性があるため、そのような配置を回避するのは好ましいことです。たとえば、ミラー化したボリュームがエンクロージャ enc1 のディスク上のみ設定された場合は、スイッチとエンクロージャ間のケーブル障害により、ボリューム全体が利用できなくなることがあります。

必要に応じて、DMP がエンクロージャに割り当てたデフォルト名を自分の設定に意味のある名前に交換することができます。

図 1-4 に、ホスト上の独立したコントローラをエンクロージャへの独立したパスを持つ個々のスイッチに接続して、ストレージへの冗長ループアクセスを実現する高可用性 (HA) 設定を示します。

図 1-4 冗長ループアクセスを実行するために複数のスイッチを使った HA の設定例



このような設定により、ホストコントローラ (c1 と c2) のいずれかに障害が発生したり、ホストといずれかのスイッチをつなぐケーブルに障害が発生しても可用性を維持することができます。この例では、VxVM がアクセスできるすべてのパスで、各ディスクは同じ名前になっています。たとえば、ディスクデバイス enc0\_0 は 1 つのディスクを表しますが、オペレーティングシステムには 2 つの異なるパス、sdf と sdm が認識されています。

p.18 の「DMP でのディスクデバイスの命名」を参照してください。

p.102 の「ディスクデバイスの名前の付け方の変更」を参照してください。

データの冗長性を設定するとき、ドメインに障害が発生することを考慮するために、ミラー化したボリュームを、エンクロージャをまたがってレイアウトする方法を制御できます。

## エンクロージャに基づく名前の付け方の概略

デフォルトでは、DMP はエンクロージャに基づく名前の付け方を 사용합니다。

エンクロージャに基づく名前の付け方を次に示します。

- サポートされているディスクアレイ内にあるファブリックディスクまたは非ファブリックディスクには、`enclosure_name_#` 形式で名前が設定されます。たとえば、サポートされているディスクアレイ `enggdept` 内のディスクには、`enggdept_0`、`enggdept_1`、`enggdept_2` のように名前が設定されます。  
`vxmpadm` コマンドを使ってエンクロージャ名を管理できます。  
`vxmpadm(1M)` のマニュアルページを参照してください。
- `DISKS` カテゴリ内のディスク (JBOD ディスク) には、`Disk_#` 形式で名前が設定されます。
- ディスクパーティションは、名前に追加される `s#` (`#` はパーティション番号) で示されます。たとえば、`Disk_0s5` および `Disk_0s6` は、`sliced` ディスク `Disk_0` のプライベートリージョンとパブリックリージョンに使われる拡張パーティションを表します。`ACME_0s5` は、`simple` ディスク `ACME_0` の拡張パーティションを表します。`CDS` ディスクでは、パーティション `3` がプライベートリージョンとパブリックリージョンの両方に使われます。
- `OTHER_DISKS` カテゴリ内のディスク (DMP によってマルチパス化されていないディスク) には、`hdx[N]` 形式または `sdx[N]` 形式で名前が設定されます。
- カプセル化されたルートディスクは、常に `hdx[N]` 形式または `sdx[N]` 形式を使います。

デフォルトでは、エンクロージャに基づく名前は永続的です。したがって、その名前は再起動後も変わりません。

CVM クラスタが対称的である場合、クラスタ内の各ノードは同じディスクセットにアクセスします。エンクロージャに基づく名前は一貫性のある名前付けシステムを提供するため、各ノードでデバイス名は同じになります。

DMP ディスク (`mydg01` など) の OS ネーティブなデバイス名を表示するには、次のコマンドを使います。

```
# vxdisk path | grep diskname
```

p.88 の「[ディスクカテゴリ](#)」を参照してください。

p.22 の「[エンクロージャに基づく命名規則とアレイボリューム ID \(AVID\) 属性](#)」を参照してください。

## エンクロージャに基づく命名規則とアレイボリューム ID (AVID) 属性

DMP (Dynamic Multi-Pathing) はデフォルトで、アレイボリューム ID (AVID) と呼ばれるアレイ固有の属性を使って、エンクロージャベースの名前を DMP メタデバイスに割り当てます。AVID は、アレイによって提供される LUN の重複のない ID を提供します。アレイに対応する ASL が AVID プロパティを提供します。アレイエンクロージャ内で、DMP は DMP メタノード名内のインデックスとして、AVID を使います。DMP メタノード名は、`enclosureID_AVID` という形式になっています。

エンクロージャに基づく命名 (EBN) 命名規則に AVID が導入されたことにより、ストレージデバイスの識別がずっと容易になりました。アレイボリューム ID (AVID) により、同じストレージに接続された複数のノード間で一貫したデバイスの命名が可能になります。ディスクアクセス名はアレイ自体によって定義された名前がベースになるため、変化することはありません。

---

**メモ:** DMP はサードパーティ製のドライバでの AVID をサポートしません。

---

DMP がデバイスの AVID にアクセスできない場合、LUN シリアル番号と呼ばれる、重複のない別の LUN 識別子を取得します。DMP は LUN シリアル番号 (LSN) に基づいてデバイスをソートしてから、インデックス番号を割り当てます。すべてのホストが同じデバイス集合を参照するため、すべてのホストでソート済みのリストが同じになり、クラスタ間でデバイスのインデックスの一貫性が保たれます。この場合、DMP メタノード名は、`enclosureID_index` の形式です。

DMP は拡張性のあるフレームワークもサポートします。このフレームワークでは、カスタム の名前をキャビネットおよび LUN のシリアル番号と関連付けるデバイス命名ファイルを適用することによって、ホスト上でデバイス名を完全にカスタマイズすることが可能です。

CVM (CLuster Volume Manager) クラスタが対称的である場合、クラスタ内の各ノードは同じディスクセットにアクセスします。エンクロージャに基づく名前は一貫性のある名前付けシステムを提供するため、各ノードでデバイス名は同じになります。

`vxdisk list` などの Dynamic Multi-Pathing (DMP) ユーティリティは、AVID プロパティを含む DMP メタノード名を表示します。アレイ管理インターフェース (GUI または CLI) に表示される LUN に DMP メタノード名を関連付けるには、AVID を使います。

たとえば、エンクロージャが `emc_clariion0` で、ASL によって提供されたアレイボリューム ID が `91` である EMC CX アレイでは、DMP メタノード名は `emc_clariion0_91` です。次の出力例は DMP メタノード名を示しています。

```
$ vxdisk list
emc_clariion0_91 auto:cdsdisk emc_clariion0_91 dg1 online shared
emc_clariion0_92 auto:cdsdisk emc_clariion0_92 dg1 online shared
emc_clariion0_93 auto:cdsdisk emc_clariion0_93 dg1 online shared
emc_clariion0_282 auto:cdsdisk emc_clariion0_282 dg1 online shared
```

```
emc_clariion0_283 auto:cdsdisk emc_clariion0_283 dg1 online shared
emc_clariion0_284 auto:cdsdisk emc_clariion0_284 dg1 online shared
```

```
# vxddladm get namingscheme
```

```
NAMING_SCHEME      PERSISTENCE      LOWERCASE      USE_AVID
=====
Enclosure Based    Yes                Yes             Yes
```

# ネイティブデバイス管理のための DMP の設定

この章では以下の項目について説明しています。

- [ネイティブデバイス管理のための DMP の設定について](#)
- [ネイティブマルチパス設定の表示](#)
- [DMP への LVM ボリュームグループの移行](#)
- [EMC PowerPath から DMP への移行](#)
- [Hitachi Data Link Manager \(HDLM\) から DMP への移行](#)
- [Linux Device Mapper Multipath から DMP への移行](#)
- [Oracle Automatic Storage Management \(ASM\) での Dynamic Multi-Pathing \(DMP\) デバイスの使用](#)
- [既存の LVM ボリュームグループへの DMP デバイスの追加または LVM ボリュームグループの新規作成](#)
- [ネイティブデバイスの DMP サポートの削除](#)

## ネイティブデバイス管理のための DMP の設定について

サードパーティ製ドライバの代わりに DMP を使うことで、高度なストレージ管理が可能になります。この項では、ネイティブ LVM デバイスと、それらのデバイス上で動作する論理ボリュームを管理するために DMP を設定する方法について説明します。

DMP をインストールした後に、LVM で使うために DMP を設定します。LVM で使うために DMP を設定するには、`dmp_native_support` チューニングパラメータを有効にします。このチューニングパラメータを有効にすると、VxVM ラベルが付いておらず、サードパーティ製マルチパス (TPD) ソフトウェアの制御下でないすべてのデバイスで、DMP は



```
-----
dmp_native_support          on          off
```

## ネーティブマルチパス設定の表示

ネーティブデバイスで DMP が有効な場合は、`dmp_native_support` チューニングパラメータが ON に設定されます。チューニングパラメータが ON の場合、以下を除くすべての DMP ディスクがネーティブボリュームで利用可能です。

- VxVM ラベルの付いたデバイス  
VxVM で使用するためにディスクを初期化すると、そのディスクのネーティブマルチパス機能が自動的に無効になります。  
VxVM が使うディスクを削除する場合は、ネーティブマルチパスのディスクを使うことができます。
- サードパーティ製ドライバによってマルチパス化されるデバイス  
ディスクがすでにサードパーティ製ドライバ (TPD) によってマルチパス化されている場合、TPD サポートを削除しないかぎり、DMP はデバイスを管理しません。

DMP が有効かどうかを表示するには

- 1 属性 `dmp_native_support` を表示します。

```
# vxddmpadm gettune dmp_native_support

Tunable                Current Value  Default Value
-----
dmp_native_support     on            off
```

- 2 `dmp_native_support` チューニングパラメータが ON のときは、`vxdisk list` コマンドを使って利用可能なディスクを表示します。LVM で利用可能なディスクはタイプ (TYPE) が `auto:none` と表示されます。LVM によってすでに使われているディスクはタイプ (TYPE) が `auto:LVM` と表示されます。

## DMP への LVM ボリュームグループの移行

サードパーティ製ドライバの代わりに DMP を使うことで、高度なストレージ管理が可能になります。ここでは、LVM ボリュームグループと、プール上で稼働するファイルシステムを管理するために DMP を設定する方法について説明します。

DMP を設定するには、デバイスを既存のサードパーティ製デバイスドライバから DMP に移行します。

表 2-1 に、サポートされているネーティブソリューションと移行パスを示します。

表 2-1 サポートされている移行パス

オペレーティングシステム	ネイティブソリューション	移行手順
Linux	EMC PowerPath	p.27 の「 <a href="#">EMC PowerPath から DMP への移行</a> 」を参照してください。
Linux	Hitachi Data Link Manager (HDLM)	p.28 の「 <a href="#">Hitachi Data Link Manager (HDLM) から DMP への移行</a> 」を参照してください。
Linux	Linux Device Mapper Multipath	p.29 の「 <a href="#">Linux Device Mapper Multipath から DMP への移行</a> 」を参照してください。

## EMC PowerPath から DMP への移行

ここでは、デバイスを EMC PowerPath 制御から削除し、デバイスで DMP を有効にする手順について説明します。

移行する PowerPath デバイスに属するすべてのパスが移行時に健全な状態であることを確認します。

次の手順のために、アプリケーションのダウンタイムを計画します。

**EMC PowerPath 制御からデバイスを削除して DMP を有効にするには**

- 1 PowerPath メタデバイスを使うアプリケーションを停止します。

VCS 環境では、アプリケーションの VCS サービスグループを停止することによってアプリケーションを停止します。

- 2 PowerPath デバイス上のボリュームグループを使うすべてのファイルシステムをマウント解除します。
- 3 PowerPath デバイスを使う LVM ボリュームグループを停止します。

```
# lvchange -a n lvmname
```

- 4 PowerPath デバイスのディスクアクセス名を VxVM から削除します。

```
# vxdisk rm emcpowerXXXX
```

`emcpowerXXXX` は EMC PowerPath デバイスの名前です。

- 5 デバイスを PowerPath 制御から削除します。

```
# powermt unmanage dev=pp_device_name  
or  
# powermt unmanage class=array_class
```

- 6 PowerPath デバイスが PowerPath 制御から削除されたことを確認します。

```
# powermt display dev=all
```

- 7 デバイススキャンを実行して、デバイスを DMP 制御下に置きます。

```
# vxdisk scandisks
```

- 8 LVM ボリュームグループの DMP サポートを有効にします。

```
# vxddm padm settune dmp_native_support=on
```

このコマンドによって、LVM ルートの DMP サポートも有効になります。

ルートをサポートするには再ブートが必要です。

- 9 ファイルシステムをマウントします。

- 10 アプリケーションを再起動します。

## Hitachi Data Link Manager (HDLM) から DMP への移行

ここでは、デバイスを HDLM 制御から削除し、デバイスで DMP を有効にする手順について説明します。

---

**メモ:** DMP は HDLM と共存できません。HDLM をシステムから削除する必要があります。

---

以下の手順のために、アプリケーションとシステムのダウンタイムを計画します。

**Hitachi Data Link Manager (HDLM) からデバイスを削除して DMP を有効にするには**

- 1 HDLM メタデバイスを使っているアプリケーションを停止します。
- 2 HDLM デバイス上のボリュームグループを使うすべてのファイルシステムをマウント解除します。

VCS 環境では、アプリケーションの VCS サービスグループを停止することによってアプリケーションを停止します。

- 3 HDLM デバイスを使う LVM ボリュームグループを停止します。

```
# lvchange -a n lvpath
```

- 4 HDLM パッケージをアンインストールします。
- 5 LVM ボリュームグループの DMP サポートを有効にします。

```
# vxddmoadm settune dmp_native_support=on
```

このコマンドによって、LVM ルートの DMP サポートも有効になります。

- 6 システムの再起動。
- 7 再ブート後は、DMP がデバイスを制御します。HDLM デバイス上に LVM ボリュームグループが存在する場合は、DMP デバイス上に移行されます。
- 8 ファイルシステムをマウントします。
- 9 アプリケーションを再起動します。

## Linux Device Mapper Multipath から DMP への移行

ここでは、デバイスを Linux Device Mapper Multipath 制御から削除し、デバイスで DMP を有効にする手順について説明します。

以下の手順のために、システムの停止時間を計画してください。

以下の理由により、移行の際にホストでシステム停止時間が必要です。

- アプリケーションを停止する必要がある
- VCS を使っている場合は VCS サービスを停止する必要がある
- 手順の実施中にホストの再ブートが 1 回以上必要

**Device Mapper Multipath 制御からデバイスを削除して DMP を有効にするには**

- 1 Device Mapper Multipath デバイスを使うアプリケーションを停止します。
- 2 Device Mapper Multipath デバイスを使うすべてのファイルシステムをマウント解除します。
- 3 Device Mapper Multipath デバイス上のすべてのボリュームを無効にします。

```
# lvchange -a n lvname
```

- 4 /etc/multipath.conf ファイルを更新し、すべての Device Mapper デバイスをブラックリストに登録します。この手順により、すべてのデバイスでマルチパスが無効になります。

```
# Blacklist all devices by default.
blacklist {
    devnode "*"
}
```

- 5 multipathd を再起動して、すべての /dev/dm-\* および /dev/mpath/\* デバイスエントリを削除します。

```
# service multipathd restart
```

- 6 multipathd デーモンを停止します。

```
# service multipathd stop
```

- 7 再ブート後に multipathd が起動しないようにします。

```
# chkconfig multipathd off
```

- 8 LVM ボリュームグループの DMP サポートを有効にします。

```
# vxdmpadm settune dmp_native_support=on
```

このコマンドによって、LVM ルートの DMP サポートも有効になります。

この手順は、LV の数によっては少し時間がかかることがあります。

- 9 ファイルシステムをマウントします。

- 10 アプリケーションを再起動します。

## Oracle Automatic Storage Management (ASM) での Dynamic Multi-Pathing (DMP) デバイスの使用

DMP は、Oracle Automatic Storage Management (ASM) での DMP デバイスの使用をサポートします。DMP は次の操作をサポートします。

- p.31 の「[Oracle Automatic Storage Management \(ASM\) で使用可能にするための Dynamic Multi-Pathing \(DMP\) デバイスの有効化](#)」を参照してください。
- p.32 の「[Oracle Automatic Storage Management \(ASM\) ディスクのリストからの Dynamic Multi-Pathing \(DMP\) デバイスの削除](#)」を参照してください。

- p.33 の「オペレーティングシステムデバイス上の Oracle Automatic Storage Management (ASM) ディスクグループの Dynamic Multi-Pathing (DMP) デバイスへの移行」を参照してください。

## Oracle Automatic Storage Management (ASM) で使用可能にするための Dynamic Multi-Pathing (DMP) デバイスの有効化

DMP デバイスを利用可能なディスクとして Oracle Automatic Storage Management (ASM) から認識するには、ASM で DMP サポートを有効にします。ASM の DMP サポートは `char` デバイス (`/dev/vx/rdmp/*`) で利用できます。

### DMP デバイスを ASM から認識するには

- 1 ASM で、`ASM_DISKSTRING` が正しい値に設定されていることを確認します。

```
/dev/vx/rdmp/*
```

次に例を示します。

```
SQL> show parameter ASM_DISKSTRING;
NAME                                TYPE                                VALUE
-----
asm_diskstring                       string                              /dev/vx/rdmp/*
```

- 2 `root` ユーザーとして、ASM で使う DMP デバイスを有効にします。

```
# vxdmpraw enable usernamegroupnamemode [devicename ...]
```

`username` は ASM インスタンスを実行している ASM ユーザー、`groupname` は指定したユーザー ID の UNIX/Linux グループ名、`mode` はデバイスに設定する権限をそれぞれ表します。1 つ以上の `devicenames` を指定すると、それらのデバイスで ASM の DMP サポートが有効になります。`devicename` を指定しない場合、システム内のデバイスのうち、ASM 署名があるすべてのデバイスで DMP サポートが有効になります。

次に例を示します。

```
# vxdmpraw enable oracle dba 765 eva4k6k0_1
```

ASM サポートは有効になっています。DMP デバイスのアクセス権限は `mode` で指定された権限に設定されます。変更は再ブート後も変化しません。

- 3 ASM で、これらの新しいデバイスが ASM から参照できることを確認します。

```
SQL> select name,path,header_status from v$asm_disk;
```

NAME	PATH	HEADER_STATUS
.....	.....	....
	/dev/vx/rdmp/eva4k6k0_1	CANDIDATE
.....	.....	....

- 4 ASM から、Oracle ハートビート待ち時間をデフォルト値の 15 秒から増やします。DMP フェールオーバー中に Oracle アプリケーションがディスクをオフラインとマーク付けしないように、\_asm\_hbeatowait のデフォルト値を増やします。

- たとえば、値を 360 秒に設定するには、次のように実行します。

```
SQL> alter system set "_asm_hbeatowait"=360 scope=spfile
sid='*';
```

- ASM インスタンスを再起動して、新しいパラメータを有効にします。

## Oracle Automatic Storage Management (ASM) ディスクのリストからの Dynamic Multi-Pathing (DMP) デバイスの削除

ASM ディスクのリストから DMP デバイスを削除するには、ASM での DMP サポートをデバイスから無効にします。ASM ディスクグループに属するデバイスからは、ASM の DMP サポートを削除できません。

**ASM ディスクのリストから DMP デバイスを削除するには**

- 1 デバイスが ASM ディスクグループに属する場合、ASM ディスクグループからデバイスを削除します。
- 2 root ユーザーとして、ASM で使う DMP デバイスを無効にします。

```
# vxdmpraw disable diskname
```

次に例を示します。

```
# vxdmpraw disable eva4k6k0_1
```

## オペレーティングシステムデバイス上の Oracle Automatic Storage Management (ASM) ディスクグループの Dynamic Multi-Pathing (DMP) デバイスへの移行

既存の ASM ディスクグループがオペレーティングシステムのネイティブデバイスをディスクとして使うと、それらのデバイスを Dynamic Multi-Pathing 制御に移行できます。OS デバイスが他のマルチパスドライバによって制御されている場合、この操作には、デバイスを DMP 制御に移行するためのシステム停止時間が必要です。

以下の手順のために、システムの停止時間を計画してください。

この手順の実行後、ASM ディスクグループは移行された DMP デバイスをそのディスクとして使います。

「ASM で」は、ASM インスタンスを実行するユーザーとして手順を実行することを示します。

「root ユーザーとして」は、root ユーザーとして手順を実行することを示します。

**ASM ディスクグループをオペレーティングシステムデバイスから DMP デバイスに移行するには**

- 1 アプリケーションを終了し、データベースをシャットダウンします。
- 2 ASM で、移行する ASM ディスクグループと、その制御下にあるディスクを識別します。
- 3 ASM で、ASM ディスクグループをマウント解除します。
- 4 他のマルチパスドライバによってデバイスが制御されている場合、デバイスを DMP 制御に移行します。root ユーザーとして次の手順を実行します。

PowerPath または Device Mapper Multipath から移行します。

p.24 の「[ネイティブデバイス管理のための DMP の設定について](#)」を参照してください。

- 5 root ユーザーとして、raw コマンドを使って、特定の OS デバイスに対して作成された RAW デバイスを削除します。
- 6 root ユーザーとして、手順 2 で識別した ASM ディスクグループの DMP サポートを有効にします。

```
# vxdmpraw enable usernamegroupnamemode [devicename ...]
```

*username* は ASM インスタンスを実行している ASM ユーザー、*groupname* は指定したユーザー ID の UNIX/Linux グループ名、*mode* はデバイスに設定する権限をそれぞれ表します。1 つ以上の *devicenames* を指定すると、それらのデバイスで ASM の DMP サポートが有効になります。*devicename* を指定しない場合、システム内のデバイスのうち、ASM 署名があるすべてのデバイスで DMP サポートが有効になります。

- 7 必要に応じて、ASM から、ASM\_DISKSTRING を設定します。適切な設定は /dev/vx/rdmp/\* です。
- 8 ASM で、デバイスが ASM から利用可能であることを確認します。
- 9 ASM で、ASM ディスクグループをマウントします。ディスクグループは DMP デバイス上にマウントされます。

例: ASM ディスクグループをオペレーティングシステムデバイスから DMP デバイスに移行するには

- 1 ASM で、移行する ASM ディスクグループと、その制御下にあるディスクを識別します。

```
SQL> select name, state from v$asm_diskgroup;
NAME                                STATE
-----
ASM_DG1                             MOUNTED

SQL> select path , header_status from v$asm_disk where
header_status='MEMBER';
```

NAME	PATH	HEADER_STATUS
ASM_DG1_0000	/dev/vx/rdmp/sda	MEMBER
ASM_DG1_0001	/dev/vx/rdmp/sdc	MEMBER
ASM_DG1_0002	/dev/vx/rdmp/sdd	MEMBER

- 2 ASM で、ASM ディスクグループをマウント解除します。

```
SQL> alter diskgroup ASM_DG1 dismount;
Diskgroup altered.

SQL> select name , state from v$asm_diskgroup;
NAME                                STATE
-----
ASM_DG1                             DISMOUNTED
```

- 3 他のマルチパスドライバによってデバイスが制御されている場合、デバイスを DMP 制御に移行します。root ユーザーとして次の手順を実行します。  
 p.24 の「[ネーティブデバイス管理のための DMP の設定について](#)」を参照してください。
- 4 root ユーザーとして、次のいずれかの方法で、手順 2 で識別した ASM ディスクグループの DMP サポートを有効にします。

- 選択した ASM ディスクグループを移行するには、vxddmpadm コマンドを使って、OS デバイスに対応する DMP ノードを確認します。

```
# vxddmpadm getddmpnode nodename=sdd
NAME          STATE    ENCLR-TYPE  PATHS  ENBL  DSBL  ENCLR-NAME
=====
EVA4k6k0_0  ENABLED EVA4K6K      4      4     0     EVA4k6k0
```

次のコマンドでデバイス名を使います。

```
# vxddmpraw enable oracle dba 660 eva4k6k0_0 ¥
      eva4k6k0_9 emc_clariion0_243
```

- **devicename** を指定しない場合、ディスクグループ内のデバイスのうち、ASM 署名があるすべてのデバイスで DMP サポートが有効になります。次に例を示します。

```
# vxddmpraw enable oracle dba 660
```

## 5 ASM から、ASM\_DISKSTRING を設定します。

```
SQL> alter system set ASM_DISKSTRING='/dev/vx/rdmp/*';
System altered.
```

```
SQL> show parameter ASM_DISKSTRING;
```

NAME	TYPE	VALUE
asm_diskstring	string	/dev/vx/rdmp/*

**6 ASM で、デバイスが ASM から利用可能であることを確認します。**

```
SQL> select path , header_status from v$asm_disk where
header_status='MEMBER';
```

NAME	PATH	HEADER_STATUS
	/dev/vx/rdmp/emc_clariion0_243	MEMBER
	/dev/vx/rdmp/eva4k6k4k0_0	MEMBER
	/dev/vx/rdmp/eva4k6k0_1	MEMBER

**7 ASM で、ASM ディスクグループをマウントします。ディスクグループは DMP デバイス上にマウントされます。**

```
SQL> alter diskgroup ASM_DG1 mount;
Diskgroup altered.
```

```
SQL> select name, state from v$asm_diskgroup;
```

NAME	STATE
ASM_DG1	MOUNTED

```
SQL> select path , header_status from v$asm_disk where
header_status='MEMBER';
```

NAME	PATH	HEADER_STATUS
ASM_DG1_0002	/dev/vx/rdmp/emc_clariion0_243	MEMBER
ASM_DG1_0000	/dev/vx/rdmp/eva4k6k0_1	MEMBER
ASM_DG1_0001	/dev/vx/rdmp/eva4k6k0_9	MEMBER

## 既存の LVM ボリュームグループへの DMP デバイスの追加または LVM ボリュームグループの新規作成

dmp\_native\_support が ON の場合、利用可能な DMP デバイス上に新しい LVM ボリュームグループを作成できます。利用可能な DMP デバイスを既存の LVM ボリュームグループに追加することもできます。LVM ボリュームグループを DMP デバイス上に配置した後に、任意の LVM コマンドを使ってボリュームグループを管理できます。

**DMP デバイス上に LVM ボリュームグループを新規作成または DMP デバイスを既存の LVM ボリュームグループに追加するには**

- 1 LVM のために利用可能なディスクを選択します。

vxdisk list コマンドを使って、次の種類のディスクを識別します。

- VxVM が使っていないディスク

vxdisk list の出力に、タイプが **auto:none** で状態が **online invalid** のディスクが表示されます。

利用可能なディスクの表示例を示します。

```
# vxdisk list

DEVICE                TYPE          DISK  GROUP  STATUS
. . .
tagmastore-usp0_0035 auto:none -    -      online invalid
tagmastore-usp0_0036 auto:none -    -      online invalid
```

## 2 DMP デバイス上に LVM ボリュームグループを新規作成します。

DMP デバイスの完全なパス名を使います。

```
# pvcreate /dev/vx/dmp/tagmastore-usp0_0035
Physical volume "/dev/vx/dmp/tagmastore-usp0_0035" successfully
created
#
# vgcreate /dev/newvg /dev/vx/dmp/tagmastore-usp0_0035
Volume group "newvg" successfully created
# vdisplay -v newvg |grep Name
Using volume group(s) on command line
Finding volume group "newvg"
VG Name                newvg
PV Name                 /dev/vx/dmp/tagmastore-usp0_0035s3
```

- 3 既存の LVM ボリュームグループに DMP デバイスを追加します。

DMP デバイスの完全なパス名を使います。

```
# pvcreate /dev/vx/dmp/tagmastore-usp0_0036
Physical volume "/dev/vx/dmp/tagmastore-usp0_0036"
successfully created

# vgextend newvg /dev/vx/dmp/tagmastore-usp0_0036
Volume group "newvg" successfully extended

# vdisplay -v newvg |grep Name
Using volume group(s) on command line
Finding volume group "newvg"
VG Name                newvg
PV Name                 /dev/vx/dmp/tagmastore-usp0_0035s3
PV Name                 /dev/vx/dmp/tagmastore-usp0_0036s3
```

- 4 次のコマンドを実行して、デバイスの DMP 検出をトリガします。

```
# vxdisk scandisks
```

- 5 検出が完了した後に、ディスクが LVM によって使用中であることが示されます。

```
# vxdisk list
. . .
tagmastore-usp0_0035 auto:LVM - - LVM
tagmastore-usp0_0036 auto:LVM - - LVM
```

- 6 LVM ボリュームエントリの場合は、`/etc/fstab` のマウントオプションに「`_netdev`」を追加します。このオプションを指定すると、DMP デバイスが検出された後にこれらのボリュームが有効になります。

## ネーティブデバイスの DMP サポートの削除

`dmp_native_support` チューニングパラメータは、再ブートや製品のアップグレードの前で変化しません。VxVM 用にデバイスを初期化する場合や、そのデバイスで TPD マルチパスを設定する場合は、個々のデバイスのネーティブサポートを無効にできます。

すべての DMP デバイスからネーティブデバイスのサポートを削除するには、

`dmp_native_support` チューニングパラメータを無効にします。

`dmp_native_support` チューニングパラメータを無効にするには、次のコマンドを使います。

```
# vxdmpadm settune dmp_native_support=off
```

**dmp\_native\_support** チューニングパラメータの値を表示するには、次のコマンドを使います。

```
# vxdmpadm gettune dmp_native_support
```

Tunable	Current Value	Default Value
dmp_native_support	off	off

# DMP の管理

この章では以下の項目について説明しています。

- [コントローラとストレージプロセッサに対する I/O の有効化と無効化について](#)
- [DMP データベース情報の表示について](#)
- [ディスクへのパスの表示](#)
- [DMP ノードのカスタム名の設定](#)
- [vxddmpadm ユーティリティを使った DMP の管理](#)

## コントローラとストレージプロセッサに対する I/O の有効化と無効化について

DMP を使うと、ホストバスアダプタ (HBA) コントローラ、またはストレージプロセッサのアレイポートを介して I/O を無効にして、管理上の操作を実行できるようになります。この機能は、ホスト上の HBA コントローラ、または DMP に対応しているディスクアレイに接続したアレイポートでメンテナンスを実施するときに使うことができます。HBA コントローラまたはアレイポートの I/O 操作は、管理タスクが完了した後で有効に戻すことができます。この一連の操作は、vxddmpadm コマンドを使って実行することができます。

アクティブ/アクティブタイプのディスクアレイの場合は、HBA コントローラまたはアレイポートを介した I/O を無効にすると、I/O は残りのパス上で継続されます。アクティブ/パッシブタイプのディスクアレイの場合は、HBA コントローラまたはアレイポートを介した I/O を無効にするとすべてのプライマリパスが無効になるため、DMP はセカンダリパスにフェールオーバーし、I/O はそれらのパス上で継続されます。

管理操作が終了したら、vxddmpadm を使って、HBA コントローラまたはアレイポートを介したパスを再び有効にしてください。

p.74 の「[パス、コントローラ、アレイポート、DMP ノードに対する I/O の無効化](#)」を参照してください。

p.75 の「パス、コントローラ、アレイポート、DMP ノードに対する I/O の有効化」を参照してください。

一定の再設定操作は動的にオンラインでも実行できます。

## DMP データベース情報の表示について

`vxddmpadm` コマンドを使って、DMP データベース情報の一覧表示と他の管理タスクを実行することができます。このコマンドを実行すると、ディスクに接続されているすべてのコントローラおよび DMP データベースに保存されている他の関連情報を一覧表示することができます。この情報は、システムのハードウェアの配置および有効化や無効にする必要のあるコントローラの判定に役立てることができます。

`vxddmpadm` コマンドでは、ディスクアレイシリアル番号や、どの DMP デバイス(ディスク)がディスクアレイに接続されているのか、どのパスが特定のコントローラ、エンクロージャ、アレイポートに接続されているのかといった、有用な情報が提供されます。

p.44 の「`vxddmpadm` ユーティリティを使った DMP の管理」を参照してください。

## ディスクへのパスの表示

`vxddisk` コマンドは、特定のメタデバイスに関するマルチパス情報を表示するのに使われます。メタデバイスは、システムの HBA コントローラを通じた物理パスを複数持つ物理ディスクを表すデバイスです。DMP (Dynamic Multi-Pathing) では、システム内のすべての物理ディスクを、1 つ以上の物理パスを持つメタデバイスとして表します。

システム上のマルチパス情報を表示するには

- ◆ `vxddisk path` コマンドを実行すると、次に示すように、システム上のデバイスパス、ディスクアクセス名、ディスクメディア名、ディスクグループの関係が表示されます。

```
# vxddisk path
```

SUBPATH	DANAME	DMNAME	GROUP	STATE
sda	sda	mydg01	mydg	ENABLED
sdi	sdi	mydg01	mydg	ENABLED
sdb	sdb	mydg02	mydg	ENABLED
sdj	sdj	mydg02	mydg	ENABLED
.	.	.	.	.

この出力例では、2 つのディスク(mydg01 と mydg02) にそれぞれ 2 つのパスが存在し、各ディスクが ENABLED 状態であることが示されています。

特定のメタデバイスのマルチパス情報を表示するには

1 次のコマンドを実行します。

```
# vxdisk list devicename
```

たとえば、デバイス `sdl` のマルチパス情報を表示するには、次のコマンドを使います。

```
# vxdisk list sdl
```

`vxdisk list` コマンドからの出力では、次の例に示すようにマルチパスの情報が表示されます。

```
Device:      sdl
devicetag:   sdl
type:        sliced
hostid:      sys1
.
.
.
Multipathing information:
numpaths:    2
sdl  state=enabled      type=primary
sdp  state=disabled     type=secondary
```

`numpaths` 行では、デバイスに対して 2 つのパスがあることが示されています。出力の [Multipathing information] セクションの次の 2 行では、1 つのパスがアクティブ (`state=enabled`) で、もう 1 つのパスでエラーが発生している (`state=disabled`) ことを示します。

`type` フィールドは、EMC CLARiON、Hitachi HDS 9200 および 9500、Sun StorEdge 6xxx および Sun StorEdge T3 アレイといったアクティブ/パッシブタイプのディスクアレイ上のディスクの場合に表示されます。このフィールドでは、ディスクへのパスがプライマリパスであるか、セカンダリパスであるかが表示されます。

`type` フィールドは、EMC Symmetrix、Hitachi HDS 99xx、Sun StorEdge 99xx シリーズおよび IBM ESS シリーズといったアクティブ/アクティブタイプのディスクアレイ上のディスクの場合には表示されません。このタイプのディスクアレイに、プライマリパスおよびセカンダリパスといった概念は存在しません。

2 また、次のコマンドを使ってマルチパス情報を表示できます。

```
# vxddmpadm getsubpaths dmpnodename=devicename
```

たとえば、emc\_clariion0\_893 のマルチパス情報を表示するには、次のコマンドを使います。

```
# vxddmpadm getsubpaths dmpnodename=emc_clariion0_893
```

vxddmpadm getsubpaths コマンドの通常の出力行は、次のとおりです。

NAME	STATE [A]	PATH-TYPE [M]	CTLR-NAME	ENCLR-TYPE	ENCLR-NAME	ATTRS
sdbc	ENABLED (A)	PRIMARY	c3	EMC_CLARiion	emc_clariion0	-
sdbm	ENABLED	SECONDARY	c3	EMC_CLARiion	emc_clariion0	-
sdbw	ENABLED (A)	PRIMARY	c3	EMC_CLARiion	emc_clariion0	-
sdck	ENABLED (A)	PRIMARY	c2	EMC_CLARiion	emc_clariion0	-
sdcu	ENABLED	SECONDARY	c2	EMC_CLARiion	emc_clariion0	-
sdde	ENABLED (A)	PRIMARY	c2	EMC_CLARiion	emc_clariion0	-

## DMP ノードのカスタム名の設定

DMP (Dynamic Multi-Pathing) ノード名はディスクへの複数のパスを表すメタデバイス名です。DDL (Device Discovery Layer) は、Dynamic Multi-Pathing (DMP) の名前の付け方に従ってデバイス名から DMP ノード名を生成します。

p.18 の「[DMP でのディスクデバイスの命名](#)」を参照してください。

DMP ノードのカスタム名を指定できます。ユーザーが指定した名前は、名前の永続性が無効になっても永続的に保持されます。

すでにデバイスで使用中のカスタム名は、割り当てることができません。ただし、DDL が生成する名前と同じ名前の付け方に従って名前を割り当てると、デバイスが追加されるときに名前の衝突が起きる可能性があります。DMP デバイスのユーザー定義の名前が、DDL によって生成された別の DMP デバイスの名前と同じ場合、vxddisk list コマンドの出力行では、デバイスの 1 つがエラーとして表示されます。

**DMP のノードのカスタム名を指定するには**

◆ 次のコマンドを実行します。

```
# vxddmpadm setattr dmpnode dmpnodename name=name
```

名前は入力ファイルから割り当てることができます。これにより、意味のある名前を使ってシステムの DMP ノードをカスタマイズできます。

エンクロージャのカスタム名を指定するには

- ◆ 次のコマンドを実行します。

```
# vxddmpadm setattr enclosure enc_name name=custom_name
```

ファイルから DMP ノードを割り当てるには

- 1 設定のデバイスの名前を設定されるファイルを取得するには、次のコマンドを使います。

```
# vxddladm -l assign names > filename
```

サンプルファイルは必要な形式を示し、カスタム名を指定するためのテンプレートとして機能します。

スクリプト `vxgetdmpnames` を使って、設定内のデバイスから設定されたサンプルファイルを取得します。

- 2 必要に応じてファイルを修正します。ファイルの正しい形式を必ず保持してください。
- 3 名前を割り当てるには、次のコマンドにファイルの名前とパスを指定してください。

```
# vxddladm assign names file=pathname
```

カスタム名を消去するには

- ◆ 名前を消去し、デフォルトのオペレーティングシステムに基づく名前付けまたはエンクロージャに基づく名前付けを使うには、次のコマンドを使います。

```
# vxddladm -c assign names
```

## vxddmpadm ユーティリティを使った DMP の管理

vxddmpadm ユーティリティは、DMP (Dynamic Multi-Pathing) に対するコマンドライン管理インターフェースです。

vxddmpadm ユーティリティを使って、次のタスクを実行できます。

- 特定のパスに対する DMP デバイス名の取得  
p.46 の「[DMP ノードに関する情報の取得](#)」を参照してください。
- DMP ノードについての統合された情報の表示  
p.47 の「[DMP ノードについての統合された情報の表示](#)」を参照してください。
- LUN グループのメンバーの表示  
p.48 の「[LUN グループのメンバーの表示](#)」を参照してください。

- DMP デバイスノード、HBA コントローラ、エンクロージャ、アレイポート下のすべてのパスの一覧表示  
 p.49 の「[DMP ノード、コントローラ、エンクロージャ、アレイポートによって制御されるパスの表示](#)」を参照してください。
- ホスト上の HBA コントローラに関する情報の表示  
 p.51 の「[コントローラに関する情報の表示](#)」を参照してください。
- エンクロージャに関する情報の表示  
 p.53 の「[エンクロージャに関する情報の表示](#)」を参照してください。
- エンクロージャのストレージプロセッサに接続したアレイポートに関する情報の表示  
 p.53 の「[アレイポートに関する情報の表示](#)」を参照してください。
- ALUA アレイの非対称アクセス状態の表示。  
 p.54 の「[ALUA アレイのためのユーザーフレンドリな CLI 出力](#)」を参照してください。
- 他社のマルチパス化ドライバにより制御されるデバイスに関する情報の表示  
 p.55 の「[サードパーティ製のドライバにより制御されるデバイスに関する情報の表示](#)」を参照してください。
- 拡張されるデバイスの属性の表示  
 p.56 の「[拡張デバイス属性の表示](#)」を参照してください。
- p.58 の「[VxVM の制御下におけるデバイスの無効化と有効化](#)」を参照してください。  
 DMP の制御下におけるデバイスの無効化と有効化
- DMP ノード、エンクロージャ、パス、コントローラの I/O 統計情報の収集  
 p.59 の「[I/O 統計情報の収集と表示](#)」を参照してください。
- エンクロージャへのパスに関する属性の設定  
 p.65 の「[エンクロージャへのパスに関する属性の設定](#)」を参照してください。
- デバイスまたはエンクロージャの冗長レベルの表示。  
 p.66 の「[デバイスまたはエンクロージャの冗長レベルの表示](#)」を参照してください。
- アクティブパスの最小数の指定。  
 p.67 の「[アクティブパスの最小数の指定](#)」を参照してください。
- エンクロージャへのパスに対して使われる I/O ポリシーの表示または設定  
 p.68 の「[I/O ポリシーの指定](#)」を参照してください。
- システム上のパス、HBA コントローラ、アレイポートに対する I/O の有効化または無効化  
 p.74 の「[パス、コントローラ、アレイポート、DMP ノードに対する I/O の無効化](#)」を参照してください。
- エンクロージャ名の変更  
 p.76 の「[エンクロージャ名の変更](#)」を参照してください。

- I/O 要求エラーに対する DMP の応答方法の設定  
 p.76 の「[I/O エラーに対する応答の設定](#)」を参照してください。
- I/O 調整機構の設定  
 p.78 の「[I/O 調整機構の設定](#)」を参照してください。
- DMP パスリストアスレッドの動作の制御  
 p.81 の「[DMP パスリストアポリシーの設定](#)」を参照してください。
- アレイポリシーモジュール (Array Policy Module) の設定。  
 p.83 の「[アレイポリシーモジュール \(Array Policy Module\) の設定](#)」を参照してください。
- DMP で使われる各種のチューニングパラメータの値の取得または設定  
 p.134 の「[DMP チューニングパラメータ](#)」を参照してください。

vxddmpadm(1M) のマニュアルページを参照してください。

## DMP ノードに関する情報の取得

次のコマンドを実行すると、指定した物理パスを制御する DMP (Dynamic Multi-Pathing) ノードが表示されます。

```
# vxddmpadm getdmpnode nodename=pathname
```

nodename 属性には物理パスを指定できます。この物理パスはデバイスのディレクトリに一覧表示された有効なパスである必要があります。

デバイスのディレクトリは /dev ディレクトリです。

このコマンドの出力は、次の出力例のようになります。

```
# vxddmpadm getdmpnode nodename=sdbc
```

```
NAME                STATE    ENCLR-TYPE    PATHS ENBL DSBL ENCLR-NAME
=====
emc_clariion0_89    ENABLED  EMC_CLARiION  6      6    0    emc_clariion0
```

LUN のシリアル番号とアレイのボリューム ID を表示するには、-v オプションを使います。

```
# vxddmpadm -v getdmpnode nodename=sdbc
```

```
NAME                STATE    ENCLR-TYPE    PATHS ENBL DSBL ENCLR-NAME SERIAL-NO ARRAY_VOL_ID
=====
emc_clariion0_89    ENABLED  EMC_CLARiION  6      6    0    emc_clariion0 600601601 893
```

指定したエンクロージャのすべての DMP ノードのリストを取得するには、getdmpnode で enclosure 属性を使います。

```
# vxddmpadm getdmpnode enclosure=enc0
```

NAME	STATE	ENCLR-TYPE	PATHS	ENBL	DSBL	ENCLR-NAME
sdm	ENABLED	ACME	2	2	0	enc0
sdn	ENABLED	ACME	2	2	0	enc0
sdo	ENABLED	ACME	2	2	0	enc0
sdp	ENABLED	ACME	2	2	0	enc0

特定の DMP ノードの DMP 情報を表示するには、getdmpnode で dmpnodename 属性を使います。

```
# vxddmpadm getdmpnode dmpnodename=emc_clariion0_158
```

NAME	STATE	ENCLR-TYPE	PATHS	ENBL	DSBL	ENCLR-NAME
emc_clariion0_158	ENABLED	EMC_CLARiION	1	1	0	emc_clariion0

## DMP ノードについての統合された情報の表示

vxddmpadm list dmpnode コマンドは DMP (Dynamic Multi-Pathing) ノードの詳細情報を表示します。情報には、エンクロージャ名、LUN シリアル番号、ポート ID 情報、デバイス属性などが含まれます。

次のコマンドは、システムのすべての DMP ノードに関する統合された情報を表示します。

```
# vxddmpadm list dmpnode all
```

指定したエンクロージャのすべての DMP ノードのリストを取得するには、list dmpnode で enclosure 属性を使います。

```
# vxddmpadm list dmpnode enclosure=enclosurename
```

たとえば、次のコマンドを実行すると、enc0 エンクロージャ内のすべての DMP ノードに関する統合された情報が表示されます。

```
# vxddmpadm list dmpnode enclosure=enc0
```

特定の DMP ノードの DMP 情報を表示するには、list dmpnode で dmpnodename 属性を使います。DMP ノードは名前またはパス名で指定できます。指定した DMP ノードの詳細情報には、一覧表示された DMP ノードの各サブパスについてのパス情報が含まれています。

パスの状態は、障害により無効になったパスと、管理上の目的で手動で無効にされたパスでは異なります。vxddmpadm disable コマンドを使って手動で無効にされたパスは、disabled (m) として表示されます。

```
# vxddmpadm list dmpnode dmpnodename=dmpnodename
```

たとえば、次のコマンドを実行すると、DMP ノード `emc_clariion0_158` についての統合された情報が表示されます。

```
# vxddmpadm list dmpnode dmpnodename=emc_clariion0_158
```

```
dmpdev      = emc_clariion0_158
state       = enabled
enclosure   = emc_clariion0
cab-sno     = CK200070400359
asl         = libvxCLARiiON.so
vid         = DGC
pid         = DISK
array-name  = EMC_CLARiiON
array-type  = CLR-A/PF
iopolicy    = MinimumQ
avid        = 158
lun-sno     = 600601601A141B001D4A32F92B49DE11
udid        = DGC%5FDISK%5FCK200070400359%5F600601601A141B001D4A32F92B49DE11
dev-attr    = lun
###path    = name state type transport ctlr hwpath aprotID aprotWWN attr
path        = sdck enabled(a) primary FC c2 c2 A5 50:06:01:61:41:e0:3b:33 -
path        = sdde enabled(a) primary FC c2 c2 A4 50:06:01:60:41:e0:3b:33 -
path        = sdcu enabled secondary FC c2 c2 B4 50:06:01:68:41:e0:3b:33 -
path        = sdbm enabled secondary FC c3 c3 B4 50:06:01:68:41:e0:3b:33 -
path        = sdbw enabled(a) primary FC c3 c3 A4 50:06:01:60:41:e0:3b:33 -
path        = sdbc enabled(a) primary FC c3 c3 A5 50:06:01:61:41:e0:3b:33 -
```

## LUN グループのメンバーの表示

次のコマンドを実行すると、指定した DMP (Dynamic Multi-Pathing) ノードと同じ LUN グループに属する DMP ノードが表示されます。

```
# vxddmpadm getlungroup dmpnodename=dmpnode
```

次に例を示します。

```
# vxddmpadm getlungroup dmpnodename=sdq
```

NAME	STATE	ENCLR-TYPE	PATHS	ENBL	DSBL	ENCLR-NAME
sdo	ENABLED	ACME	2	2	0	encl
sdp	ENABLED	ACME	2	2	0	encl

```
sdq      ENABLED  ACME      2         2         0         enc1
sdr      ENABLED  ACME      2         2         0         enc1
```

## DMP ノード、コントローラ、エンクロージャ、アレイポートによって制御されるパスの表示

`vxddmpadm getsubpaths` コマンドを実行すると、**DMP (Dynamic Multi-Pathing)** が認識するすべてのパスが一覧表示されます。`vxddmpadm getsubpaths` コマンドには、特定の **DMP ノード、コントローラ、エンクロージャ、またはアレイポート上のサブパスを一覧表示するオプション** もあります。アレイポート上のパスを一覧表示するには、エンクロージャ名とアレイポート ID の組み合わせ、またはアレイポートのワールドワイドネーム (WWN) を指定します。

**DMP** が認識するすべてのサブパスを一覧表示するには、次のコマンドを実行します。

```
# vxddmpadm getsubpaths
```

```
NAME      STATE [A]  PATH-TYPE [M]  DMPNODENAME  ENCLR-NAME  CTLR  ATTRS
=====
sdaf      ENABLED (A)  PRIMARY      ams_wms0_130  ams_wms0    c2    -
sdc       ENABLED      SECONDARY     ams_wms0_130  ams_wms0    c3    -
sdb       ENABLED (A)  -            punr710vm04_disk_24  disk        c0    -
sda       ENABLED (A)  -            punr710vm04_disk_25  disk        c0    -
sdav      ENABLED (A)  PRIMARY      emc_clariion0_1017  emc_clariion0  c3    -
sdbf      ENABLED      SECONDARY     emc_clariion0_1017  emc_clariion0  c3    -
```

`vxddmpadm getsubpaths` コマンドを `dmpnodename` 属性と組み合わせると、`/dev/vx/dmp` ディレクトリで指定した **DMP ノード名** によって制御される **LUN** すべてのパスが表示されます。

```
# vxddmpadm getsubpaths dmpnodename=sdu
```

```
NAME      STATE [A]  PATH-TYPE [M]  CTLR-NAME  ENCLR-TYPE  ENCLR-NAME  ATTRS
=====
sdu       ENABLED (A)  PRIMARY      c2         ACME        enc0        -
sdt       ENABLED      PRIMARY      c1         ACME        enc0        -
```

**A/A** アレイの場合、**I/O** に使える有効パスはすべて **ENABLED (A)** として表示されます。

**I/O** ポリシーが `singleactive` に設定されている **A/P** アレイの場合、1 つのパスのみが **ENABLED (A)** として表示されます。その他のパスは **ENABLED** でも使えません。**I/O** ポリシーが `singleactive` に設定されていない場合、**DMP** は、状態が **ENABLED (A)** と表示されているパスのグループ (すべてプライマリまたはすべてセカンダリ) を使えます。

p.68 の「**I/O** ポリシーの指定」を参照してください。

**DISABLED** 状態のパスは I/O 操作には使えません。

システム管理者によって手動で無効にされたパスは **DISABLED(M)** と表示されます。障害が起きたパスは **DISABLED** と表示されます。

`getsubpaths` を使うと、特定の HBA コントローラに接続しているすべてのパスに関する情報を入手できます。

```
# vxddmpadm getsubpaths ctrlr=c2
```

```
NAME STATE[-] PATH-TYPE[-] DMPNODENAME ENCLR-TYPE ENCLR-NAME ATTRS
=====
sdk   ENABLED (A) PRIMARY      sdk           ACME          enc0         -
sdl   ENABLED (A) PRIMARY      sdl           ACME          enc0         -
sdm   DISABLED SECONDARY   sdm           ACME          enc0         -
sdn   ENABLED SECONDARY   sdn           ACME          enc0         -
```

また、`getsubpaths` を使うと、アレイ上のポートに接続しているすべてのパスに関する情報を入手できます。アレイポートは、エンクロージャ名とアレイポート ID で指定するか、またはアレイポートの **WWN** (ワールドワイド名) 識別子で指定します。

```
# vxddmpadm getsubpaths enclosure=enclosure portid=portid
# vxddmpadm getsubpaths pwwn=pwwn
```

たとえば、エンクロージャとアレイポート ID を指定してアレイポート上のサブパスを一覧表示するには、次のコマンドを実行します。

```
# vxddmpadm getsubpaths enclosure=emc_clariion0 portid=A5
```

```
NAME STATE[A] PATH-TYPE[M] DMPNODENAME ENCLR-NAME CTRLR ATTRS
=====
sdav  ENABLED (A) PRIMARY      emc_clariion0_1017 emc_clariion0 c3 -
sdc  ENABLED (A) PRIMARY      emc_clariion0_1017 emc_clariion0 c2 -
sdau  ENABLED (A) PRIMARY      emc_clariion0_1018 emc_clariion0 c3 -
sdcc  ENABLED (A) PRIMARY      emc_clariion0_1018 emc_clariion0 c2 -
```

たとえば、**WWN** を通してアレイポート上のサブパスを一覧表示するには、次のコマンドを実行します。

```
# vxddmpadm getsubpaths pwwn=50:06:01:61:41:e0:3b:33
```

```
NAME STATE[A] PATH-TYPE[M] CTRLR-NAME ENCLR-TYPE ENCLR-NAME ATTRS
=====
sdav  ENABLED (A) PRIMARY      c3           EMC_CLARiion emc_clariion0 -
sdc  ENABLED (A) PRIMARY      c2           EMC_CLARiion emc_clariion0 -
sdau  ENABLED (A) PRIMARY      c3           EMC_CLARiion emc_clariion0 -
sdcc  ENABLED (A) PRIMARY      c2           EMC_CLARiion emc_clariion0 -
```

```
# vxddmpadm getsubpaths pwwn=20:00:00:E0:8B:06:5F:19
```

getsubpaths を使うと、エンクロージャのすべてのサブパスに関する情報を入手できます。

```
# vxddmpadm getsubpaths enclosure=enclosure_name [ctrl=ctrlname]
```

エンクロージャ上のすべてのサブパスを一覧表示するには、次のコマンドを実行します。

```
# vxddmpadm getsubpaths enclosure=emc_clariion0
NAME          STATE[A]    PATH-TYPE[M]  DMPNODENAME  ENCLR-NAME  CTRL  ATTRS
=====
sdav          ENABLED(A)  PRIMARY      emc_clariion0_1017  emc_clariion0  c3    -
sdbf          ENABLED     SECONDARY    emc_clariion0_1017  emc_clariion0  c3    -
sdau          ENABLED(A)  PRIMARY      emc_clariion0_1018  emc_clariion0  c3    -
sdbe          ENABLED     SECONDARY    emc_clariion0_1018  emc_clariion0  c3    -
```

エンクロージャ上のコントローラのすべてのサブパスを一覧表示するには、次のコマンドを実行します。

```
# vxddmpadm getsubpaths enclosure=Disk ctrl=c1
```

デフォルトでは、vxddmpadm getsubpaths コマンドの出力は、エンクロージャ名、DMP ノード名によってソートされ、さらにその中ではパス名でソートされます。

パス名、DMP ノード名、エンクロージャ名、またはホストコントローラ名に基づいて出力をソートするには、-s オプションを使います。

サブパスの情報をソートするには、次のコマンドを使います。

```
# vxddmpadm -s {path | dmpnode | enclosure | ctrl} getsubpaths ¥
[all | ctrl=ctrl_name | dmpnodename=dmp_device_name | ¥
enclosure=enclr_name [ctrl=ctrl_name | portid=array_port_ID] | ¥
pwwn=port_WWN | tpdnodename=tpd_node_name]
```

p.43 の「[DMP ノードのカスタム名の設定](#)」を参照してください。

## コントローラに関する情報の表示

次の Dynamic Multi-Pathing (DMP) コマンドを実行すると、システム上のすべての HBA コントローラの属性が一覧表示されます。

```
# vxddmpadm listctrl all
```

```
CTRL-NAME  ENCLR-TYPE  STATE  ENCLR-NAME  PATH_COUNT
=====
c1          OTHER       ENABLED  other0      3
c2          X1          ENABLED  jbod0       10
```

```
c3          ACME          ENABLED  enc0          24
c4          ACME          ENABLED  enc0          24
```

この出力では、コントローラ c1 はエンクロージャタイプが OTHER であるため、認識される DMP カテゴリに含まれないディスクに接続されていることがわかります。

その他のコントローラは、認識される DMP カテゴリに含まれるディスクに接続されています。

すべてのコントローラが、I/O 操作に利用可能であることを示す ENABLED 状態になっています。

状態が DISABLED であれば、コントローラが I/O 操作に利用不能であるという意味です。利用不能の場合は、ハードウェア障害が起きているか、vxddmpadm disable コマンドを使ってそのコントローラ上での I/O 操作が無効にされている可能性があります。

この形式のコマンドを実行すると、特定のエンクロージャまたは特定のエンクロージャタイプに属するコントローラが一覧表示されます。

```
# vxddmpadm listctlr enclosure=enc0
```

または

```
# vxddmpadm listctlr type=ACME
```

```
CTLR-NAME  ENCLR-TYPE  STATE  ENCLR-NAME  PATH_COUNT
=====
c2          ACME        ENABLED  enc0        10
c3          ACME        ENABLED  enc0        24
```

vxddmpadm getctlr コマンドを実行すると、HBA ベンダーの詳細とコントローラ ID が表示されます。iSCSI デバイスでは、コントローラ ID は IQN または IEEE 形式に基づく名前になります。FC デバイスの場合、コントローラ ID は WWN です。WWN は ESD から取得されます。したがって、ESD が動作していなければこのフィールドはブランクになります。ESD は、イベントの発生を DDL に通知するために使われるデーモンプロセスです。「コントローラ ID」として表示される WWN は、ホストコントローラに関連付けられた HBA ポートの WWN にマップされます。

```
# vxddmpadm getctlr c5
```

```
LNAME      PNAME  VENDOR  CTLR-ID
=====
c5         c5     qllogic  20:07:00:a0:b8:17:e1:37
```

## エンクロージャに関する情報の表示

DMP (Dynamic Multi-Pathing) は、可能な場合、エンクロージャタイプ、エンクロージャシリアル番号、状態、アレイタイプ、LUN 数、ファームウェアバージョンといったエンクロージャの属性を表示できます。

指定されたエンクロージャの属性を表示するには、次の DMP コマンドを使います。

```
# vxdmpadm listenclosure emc0
ENCLR_NAME ENCLR_TYPE ENCLR_SNO          STATUS   ARRAY_TYPE LUN_COUNT FIRMWARE
=====
emc0       EMC           000292601383      CONNECTED A/A        30        5875
```

システムのすべてのエンクロージャの属性を表示するには、次の DMP コマンドを使います。

```
# vxdmpadm listenclosure all
ENCLR_NAME      ENCLR_TYPE      ENCLR_SNO      STATUS   ARRAY_TYPE LUN_COUNT FIRMWARE
=====
Disk            Disk            DISKS          CONNECTED Disk        6         -
emc0            EMC             000292601383  CONNECTED A/A        1         5875
hitachi_usp-vm0 Hitachi_USP-VM  25847          CONNECTED A/A        1         6008
emc_clariion0   EMC_CLARiiON    CK20007040035 CONNECTED CLR-A/PF  2         0324
```

## アレイポートに関する情報の表示

アレイポートについての情報を表示するには、このセクションの DMP (Dynamic Multi-Pathing) コマンドを使います。アレイポートに関して表示される情報には、そのエンクロージャの名前、ID、WWN 識別子が含まれます。

パス、DMP ノード、HBA コントローラを介してアクセス可能なアレイポートの属性を表示するには、次のいずれかのコマンドを使います。

```
# vxdmpadm getportids path=path_name
# vxdmpadm getportids dmpnodename=dmpnode_name
# vxdmpadm getportids ctlr=ctlr_name
```

次のコマンド形式を実行すると、特定のエンクロージャ内のすべてのアレイポートに関する情報を表示します。

```
# vxdmpadm getportids enclosure=enclr_name
```

次の例では、DMP ノード `sdg` を介してアクセス可能なアレイポートに関する情報が表示されます。

```
# vxddmpadm getportids dmpnodename=sdg
```

```
NAME          ENCLR-NAME  ARRAY-PORT-ID pWWN
=====
sdg           HDS9500V0  1A             20:00:00:E0:8B:06:5F:19
```

## ALUA アレイのためのユーザーフレンドリな CLI 出力

DMP は ALUA 標準を使うストレージアレイをサポートします。Veritas InfoScale 7.1 以降より、DMP はマルチコントローラ (2 台より多くのコントローラ) を搭載する ALUA 準拠アレイをサポートします。

`dmp_display_alua_states` チューニングパラメータは、ALUA アレイに対して **PATH-TYPE[M]** 列に **PRIMARY** または **SECONDARY** を表示する代わりに、論理ユニット番号 (LUN) の非対称アクセス状態を表示します。

---

**メモ:** デフォルトのチューニングパラメータ値は **on** です。

---

ALUA LUN の非対称アクセス状態を表示するには、次のように入力します。

```
# vxddmpadm getsubpaths dmpnodename=dmpnode_name
```

通常の実出力例は、次のとおりです。

```
# vxddmpadm getsubpaths dmpnodename=emc_clariion0_786
```

```
NAME          STATE [A]    PATH-TYPE [M]          CTLR-NAME ENCLR-TYPE  ENCLR-NAME  ATTRS
=====
hdisk40      ENABLED      Active/Non-Optimized  fscsi0    EMC_CLARiion  emc_clariion0  -
hdisk58      ENABLED      Active/Non-Optimized  fscsi1    EMC_CLARiion  emc_clariion0  -
hdisk67      ENABLED (A)  Active/Optimized (P)  fscsi1    EMC_CLARiion  emc_clariion0  -
hdisk77      ENABLED (A)  Active/Optimized (P)  fscsi0    EMC_CLARiion  emc_clariion0  -
```

---

**メモ:** この出力の (P) は、バスがデバイスサーバーによる推奨にマークが付いているターゲットポートグループに接続されることを意味します。

---

これまで **PATH-TYPE[M]** 列に **PRIMARY** または **SECONDARY** を表示していたすべての **VxVM/DMP** 出力には、非対称アクセス状態が表示されます。

**PATH-TYPE[M]** 列に **PRIMARY** または **SECONDARY** を表示する CLI 出力の前のバージョンに戻る場合は、次のコマンドを入力して `dmp_display_alua_states` チューニングパラメータを無効にします。

```
# vxddmpadm settune dmp_display_alua_states=off
```

チューニングパラメータ値が即座に変更されます。

## サードパーティ製のドライバにより制御されるデバイスに関する情報の表示

サードパーティ製ドライバ (TPD) 共存機能を使うと、DMP (Dynamic Multi-Pathing) の監視機能を残したまま、サードパーティ製マルチパス化ドライバによって制御されている I/O に DMP をバイパスさせることができます。次のコマンドは、指定した TPD デバイスに対して、DMP が検出したパスを表示するコマンドおよび指定した TPD 制御ノードに対して DMP が検出した対応する TPD デバイスを表示するコマンドです。

```
# vxdmadm getsubpaths tpdnodename=TPD_node_name
# vxdmadm gettpdnode nodename=TPD_path_name
```

p.105 の「サードパーティ製ドライバ制御のエンクロージャに対するデバイスの命名の変更」を参照してください。

たとえば、EMC Symmetrix アレイ内の PowerPath によって制御され、かつ、DMP にも認識される次のようなディスクを想定します。

```
# vxdisk list
```

DEVICE	TYPE	DISK	GROUP	STATUS
emcpowerp	auto:cdsdisk	-	-	online
emcpowerq	auto:cdsdisk	-	-	online
emcpowerr	auto:cdsdisk	-	-	online
emcpowers	auto:cdsdisk	-	-	online
emcpowert	auto:cdsdisk	-	-	online

次のコマンドを実行すると、DMP が PowerPath 制御のノード emcpowerp に対応するパスを検出し、表示します。

```
# vxdmadm getsubpaths tpdnodename=emcpowerp
```

NAME	TPDNODENAME	PATH-TYPE [-]	DMPNODENAME	ENCLR-TYPE	ENCLR-NAME
sdt	emcpowerp	-	emcpowerp	PP_EMCLARiION	pp_emc_clariion0
sdo	emcpowerp	-	emcpowerp	PP_EMCLARiION	pp_emc_clariion0
sdj	emcpowerp	-	emcpowerp	PP_EMCLARiION	pp_emc_clariion0
sde	emcpowerp	-	emcpowerp	PP_EMCLARiION	pp_emc_clariion0

逆に次のコマンドは、DMP が、パス sdt に対応する PowerPath ノードを検出し、PowerPath ノードに関する情報を表示します。

```
# vxdmadm gettpdnode nodename=sdt
```

NAME	STATE	PATHS	ENCLR-TYPE	ENCLR-NAME
------	-------	-------	------------	------------

```
=====
emcpowerp          ENABLED          4          PP_EMCC_LARiion  pp_emc_clariion0
```

## 拡張デバイス属性の表示

DDL (Device Discovery Layer の略でデバイス検出層の意味) 拡張属性は、DDL によって検出される VxVM (Veritas Volume Manager)、DMP (Dynamic Multi-Pathing) LUN またはディスクに対応する属性またはフラグです。これらの属性によって、LUN が特定のハードウェアカテゴリに識別されます。

表 3-1 は、カテゴリの一覧です。

表 3-1 拡張属性のカテゴリ

カテゴリ	説明
ハードウェア RAID のタイプ	LUN が属するストレージ RAID グループのタイプを表示します。
シンプロビジョニングの検出と再生	LUN のシン再生機能を表示します。
デバイスメディアのタイプ	メディアのタイプについて、ソリッドステートドライブ (SSD) かどうかを表示します。
ストレージベースのスナップショット/クローン	LUN がプライマリ LUN のスナップショットまたはクローンのどちらであるかを表示します。
ストレージベースのレプリケーション	LUN が、リモートサイト全体にわたってレプリケートされるグループの一部であるかどうかを表示します。
トランスポート	この LUN への接続に使われる HBA のタイプ (FC、SATA、iSCSI) を表示します。

各 LUN には、これらの拡張属性が 1 つ以上存在する場合があります。DDL は、ASL (Array Support Library) からのデバイス検出中に拡張属性を検出します。また、VOM (Veritas Operations Manager) がある場合、DDL は、管理対象ホストとして設定されているホストの VOM 管理サーバーから拡張属性を取得することもできます。

vxdisk -p list コマンドを実行すると、DDL 拡張属性が表示されます。たとえば、次のコマンドを実行すると、この LUN の std、fc、RAID\_5 属性が表示されます。

```
# vxdisk -p list
DISK          : tagmastore-usp0_0e18
DISKID        : 1253585985.692.rx2600h11
VID           : HITACHI
UDID          : HITACHI%5FOPEN-V%5F02742%5F0E18
REVISION      : 5001
PID           : OPEN-V
```

```

PHYS_CTLR_NAME : 0/4/1/1.0x50060e8005274246
LUN_SNO_ORDER  : 411
LUN_SERIAL_NO  : 0E18
LIBNAME        : libvxhdsusp.sl
HARDWARE_MIRROR: no
DMP_DEVICE     : tagmastore-usp0_0e18
DDL_THIN_DISK  : thick
DDL_DEVICE_ATTR: std fc RAID_5
CAB_SERIAL_NO  : 02742
ATYPE          : A/A
ARRAY_VOLUME_ID: 0E18
ARRAY_PORT_PWWN: 50:06:0e:80:05:27:42:46
ANAME          : TagmaStore-USP
TRANSPORT      : FC
    
```

vxddisk -x attribute -p list コマンドを実行すると、プロパティリストと属性の 1 行リストが表示されます。次の例は、hdprclm 属性を使ってシン再生をサポートする 2 つの日立製 LUN を示しています。

```

# vxddisk -x DDL_DEVICE_ATTR -p list
DEVICE                DDL_DEVICE_ATTR
tagmastore-usp0_0a7a  std fc RAID_5
tagmastore-usp0_065a  hdprclm fc
tagmastore-usp0_065b  hdprclm fc
    
```

ユーザーは、同じコマンド内に複数の -x オプションを指定することによって複数のエントリを表示できます。次に例を示します。

```

# vxddisk -x DDL_DEVICE_ATTR -x VID -p list

DEVICE                DDL_DEVICE_ATTR  VID
tagmastore-usp0_0a7a  std fc RAID_5    HITACHI
tagmastore-usp0_0a7b  std fc RAID_5    HITACHI
tagmastore-usp0_0a78  std fc RAID_5    HITACHI
tagmastore-usp0_0a79  std fc RAID_5    HITACHI
tagmastore-usp0_065a  hdprclm fc        HITACHI
tagmastore-usp0_065b  hdprclm fc        HITACHI
tagmastore-usp0_065c  hdprclm fc        HITACHI
tagmastore-usp0_065d  hdprclm fc        HITACHI
    
```

vxddisk -e list コマンドを使うと、ATTR という名前の最後の列に DDL\_DEVICE\_ATTR プロパティが表示されます。

```

# vxddisk -e list
DEVICE                TYPE  DISK  GROUP  STATUS  OS_NATIVE_NAME  ATTR
    
```

```
tagmastore-usp0_0a7a auto - - online c10t0d2 std fc RAID_5
tagmastore-usp0_0a7b auto - - online c10t0d3 std fc RAID_5
tagmastore-usp0_0a78 auto - - online c10t0d0 std fc RAID_5
tagmastore-usp0_0655 auto - - online c13t2d7 hdprclm fc
tagmastore-usp0_0656 auto - - online c13t3d0 hdprclm fc
tagmastore-usp0_0657 auto - - online c13t3d1 hdprclm fc
```

拡張属性をサポートする **ASL** の一覧とこれらの属性の説明について詳しくは、次の URL にあるハードウェア互換性リスト (**HCL**) を参照してください。

[https://www.veritas.com/support/en\\_US/article.000107677](https://www.veritas.com/support/en_US/article.000107677)

## VxVM の制御下におけるデバイスの無効化と有効化

`vxddmpadm exclude` コマンドを実行すると、指定する基準に基づいてデバイスが **VxVM** (**Veritas Volume Manager**) 制御から除外されます。デバイスは、無効化されると **DMP** (**Dynamic Multi-Pathing**) によって要求されず、結果として **VxVM** で使えなくなります。`vxddmpadm include` コマンドを実行すると、**VxVM** の制御下にデバイスを追加し直すことができます。デバイスは、**VID:PID** の組み合わせ、パス、コントローラ、またはディスクに基づいて追加または除外できます。感嘆符 (!) を使えば、指定したものの以外のパスまたはコントローラを除外または追加できます。

ルートディスクは無効にできません。外部ディスクのベンダー ID とプロダクト ID が、ルートディスクと同じベンダー ID とプロダクト ID で、ルートディスクが **VxVM** でカプセル化されている場合、操作は失敗します。

---

**メモ:** ! 文字は一部のシェルでの特殊文字です。次の構文は、**bash** シェルでこの文字をエスケープ処理する方法を示しています。

---

```
# vxddmpadm exclude { all | product=VID:PID |
ctrl=[\!]ctrlname | dmpnodename=diskname [ path=[\!]pathname ] }

# vxddmpadm include { all | product=VID:PID |
ctrl=[\!]ctrlname | dmpnodename=diskname [ path=[\!]pathname ] }
```

ここで、

<code>all</code>	すべてのデバイス
<code>product=VID:PID</code>	指定した <b>VID:PID</b> を持つすべてのデバイス
<code>ctrl=ctrlname</code>	指定したコントローラ上のすべてのデバイス
<code>dmpnodename=diskname</code>	<b>DMP</b> ノード下のすべてのデバイス

`dmnnode=diskname path=¥!pathname` 指定したものを除く、DMP ノード下のすべてのデバイス

## I/O 統計情報の収集と表示

`vxmpadm iostat` コマンドを使って、指定した DMP ノード、エンクロージャ、パス、ポート、コントローラについての I/O 統計情報を収集し、表示できます。

表示される統計情報は、統計情報の収集に使われた CPU ごとの CPU 使用率とメモリサイズ、読み取り操作および書き込み操作の数、読み取りおよび書き込みが行われた KB 数、読み取りまたは書き込みが行われた KB あたりの平均時間(ミリ秒)です。

統計情報の収集を有効にするには、次のコマンドを入力します。

```
# vxmpadm iostat start [memory=size]
```

`memory` 属性は各 CPU の I/O 統計情報の記録に使うメモリの最大サイズを制限します。デフォルトでは、各 CPU に対して 32k (32 KB) に制限されます。

I/O のカウンタを 0 にリセットするには、次のコマンドを入力します。

```
# vxmpadm iostat reset
```

収集した統計情報を一定の間隔で表示するには、次のコマンドを入力します。

```
# vxmpadm iostat show {filter} [interval=seconds [count=N]]
```

上記のコマンドは `filter` によって指定されたデバイスの I/O 統計情報を表示したものです。`filter` は次のいずれかになります。

- `all`
- `ctrl=ctrl-name`
- `dmnnode=dmp-node`
- `enclosure=enclr-name [portid=array-portid ] [ctrl=ctrl-name]`
- `pathname=path-name`
- `pwwn=array-port-wwn[ctrl=ctrl-name]`

`interval` 属性および `count` 属性を使うと、I/O 統計情報を表示する間隔(秒)と表示する行数をそれぞれ指定できます。統計情報の記録に使えるメモリが十分でない場合は、指定した値よりも間隔が短くなる場合があります。

DMP には、指定した条件で収集した I/O 累積統計情報を表示する `groupby` オプションもあります。

p.60 の「[累積 I/O 統計情報の表示](#)」を参照してください。

統計情報の収集を無効にするには、次のコマンドを入力します。

```
# vxddmpadm iostat stop
```

## 累積 I/O 統計情報の表示

vxddmpadm iostat コマンドにより、さまざまな I/O チャンネルまたは I/O チャンネルの一部にわたる I/O 負荷の分散を分析する機能が提供されます。適切な *filter* を選択して、DMP ノード、コントローラ、アレイエンクロージャ、パス、ポート、仮想マシンの I/O 統計情報を表示します。次に、*groupby* 節を使って、分析する基準に従って累積統計を表示します。*groupby* 節を指定しない場合、統計はパスごとに表示されます。

*filter* と *groupby* 節を組み合わせるときに、必須の使用例のシナリオの I/O 負荷を分析できます。次に例を示します。

- HBA、エンクロージャ、アレイポートにわたる I/O 負荷を比較するには、指定した属性を持つ *groupby* 節を使います。
- 指定の I/O チャンネル (ポートリンクを配列する HBA) をわたる I/O 負荷を分析するには、HBA と PWWN またはエンクロージャとポートで *filter* を使います。
- HBA へのリンクにわたる I/O 負荷の分散を分析するには、HBA による *filter* と *groupby* アレイポートを使います。

iostat コマンドの次の形式を使って I/O 負荷を分析します。

```
# vxddmpadm [-u unit] iostat show [groupby=criteria] {filter} ¥  
    [interval=seconds [count=N]]
```

上記のコマンドは *filter* によって指定されたデバイスの I/O 統計情報を表示したものです。*filter* は次のいずれかになります。

- all
- ctrlr=ctrlr-name
- dmpnodename=dmp-node
- enclosure=enclr-name [portid=array-portid ] [ctrlr=ctrlr-name]
- pathname=path-name
- pwwn=array-port-wwn[ctrlr=ctrlr-name]

次の *groupby* 基準に従って統計情報を集計できます。

- arrayport
- ctrlr
- dmpnode
- enclosure

デフォルトでは、読み書き時間はミリ秒単位で小数点以下2桁まで表示されます。スループットデータは「ブロック」単位で表示され、その出力は拡大縮小されます。つまり、有効数字を一定に保ったまま、小さい値は小さい単位で表示され、大きい値は大きい単位で表示されます。統計データの表示単位を指定できます。-u オプションには次のオプションを指定できます。

- h または H           スループットを最も大きい単位で表示します。
- k                     スループットをキロバイト単位で表示します。
- m                     スループットをメガバイト単位で表示します。
- g                     スループットをギガバイト単位で表示します。
- bytes | b            スループットを正確なバイト数で表示します。
- us                    読み取り書き込みの平均時間をマイクロ秒単位で表示します。

DMP ノードでグループ化するには、次のコマンドを使います。

```
# vxddmpadm [-u unit] iostat show groupby=dmpnode ¥
[all | dmpnodename=dmpnodename | enclosure=enclr-name]
```

コントローラでグループ化するには、次のコマンドを使います。

```
# vxddmpadm [-u unit] iostat show groupby=ctrlr [ all | ctrlr=ctrlr ]
```

次に例を示します。

```
# vxddmpadm iostat show groupby=ctrlr ctrlr=c5
```

CTRLNAME	OPERATIONS		BLOCKS		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
c5	224	14	54	7	4.20	11.10

アレイポートでグループ化するには、次のコマンドを使います。

```
# vxddmpadm [-u unit] iostat show groupby=arrayport [ all ¥
| pwwn=array_pwwn | enclosure=enclr portid=array-port-id ]
```

次に例を示します。

```
# vxddmpadm -u m iostat show groupby=arrayport ¥
enclosure=HDS9500-ALUA0 portid=1A
```

PORTNAME	OPERATIONS		BYTES		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
1A	743	1538	11m	24m	17.13	8.61

エンクロージャでグループ化するには、次のコマンドを使います。

```
# vxddmpadm [-u unit] iostat show groupby=enclosure [ all ¥  
| enclosure=enclr ]
```

次に例を示します。

```
# vxddmpadm -u h iostat show groupby=enclosure enclosure=EMC_CLARiion0
```

OPERATIONS	BLOCKS		AVG TIME (ms)			
	READS	WRITES	READS	WRITES	READS	WRITES
ENCLOSURENAME						
EMC_CLARiion0	743	1538	11392k	24176k	17.13	8.61

すべてのデータエントリがゼロであるエンティティはフィルタで除外することもできます。このオプションは多くのフェールオーバーデバイスを含んでいるクラスタ環境に特に有用です。アクティブパスの統計のみを表示できます。

iostat show コマンドの出力からすべてがゼロのエントリをフィルタ処理するには、次のコマンドを使います。

```
# vxddmpadm [-u unit] -z iostat show [all|ctrl=ctrl_name |  
dmpnodename=dmp_device_name | enclosure=enclr_name [portid=portid]  
|  
pathname=path_name|pwwn=port_WWN] [interval=seconds [count=N]]
```

次に例を示します。

```
# vxddmpadm -z iostat show dmpnodename=emc_clariion0_893
```

```
cpu usage = 9852us    per cpu memory = 266240b
```

PATHNAME	OPERATIONS		BLOCKS		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
sdbc	32	0	258	0	0.04	0.00
sdbw	27	0	216	0	0.03	0.00
sdck	8	0	57	0	0.04	0.00
sdde	11	0	81	0	0.15	0.00

読み取り/書き込みの平均時間をマイクロ秒単位で表示するには

```
# vxddmpadm -u us iostat show pathname=sdck
```

```
cpu usage = 9865us    per cpu memory = 266240b
```

PATHNAME	OPERATIONS		BLOCKS		AVG TIME (us)	
	READS	WRITES	READS	WRITES	READS	WRITES
sdck	8	0	57	0	43.04	0.00

## キューに入れられた I/O または無効な I/O の統計の表示

指定した DMP ノード、または指定したパスまたはコントローラについて、DMP (Dynamic Multi-Pathing) 内のキューに入れられた I/O を表示するには、`-q` オプションを付けた `vxddmpadm iostat show` コマンドを使います。DMP ノードの場合、`-q` オプションを付けると、基盤となるレイヤーに送信された、指定した DMP ノード上の I/O が表示されます。パスまたはコントローラを指定した場合、`-q` オプションを付けると、指定したパスまたはコントローラに送信され、DMP にまだ返されていない I/O が表示されます。

`vxddmpadm iostat` コマンドの詳細については、`vxddmpadm(1m)` マニュアルページを参照してください。

DMP ノード上のキューに入れられた I/O 件数を表示するには、次のコマンドを使います。

```
# vxddmpadm -q iostat show [filter] [interval=n [count=m]]
```

次に例を示します。

```
# vxddmpadm -q iostat show dmpnodename=emc_clariion0_352
```

```
cpu usage = 338us      per cpu memory = 102400b
                        QUEUED I/Os      PENDING I/Os
DMPNODENAME           READS      WRITES
emc_clariion0_352     0          0          0
```

DMP ノード、パス、またはコントローラ上でエラーで返された I/O の件数を表示するには

```
# vxddmpadm -e iostat show [filter] [interval=n [count=m]]
```

たとえば、パス上でエラーを返された I/O の件数を表示するには、次を実行します。

```
# vxddmpadm -e iostat show pathname=sdo
```

```
cpu usage = 637us      per cpu memory = 102400b
                        ERROR I/Os
PATHNAME              READS      WRITES
sdo                   0          0
```

## vxddmpadm iostat コマンドの使用例

DMP (Dynamic Multi-Pathing) では、`vxddmpadm iostat` コマンドを使って I/O 統計情報を収集し、表示できます。ここでは、`vxddmpadm iostat` コマンドを使ったセッションの例を示します。

最初に次のコマンドを実行すると I/O 統計情報の収集が有効になります。

```
# vxddmpadm iostat start
```

続いて次のコマンドを実行すると、読み取りおよび書き込み操作の総数、読み取りおよび書き込みが行われたキロバイト数など、すべてのパスに関する現在の統計情報が表示されます。

```
# vxddmpadm -u k iostat show all
                                cpu usage = 7952us      per cpu memory = 8192b
                                OPERATIONS              BYTES              AVG TIME (ms)
PATHNAME  READS      WRITES      READS      WRITES      READS      WRITES
sdf        87          0      44544k      0          0.00      0.00
sdk         0          0          0          0          0.00      0.00
sdg        87          0      44544k      0          0.00      0.00
sdl         0          0          0          0          0.00      0.00
sdh        87          0      44544k      0          0.00      0.00
sdm         0          0          0          0          0.00      0.00
sdi        87          0      44544k      0          0.00      0.00
sdn         0          0          0          0          0.00      0.00
sdj        87          0      44544k      0          0.00      0.00
sdo         0          0          0          0          0.00      0.00
sdj        87          0      44544k      0          0.00      0.00
sdp         0          0          0          0          0.00      0.00
```

次のコマンドを実行すると、vxddmpadm で統計情報の収集に使われるメモリのサイズが変更されます。

```
# vxddmpadm iostat start memory=4096
```

統計情報は、次のようにパス名、DMP ノード名およびエンクローチャ名ごとにフィルタして表示することができます(各 CPU のメモリサイズは前のコマンドに従って変更されています)。

```
# vxddmpadm -u k iostat show pathname=sdk
                                cpu usage = 8132us      per cpu memory = 4096b
                                OPERATIONS              BYTES              AVG TIME (ms)
PATHNAME  READS      WRITES      READS      WRITES      READS      WRITES
sdk         0          0          0          0          0.00      0.00
```

```
# vxddmpadm -u k iostat show dmpnodename=sdf
                                cpu usage = 8501us      per cpu memory = 4096b
                                OPERATIONS              BYTES              AVG TIME (ms)
PATHNAME  READS      WRITES      READS      WRITES      READS      WRITES
sdf        1088         0      557056k      0          0.00      0.00
```

```
# vxddmpadm -u k iostat show enclosure=Disk
                                cpu usage = 8626us      per cpu memory = 4096b
                                OPERATIONS              BYTES              AVG TIME (ms)
```

PATHNAME	READS	WRITES	READS	WRITES	READS	WRITES
sdf	1088	0	557056k	0	0.00	0.00

統計情報を表示する回数および間隔を指定することもできます。次のコマンドを実行すると、特定のパスについての統計情報の増分が表示されます。この例では、回数は 2 回、間隔は 2 秒と指定しています。

```
# vxddmpadm iostat show pathname=sdk interval=2 count=2
          cpu usage = 9621us      per cpu memory = 266240b
          OPERATIONS              BLOCKS              AVG TIME (ms)
PATHNAME  READS    WRITES    READS    WRITES    READS    WRITES
sdk       0        0         0        0         0.00    0.00

sdk       0        0         0        0         0.00    0.00
```

## エンクロージャへのパスに関する属性の設定

`vxddmpadm setattr` コマンドを使って、エンクロージャまたはディスクアレイへのパスに関する属性を設定できます。

パスに設定された属性は再起動後も、製品アップグレード後も保持されます。

設定できる属性は次のとおりです。

- active**                   スタンバイ(フェールオーバー)パスをアクティブパスに変更します。アレイへのアクティブパスを指定するには、次のコマンドを実行します。

```
# vxddmpadm setattr path sde pathtype=active
```
- nomanual**               パスの元のプライマリ属性またはセカンダリ属性を復元します。**JBOD** ディスクへのパスを復元するには、次のコマンドを実行します。

```
# vxddmpadm setattr path sdm pathtype=nomanual
```
- nopreferred**           パスの通常の優先順位を復元します。パスに対するデフォルトの優先順位を復元するには、次のコマンドを実行します。

```
# vxddmpadm setattr path sdk ¥
pathtype=nopreferred
```

**preferred** [priority=N] 特定のパスを優先パスとして指定し、オプションで優先順位番号を割り当てます。優先順位番号を指定する場合は、1 以上の整数にする必要があります。パスの優先順位の高さは、I/O 負荷の伝送能力の高さを表します。

p.68 の「I/O ポリシーの指定」を参照してください。

アクティブ/アクティブディスクアレイの I/O ポリシーに priority 属性を設定し、パスの優先順位を 2 に指定するには、次のコマンドを実行します。

```
# vxddmpadm setattr enclosure enc0 ¥
  iopolicy=priority
# vxddmpadm setattr path sdk pathtype=preferred ¥
  priority=2
```

**primary** 特定のパスを JBOD ディスクアレイのプライマリパスとして定義します。JBOD ディスクアレイのプライマリパスを指定するには、次のコマンドを実行します。

```
# vxddmpadm setattr path sdm pathtype=primary
```

**secondary** 特定のパスを JBOD ディスクアレイのセカンダリパスとして定義します。JBOD ディスクアレイのセカンダリパスを指定するには、次のコマンドを実行します。

```
# vxddmpadm setattr path sdn pathtype=secondary
```

**standby** 通常の I/O スケジュールには使わないスタンバイパスを設定します。このパスは、I/O に使えるアクティブパスが存在しない場合に使われます。A/P-C アレイのスタンバイパスを指定するには、次のコマンドを実行します。

```
# vxddmpadm setattr path sde pathtype=standby
```

## デバイスまたはエンクロージャの冗長レベルの表示

必要な冗長レベルを下回っているデバイスを一覧表示するには、vxddmpadm getdmpnode コマンドを使います。

指定するエンクロージャで、有効パス数が特定の値よりも少ないデバイスを一覧表示するには、次のコマンドを使います。

```
# vxddmpadm getdmpnode enclosure=enc1_name redundancy=value
```

たとえば、有効パス数が 3 つよりも少ないデバイスを一覧表示するには、次のコマンドを使います。

```
# vxddmpadm getdmpnode enclosure=EMC_CLARiion0 redundancy=3
```

```
NAME                STATE      ENCLR-TYPE  PATHS ENBL DSBL ENCLR-NAME
=====
```

```
emc_clariion0_162 ENABLED EMC_CLARiION 3 2 1 emc_clariion0
emc_clariion0_182 ENABLED EMC_CLARiION 2 2 0 emc_clariion0
emc_clariion0_184 ENABLED EMC_CLARiION 3 2 1 emc_clariion0
emc_clariion0_186 ENABLED EMC_CLARiION 2 2 0 emc_clariion0
```

特定のデバイスの最小冗長レベルを表示するには、次のように `vxddmpadm getattr` コマンドを使います。

```
# vxddmpadm getattr enclosure|arrayname|arraytype ¥
component-name redundancy
```

たとえば、エンクロージャ **HDS9500-ALUA0** の最小冗長レベルを表示するには、次のコマンドを実行します。

```
# vxddmpadm getattr enclosure HDS9500-ALUA0 redundancy

ENCLR_NAME  DEFAULT  CURRENT
=====
HDS9500-ALUA0  0        4
```

## アクティブパスの最小数の指定

デバイスまたはエンクロージャの最小冗長レベルを設定できます。最小冗長レベルは、デバイスまたはエンクロージャに対してアクティブにする必要のあるパスの最小数です。パス数がエンクロージャの最小冗長レベルを下回ると、メッセージがシステムコンソールに送信され、**DMP (Dynamic Multi-Pathing)** ログファイルにも記録されます。また、通知が `vxnotify` クライアントに送信されます。

最小冗長レベルに設定された値は再起動や製品アップグレード後も保持されます。最小冗長レベルを設定しない場合、デフォルト値は **0** です。

最小冗長レベルを設定するには、`vxddmpadm setattr` コマンドを使うことができます。

### アクティブパスの最小数を指定するには

- ◆ 次のように、**redundancy** 属性を使って `vxddmpadm setattr` コマンドを実行します。

```
# vxddmpadm setattr enclosure|arrayname|arraytype component-name
redundancy=value
```

**value** はアクティブパス数です。

たとえば、エンクロージャ **HDS9500-ALUA0** の最小冗長レベルを設定するには、次のコマンドを実行します。

```
# vxddmpadm setattr enclosure HDS9500-ALUA0 redundancy=2
```

## I/O ポリシーの表示

エンクロージャ、アレイまたはアレイタイプに現在設定されている I/O ポリシーおよびデフォルトで設定される I/O ポリシーを表示するには、`vxndmpadm getattr` コマンドを使います。

たとえば、JBOD ディスクに設定されているデフォルトおよび現在の I/O ポリシー (`iopolicy`) を表示するには、次のコマンドを実行します。

```
# vxndmpadm getattr enclosure Disk iopolicy
```

ENCLR_NAME	DEFAULT	CURRENT
-----		
Disk	MinimumQ	Balanced

次の例は、`enc0` エンクロージャ (パーティションサイズを 2 MB とした `balanced` I/O ポリシーが設定されている) の `partitionsizesize` の設定を表示します。

```
# vxndmpadm getattr enclosure enc0 partitionsize
```

ENCLR_NAME	DEFAULT	CURRENT
-----		
enc0	512	4096

## I/O ポリシーの指定

`vxndmpadm setattr` コマンドを使って、ディスクアレイまたはエンクロージャへの複数のパスに I/O 負荷を分散する DMP (Dynamic Multi-Pathing) I/O ポリシーを変更することができます。ポリシーは、特定のエンクロージャ (HDS01 など)、特定のタイプのすべてのエンクロージャ (HDS など)、または特定のアレイタイプのすべてのエンクロージャ (アクティブ/アクティブの場合は A/A、アクティブ/パッシブの場合は A/P など) に対して設定できます。

---

**メモ:** I/O ポリシーはシステムの再起動後にも保持されます。

---

表 3-2 は設定される可能性がある I/O ポリシーを記述します。

表 3-2 DMP I/O ポリシー

ポリシー	説明
adaptive	<p>I/O をパスに動的にスケジューリングしてディスクに対する全体的な I/O スループットを最大化するためのポリシーです。このポリシーは、I/O 負荷が場合によって異なるような環境で使うと便利です。たとえば、I/O 転送が長い場合(テーブルのスキャン)や短い場合(無作為検索)があるデータベースなどで使います。また、このポリシーはパスによってホップ数が異なる SAN 環境でも有効です。このポリシーは DMP で自動的に管理されるため、設定を変更することはできません。</p> <p>たとえば、エンクロージャ enc1 に対して I/O ポリシーを <b>adaptive</b> に設定するには、次のコマンドを実行します。</p> <pre># vxddmpadm setattr enclosure enc1 ¥   iopolicy=adaptive</pre>
adaptiveminq	<p>I/O が各パスの I/O キューの長さに従ってスケジューリングされること以外は <b>adaptive</b> ポリシーと同様です。最も短いキューが付いているパスは最高優先度に割り当てられます。</p>
balanced [partitionsize=size]	<p>ディスクドライブおよび RAID コントローラでのキャッシュ処理を最適化するためのポリシーです。キャッシュのサイズは通常 120 から 500 KB 以上で、各ハードウェアの特性によって異なります。通常の処理時は、ディスク(LUN)は複数の領域(パーティション)に論理的に分割され、指定した領域に対する I/O が 1 つのアクティブパスにのみ送出されます。そのパスに障害が発生した場合は、作業負荷は自動的に別のアクティブパスに分散されます。</p> <p>パーティションのためにサイズを指定するのに <b>partitionsize</b> 属性を使うことができます。パーティションサイズのブロック数は、2 の累乗の値(2 から 231)に調整できます。2 の累乗以外の値は、自動的に適切な値に切り下げられます。</p> <p>0 のパーティションサイズを指定することは、デフォルトのパーティションサイズを指定することと同じです。</p> <p>パーティションサイズのデフォルト値は 512 ブロック(256 KB)です。パーティションサイズに 0 を指定すると、デフォルトのパーティションサイズである 512 ブロック(256 KB)が使われます。</p> <p>デフォルト値を変更するには、<b>dmp_pathswitch_blks_shift</b> チューニングパラメータの値を調整します。</p> <p>p.134 の「DMP チューニングパラメータ」を参照してください。</p> <p><b>メモ:</b> キャッシュサイズより大きい値を設定する場合には、このポリシーを使う利点はありません。</p> <p>たとえば、日立製 HDS 9960 A/A アレイの推奨パーティションサイズは、I/O 処理パターンが主に順次読み取りまたは書き込みである場合、32,768 から 131,072 ブロック(16 MB から 64 MB)です。</p> <p>たとえば、パーティションサイズが 4096 ブロック(2 MB)のエンクロージャ enc0 の <b>balanced</b> I/O ポリシーを設定するには、次のコマンドを実行します。</p> <pre># vxddmpadm setattr enclosure enc0 ¥   iopolicy=balanced partitionsize=4096</pre>

ポリシー	説明
minimumq	<p>LUN のキューに残っている未処理の I/O 要求の数が最も少ないパスに I/O を送出するポリシーです。キューが最も短いパスは DMP で自動的に判断されるため、設定を変更することはできません。</p> <p>たとえば、JBOD に対して I/O ポリシーを minimumq に設定するには、次のコマンドを実行します。</p> <pre># vxddmpadm setattr enclosure Disk ¥   iopolicy=minimumq</pre> <p>これはすべてのアレイでデフォルトの I/O ポリシーです。</p>
priority	<p>SAN 内のパスによって処理効率が異なるため負荷分散を手動で強制的に行う場合に便利なポリシーです。使用可能なパスの設定や処理効率特性、およびシステムのその他の要素にも考慮して、各パスに優先順位を割り当てることができます。</p> <p>p.65 の「エンクロージャへのパスに関する属性の設定」を参照してください。</p> <p>たとえば、すべての SENA アレイに対して I/O ポリシーを priority に設定するには、次のコマンドを実行します。</p> <pre># vxddmpadm setattr arrayname SENA ¥   iopolicy=priority</pre>
round-robin	<p>I/O をラウンドロビンシーケンスのパス間で同等に共有するポリシーです。たとえば、3 つのパスが存在する場合、最初の I/O 要求で 1 つのパスが使われると、2 番目では別のパス、3 番目では残っている 3 つ目のパスが使われ、4 番目の I/O 要求では再度最初のパスというように割り当てられていきます。このポリシーは DMP で自動的に管理されるため、設定を変更することはできません。</p> <p>たとえば、すべてのアクティブ/アクティブアレイに対して I/O ポリシーを round-robin に設定するには、次のコマンドを実行します。</p> <pre># vxddmpadm setattr arraytype A/A ¥   iopolicy=round-robin</pre>
singleactive	<p>I/O を単一のアクティブパスに送信するポリシーです。このポリシーは、1 つのコントローラに 1 つのアクティブパスが存在し、他のパスはフェールオーバーを実行する場合に使われる、A/P アレイ用に設定できます。A/A アレイ用に設定した場合、パス間での負荷分散は実行されず、代替パスは高可用性 (HA) を得る場合にのみ使われます。現在のアクティブパスに障害が発生した場合、I/O は代替アクティブパスに切り替えられます。単一のアクティブパスは DMP で選択されるため、設定を変更することはできません。</p> <p>たとえば、JBOD ディスクに I/O ポリシー singleactive を設定するには、次のコマンドを実行します。</p> <pre># vxddmpadm setattr arrayname Disk ¥   iopolicy=singleactive</pre>

## 非対称アクティブ/アクティブまたは ALUA アレイのパスでの I/O のスケジュール設定

adaptive、balanced、minimumq、priority、round-robin I/O ポリシーとともに use\_all\_paths 属性を指定して、I/O 要求が、非対称アクティブ/アクティブ (A/A-A) アレイまたは ALUA アレイのプライマリパスに加え、セカンダリパスでもスケジュール設定するかどうかを指定できます。アレイの特性によっては、負荷分散の向上の結果として、総 I/O スループットが増加することがあります。ただし、この機能は、アレイベンダーが推奨している場合にかぎり有効にしてください。A/A-A または ALUA 以外のアレイタイプには効果がありません。

たとえば、次のコマンドでは、パーティションサイズが 4096 ブロック (2 MB) のエンクロージャ enc0 の balanced I/O ポリシーを設定し、セカンダリパスで I/O 要求をスケジュール設定できるようにしています。

```
# vxddmpadm setattr enclosure enc0 iopolicy=balanced ¥
    partitionsize=4096 use_all_paths=yes
```

この属性のデフォルト設定は use\_all\_paths=no です。

エンクロージャ、アレイ名、アレイタイプに対する use\_all\_paths の現在の設定を表示できます。これを行うには、use\_all\_paths オプションを vxddmpadm gettattr コマンドに指定します。

```
# vxddmpadm gettattr enclosure HDS9500-ALUA0 use_all_paths
```

```
ENCLR_NAME      ATTR_NAME      DEFAULT CURRENT
=====
HDS9500-ALUA0  use_all_paths  no          yes
```

use\_all\_paths 属性が適用されるのは A/A-A アレイと ALUA アレイのみです。他のアレイの場合、上記コマンドを実行すると次のメッセージが表示されます。

```
Attribute is not applicable for this array.
```

## SAN 環境における負荷分散の適用例

この例では、複数の SAN スイッチを経由した、アクティブ/パッシブデバイスへのプライマリパスが複数ある SAN 環境において、DMP (Dynamic Multi-Pathing) を使って負荷分散を設定する方法について説明します。

vxddisk list コマンドによる次のサンプル出力からわかるように、デバイス sdm には 8 つのプライマリパスがあります。

```
# vxddisk list sdq
```

```
Device: sdq
```

```

.
.
.
numpaths: 8
sdj state=enabled type=primary
sdk state=enabled type=primary
sdl state=enabled type=primary
sdm state=enabled type=primary
sdn state=enabled type=primary
sdo state=enabled type=primary
sdp state=enabled type=primary
sdq state=enabled type=primary

```

さらに、このデバイスはエンクロージャ ENC0 内にあり、ディスクグループ mydg に属し、単純な連結ボリューム myvol1 を含んでいます。

まず、次のコマンドを入力して、DMP 統計情報の収集を有効にします。

```
# vxddmpadm iostat start
```

次に dd コマンドを使って、ボリュームからの入力作業負荷を適用します。

```
# dd if=/dev/vx/rdisk/mydg/myvol1 of=/dev/null &
```

デバイスの DMP 統計情報を表示する vxddmpadm iostat コマンドを実行すると、すべての I/O が 1 つのパス sdq に対して行われていることがわかります。

```
# vxddmpadm iostat show dmpnodename=sdq interval=5 count=2
```

```

.
.
.
cpu usage = 11294us per cpu memory = 32768b

```

PATHNAME	OPERATIONS		KBYTES		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
sdj	0	0	0	0	0.00	0.00
sdk	0	0	0	0	0.00	0.00
sdl	0	0	0	0	0.00	0.00
sdm	0	0	0	0	0.00	0.00
sdn	0	0	0	0	0.00	0.00
sdo	0	0	0	0	0.00	0.00
sdp	0	0	0	0	0.00	0.00
sdq	10986	0	5493	0	0.41	0.00

次の vxddmpadm コマンドを使って、このデバイスを含むエンクロージャの I/O ポリシーを表示します。

```
# vxddmpadm getattr enclosure ENC0 iopolicy
```

```
ENCLR_NAME      DEFAULT          CURRENT
=====
ENC0            MinimumQ       Single-Active
```

この出力から、このエンクロージャのポリシーが `singleactive` に設定されており、その結果、すべての I/O が 1 つのパスで行われていることがわかります。

I/O 負荷を複数のプライマリパスに分散するために、次のようにポリシーを `round-robin` に設定します。

```
# vxddmpadm setattr enclosure ENC0 iopolicy=round-robin
# vxddmpadm getattr enclosure ENC0 iopolicy
```

```
ENCLR_NAME      DEFAULT          CURRENT
=====
ENC0            MinimumQ       Round-Robin
```

次に DMP 統計情報をリセットします。

```
# vxddmpadm iostat reset
```

作業負荷をかけたままの状態にして、I/O ポリシーを複数のプライマリパスへの負荷分散に変更した効果を確認することができます。

```
# vxddmpadm iostat show dmpnodename=sdq interval=5 count=2
```

```
.
.
.
```

```
cpu usage = 14403us per cpu memory = 32768b
```

PATHNAME	OPERATIONS		KBYTES		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
sdj	2041	0	1021	0	0.39	0.00
sdk	1894	0	947	0	0.39	0.00
sdl	2008	0	1004	0	0.39	0.00
sdm	2054	0	1027	0	0.40	0.00
sdn	2171	0	1086	0	0.39	0.00
sdo	2095	0	1048	0	0.39	0.00
sdp	2073	0	1036	0	0.39	0.00
sdq	2042	0	1021	0	0.39	0.00

次のコマンドを入力すると、エンクロージャを **single active I/O** ポリシーに戻すことができます。

```
# vxddmpadm setattr enclosure ENC0 iopolicy=singleactive
```

## パス、コントローラ、アレイポート、DMP ノードに対する I/O の無効化

パス、HBA コントローラ、アレイポート、DMP (Dynamic Multi-Pathing) ノードを介した I/O を無効にすると、DMP は、指定したパスを介して、または指定のコントローラ、アレイポートまたは DMP ノードに接続したパスを介して I/O 要求を発行できなくなります。指定したパスに保留中の I/O がある場合、`vxddmpadm disable` コマンドは、そのパスを無効にする前に、I/O が完了するまで待機します。

DMP は、マルチパスのためにサードパーティ製ドライバ (TPD) を使うコントローラの I/O を無効にする操作をサポートしません。

1 つ以上のパスに対する I/O を無効にするには、次のコマンドを使います。

```
# vxddmpadm [-c|-f] disable path=path_name1[,path_name2,path_nameN]
```

1 つ以上の HBA コントローラに接続されているパスに対する I/O を無効にするには、次のコマンドを使います。

```
# vxddmpadm [-c|-f] disable ctrlr=ctrl_name1[,ctrl_name2,ctrl_nameN]
```

アレイポートに接続したパスに対する I/O を無効にするには、次のいずれかのコマンドを使います。

```
# vxddmpadm [-c|-f] disable enclosure=enclr_name portid=array_port_ID  
# vxddmpadm [-c|-f] disable pwwn=array_port_WWN
```

ここで、アレイポートは、エンクロージャ名とアレイポート ID で指定することも、アレイポートの WWN (World Wide Name) 識別子で指定することもできます。

次に、アレイポートで I/O を無効にする方法の例を示します。

```
# vxddmpadm disable enclosure=HDS9500V0 portid=1A  
# vxddmpadm disable pwwn=20:00:00:E0:8B:06:5F:19
```

特定のパスに対する I/O を無効にするには、ファブリックの 2 つの末端を表すコントローラとポート ID を両方とも指定します。

```
# vxddmpadm [-c|-f] disable ctrlr=ctrl_name enclosure=enclr_name ¥  
portid=array_port_ID
```

特定の DMP ノードの I/O を無効にするには、DMP のノード名を指定します。

```
# vxddmpadm [-c|-f] disable dmpnodename=dmpnode
```

`-c` オプションを使うと、ディスクへの有効なパスが 1 つだけであるかどうかを確認できません。

デバイスが使用中かどうかに関係なく、`-f` オプションを使用した最後のパスの無効化操作は失敗します。

disable 操作が単一のパスを介してルートディスクに接続されたコントローラに対して発行され、代替パス上に設定されたルートディスクミラーがない場合、その操作は失敗します。そのようなミラーが存在する場合は、コマンドは成功します。disable 操作は、1本のパスのみでスワップ デバイスに接続されているコントローラに対して発行されると失敗します。

## パス、コントローラ、アレイポート、DMP ノードに対する I/O の有効化

コントローラを有効にすると、以前に無効にされたパス、HBA コントローラ、アレイポート、DMP (Dynamic Multi-Pathing) ノードで I/O を再び受け入れられるようになります。この操作は、パス、コントローラ、アレイポート、DMP ノードがホストにアクセス可能であり、このホスト上で I/O を実行できる場合にのみ成功します。アクティブ/パッシブディスクアレイを接続している場合に、enable 操作を行うと、プライマリパスに対する I/O のフェールバックが実行されます。同様に、以前に切断されたシステムボード上のコントローラに対する I/O を行うこともできるようになります。

---

**メモ:** この操作は、クラスタ共有ディスクグループが設定されているディスクアレイへのアクセスに使われるコントローラでサポートされます。

---

DMP は、マルチパスのためにサードパーティ製ドライバ (TPD) を使うコントローラの I/O を有効にする操作をサポートしません。

1 つ以上のパスに対する I/O を有効にするには、次のコマンドを使います。

```
# vxddmpadm enable path=path_name1[,path_name2,path_nameN]
```

1 つ以上の HBA コントローラに接続されているパスに対する I/O を有効にするには、次のコマンドを使います。

```
# vxddmpadm enable ctrl=ctrl_name1[,ctrl_name2,ctrl_nameN]
```

アレイポートに接続したパスに対する I/O を有効にするには、次のいずれかのコマンドを使います。

```
# vxddmpadm enable enclosure=enclr_name portid=array_port_ID  
# vxddmpadm enable pwwn=array_port_WWN
```

ここで、アレイポートは、エンクローージャ名とアレイポート ID で指定することも、アレイポートの WWN (World Wide Name) 識別子で指定することもできます。

次に、アレイポートで I/O を有効にするコマンドの使用例を示します。

```
# vxddmpadm enable enclosure=HDS9500V0 portid=1A  
# vxddmpadm enable pwwn=20:00:00:E0:8B:06:5F:19
```

特定のパスに対する I/O を有効にするには、ファブリックの 2 つの末端を表すコントローラとポート ID を両方とも指定します。

```
# vxddmpadm enable ctrl=ctrl_name enclosure=enclr_name ¥  
    portid=array_port_ID
```

特定の DMP ノードの I/O を有効にするには、DMP のノード名を指定します。

```
# vxddmpadm enable dmpnodename=dmpnode
```

## エンクロージャ名の変更

vxddmpadm setattr コマンドを使って、既存のエンクロージャに意味のある名前を設定することができます。例を次に示します。

```
# vxddmpadm setattr enclosure emc0 name=GRP1
```

この例では、エンクロージャ名を emc0 から GRP1 に変更しています。

---

**メモ:** エンクロージャ名の接頭辞の長さは、最大 23 文字です。

---

次のコマンドは、変更された名前を表示します。

```
# vxddmpadm listenclosure all
```

ENCLR_NAME	ENCLR_TYPE	ENCLR_SNO	STATUS	ARRAY_TYPE	
Disk	Disk	DISKS	CONNECTED	Disk	6
-					
GRP1	EMC	000292601383	CONNECTED	A/A	1
5875					
hitachi_usp-vm0	Hitachi_USP-VM	25847	CONNECTED	A/A	1
6008					
emc_clariion0	EMC_CLARiion	CK20007040035	CONNECTED	CLR-A/PF	2
0324					

## I/O エラーに対する応答の設定

指定したエンクロージャ、ディスクアレイ名、またはアレイタイプへのパスでエラーの発生した I/O 要求に対して DMP (Dynamic Multi-Pathing) がどのように応答するかを設定できます。デフォルトでは、DMP は、エラーになった I/O 要求を各種アクティブパス上で最大 5 回再試行する設定になっています。

エンクロージャ、アレイ名、アレイタイプへのパスに適用されている I/O 要求エラー時の処理に対する現在の設定を表示するには、`vxddmpadm getattr` コマンドを使います。

p.80 の「リカバリオプション値の表示」を参照してください。

DMP がパス上で I/O 要求の送信を再試行する回数について制限を設定するには、次のコマンドを使います。

```
# vxddmpadm setattr ¥
  {enclosure enc-name|arrayname name|arraytype type} ¥
  recoveryoption=fixedretry retrycount=n
```

`retrycount` に対する引数の値には、DMP が別の利用可能なパスで I/O 要求を再度スケジュール設定するまで、またはすべての要求を失敗するまでに再試行する回数を指定します。

固定した試行回数を指定する代わりに、DMP が I/O 要求を再試行できる回数を指定できます。I/O 要求がその回数内に成功しない場合、DMP は I/O 要求を失敗します。`iotimeout` 値を指定するには、次のコマンドを使用します。

```
# vxddmpadm setattr ¥
  {enclosure enc-name|arrayname name|arraytype type} ¥
  recoveryoption=timebound iotimeout=seconds
```

`iotimeout` のデフォルト値は、300 秒です。Oracle などの一部のアプリケーションでは、`iotimeout` をもっと大きい値に設定の方が望ましい場合があります。DMP の `iotimeout` 値は下位オペレーティングシステム層の I/O 処理時間より大きい値にします。

---

**メモ:** `fixedretry` 設定と `timebound` 設定は相互に排他的です。

---

次の例では、エンクロージャ `enc0` に対するリカバリを期限付きで設定し、`iotimeout` の値を 360 秒に設定します。

```
# vxddmpadm setattr enclosure enc0 recoveryoption=timebound ¥
  iotimeout=360
```

次の例では、すべてのアクティブ/アクティブアレイへのパスに対して固定再試行限度を 10 回に設定します。

```
# vxddmpadm setattr arraytype A/A recoveryoption=fixedretry ¥
  retrycount=10
```

`recoveryoption=default` を指定すると DMP をリカバリのデフォルト設定にリセットします。

たとえば、次のコマンドはデフォルト設定を設定します:

```
# vxddmpadm setattr arraytype A/A recoveryoption=default
```

PCI デバイスのデフォルト設定は `recoveryoption=fixedretry retrycount=5` です。

その他のデバイスのデフォルト設定は `recoveryoption=timebound iotimeout=300` です。

`recoveryoption=default` を指定すると、I/O 調整がデフォルトの設定になるという影響もあります。

p.78 の「[I/O 調整機構の設定](#)」を参照してください。

---

メモ: I/O エラーへの応答の設定はシステムの再起動後も保持されます。

---

## I/O 調整機構の設定

デフォルトでは、DMP (Dynamic Multi-Pathing) はすべてのパスの I/O 調整がオフになるように設定されます。エンクロージャ、アレイ名、またはアレイタイプへのパスに適用されている I/O 調整の現在の設定を表示するには、`vxddmpadm getattrs` コマンドを使います。

p.80 の「[リカバリオプション値の表示](#)」を参照してください。

I/O 調整を有効にすると、統計情報収集デーモンのアクティビティのために、CPU とメモリに対して少しのオーバーヘッドがかかります。I/O 調整を無効にすると、デーモンは統計情報を収集せず、I/O 調整が再度有効になるまで活動なしのままになります。

I/O 調整をオフにするには、`vxddmpadm setattr` コマンドを次の形式で使います。

```
# vxddmpadm setattr ¥
  {enclosure enc-name|arrayname name|arraytype type} ¥
  recoveryoption=nothrottle
```

次の例は、エンクロージャ `enc0` へのパスに対して I/O 調整を無効にする方法を示します。

```
# vxddmpadm setattr enclosure enc0 recoveryoption=nothrottle
```

`vxddmpadm setattr` コマンドを使うと、指定したエンクロージャ、ディスクアレイ名、アレイタイプへのパスで I/O 調整を有効にできます。

```
# vxddmpadm setattr ¥
  {enclosure enc-name|arrayname name|arraytype type} ¥
  recoveryoption=throttle [iotimeout=seconds]
```

`iotimeout` 属性が指定されている場合、その引数は、DMP がそのパス上の I/O 調整を呼び出すまでに未処理の I/O 要求の成功を待機する時間を秒単位で指定します。

iotimeout のデフォルト値は、10 秒です。iotimeout をもっと大きい値に設定すると、I/O 調整が呼び出される前に、潜在的に SCSI ドライブ内でより多くの I/O 要求がキューに入れられる原因になります。

たとえば次の例では、エンクロージャ enc0 に対して、iotimeout の値を 60 秒に設定します。

```
# vxddmpadm setattr enclosure enc0 recoveryoption=throttle ¥
    iotimeout=60
```

I/O 調整をデフォルト設定にリセットするには、次のように recoveryoption=default を指定します。

```
# vxddmpadm setattr arraytype A/A recoveryoption=default
```

上記のコマンドは、recoveryoption=nothrottle に対応するデフォルトの動作を設定します。上記のコマンドは、I/O エラーに対する応答のデフォルト動作も設定します。

p.76 の「[I/O エラーに対する応答の設定](#)」を参照してください。

---

メモ: I/O 調整設定はシステムの再起動後にも保持されます。

---

## サブパスフェールオーバーグループ(SFG)の設定

サブパスフェールオーバーグループ(SFG)機能は、dmp\_sfg\_threshold チューニングパラメータを使って有効または無効にできます。チューニングパラメータのデフォルト値は 1 で、これは機能が有効であることを表します。

機能を無効にするには、dmp\_sfg\_threshold チューニングパラメータの値を 0 に設定します。

```
# vxddmpadm settune dmp_sfg_threshold=0
```

機能を有効にするには、dmp\_sfg\_threshold の値を、SFG をトリガするのに必要なパスのエラー数に設定します。

```
# vxddmpadm settune dmp_sfg_threshold=N
```

サブパスフェールオーバーグループの ID を表示するには、次のコマンドを使います。

```
# vxddmpadm getportids {ctlr=ctlr_name | dmpnodename=dmp_device_name
¥
    | enclosure=enclr_name | path=path_name}
```

## LIPP (Low-Impact Path Probing) の設定

LIPP (Low-Impact Path Probing) 機能は `vxddmpadm settune` コマンドを使って有効または無効にできます。

```
# vxddmpadm settune dmp_low_impact_probe=[on|off]
```

パスのプロブは、同じ HBA とアレイポートに接続されているパスのサブセットをプロブすることによって最適化されます。パスのサブセットのサイズは `dmp_probe_threshold` チューニングパラメータによって制御できます。デフォルト値は **5** に設定されます。

```
# vxddmpadm settune dmp_probe_threshold=N
```

## リカバリオプション値の表示

エンクロージャ、アレイ名、アレイタイプへのパスに適用されている I/O 要求エラー時の処理に対する現在の設定を表示するには、次の **Dynamic Multi-Pathing (DMP)** コマンドを使います。

```
# vxddmpadm getattr ¥
    {enclosure enc-name|arrayname name|arraytype type} ¥
    recoveryoption
```

次の例では、`vxddmpadm getattr` コマンドを使って、エンクロージャで設定された `recoveryoption` オプション値を表示します。

```
# vxddmpadm getattr enclosure HDS9500-ALUA0 recoveryoption
ENCLR-NAME      RECOVERY-OPTION  DEFAULT[VAL]    CURRENT[VAL]
=====
HDS9500-ALUA0  Throttle         Nothrottle[0]  Nothrottle[0]
HDS9500-ALUA0  Error-Retry      Timebound[300] Timebound[300]
```

このコマンドの出力は、デフォルトと現在のポリシーオプションとその値を示します。

表 3-3 に、エラー後の I/O 再試行に関するリカバリオプションの設定の概略を示します。

表 3-3 エラー後の I/O 再試行に関するリカバリオプション

リカバリオプション	使用可能な設定	説明
<code>recoveryoption=fixedretry</code>	Fixed-Retry (retrycount)	I/O がエラーになった場合、DMP はエラーとなった I/O 要求を指定回数再試行します。
<code>recoveryoption=timebound</code>	Timebound (iotimeout)	I/O がエラーになった場合、DMP はエラーとなった I/O 要求を指定時間 (秒単位) 再試行します。

表 3-4 に、I/O 調整に関するリカバリオプションの設定の概略を示します。

表 3-4 I/O 調整に関するリカバリオプション

リカバリオプション	使用可能な設定	説明
recoveryoption=nothrottle	なし	I/O 調整は使われません。
recoveryoption=throttle	Timebound (iotimeout)	指定時間(秒単位)以内に I/O 要求が戻らない場合、DMP はパスを調整します。

## DMP パスリストポリシーの設定

DMP (Dynamic Multi-Pathing) は、指定した時間間隔に基づきパスの状態を監視するカーネルタスクを保守します。パスに対して実行される分析の種類は、設定されたチェックポリシーに応じて変わります。

---

**メモ:** DMP パスリストアタスクでは、`vxddmpadm disable` を使って無効にしたコントローラを介するパスに対する無効化の状態は変更されません。

---

DMP パスリストポリシーを設定する場合は、パスリストアスレッドを停止してから、新しい属性でそのスレッドを再起動する必要があります。

p.83 の「[DMP パスリストアスレッドの停止](#)」を参照してください。

次のいずれかのリストアポリシーを設定するには、`vxddmpadm settune dmp_restore_policy` コマンドを使います。ポリシーは、リストアスレッドが停止されるか、または `vxddmpadm settune` コマンドを使ってこれらの値が変更されるまで有効なままです。

- `check_all`  
パスリストアスレッドは、システム上のすべてのパスを分析し、オンライン状態に戻っているパスを有効にするとともに、アクセスできないパスを無効にします。このポリシーを設定するコマンドは、次のとおりです。

```
# vxddmpadm settune dmp_restore_policy=check_all
```

- `check_alternate`  
パスリストアスレッドは、オンラインに戻っているパスを有効にするとともに、少なくとも 1 つの代替パスが正常であるかどうかをチェックします。この条件が満たされない場合は、通知が生成されます。このポリシーを使うと、正常なすべてのパスで `inquiry` コマンドを実行する必要がないため、使用可能なパスが多数存在する場合には、`check_all` に比べて無駄が少なくなります。各 DMP ノードのパスが 2 つのみの場

合、このポリシーは `check_all` と同じです。このポリシーを設定するコマンドは、次のとおりです。

```
# vxddmpadm settune dmp_restore_policy=check_alternate
```

- `check_disabled`

デフォルトのパスリストアポリシーです。パスリストアスレッドは、ハードウェアの故障が原因で以前に無効にされたパスの状態をチェックし、オンライン状態に戻っている場合はそれらのパスを再び有効にします。このポリシーを設定するコマンドは、次のとおりです。

```
# vxddmpadm settune dmp_restore_policy=check_disabled
```

- `check_periodic`

パスリストアスレッドは、特定のサイクル数ごとに 1 回 `check_all` を実行し、残りのサイクルで `check_disabled` を実行します。使用可能なパスが多数存在する場合にはこのポリシーを使うと、`check_all` の実行に伴って定期的に処理速度が低下することがあります。このポリシーを設定するコマンドは、次のとおりです。

```
# vxddmpadm settune dmp_restore_policy=check_periodic
```

`check_all` ポリシーの実行間隔のデフォルト値は 10 サイクルです。

`dmp_restore_interval` チューニングパラメータは、パスリストアスレッドがパスを検査する頻度を指定します。たとえば、次のコマンドを実行すると、ポーリング間隔が 400 秒に設定されます。

```
# vxddmpadm settune dmp_restore_interval=400
```

これらの設定はすぐに適用され、再ブート後も保持されます。現在の設定を表示するには、`vxddmpadm gettune` コマンドを使います。

p.134 の「[DMP チューニングパラメータ](#)」を参照してください。

ポリシーまたは間隔を指定せずに `vxddmpadm start restore` コマンドを実行すると、パスリストアスレッドは、以前管理者が `vxddmpadm settune` コマンドを使って設定したポリシーと間隔の永続的な設定で起動します。管理者がポリシーまたは間隔を設定していない場合は、システムのデフォルト値が使われます。システムのデフォルトリストアポリシーは、`check_disabled` です。システムのデフォルト間隔は 300 秒です。

---

**警告:** システムのデフォルト値よりも短い間隔を指定すると、システムパフォーマンスに悪影響を与える可能性があります。

---

## DMP パスリストアスレッドの停止

Dynamic Multi-Pathing (DMP) パスリストアスレッドを停止するには次のコマンドを使います。

```
# vxddmpadm stop restore
```

---

**警告:** パスリストアスレッドを停止すると、自動パスフェールバックは停止します。

---

## DMP パスリストアスレッドの状態の表示

DMP (Dynamic Multi-Pathing) パスリストアスレッドの状態を示すチューニングパラメータ値を表示するには、`vxddmpadm gettune` コマンドを使用します。チューニングパラメータには次のものがあります。

`dmp_restore_state` 自動パスリストアカーネルスレッドの状態。

`dmp_restore_interval` DMP パスリストアスレッドのポーリング間隔。

`dmp_restore_policy` パスの状態の確認のために DMP が使うポリシー。

**DMP パスリストアスレッドの状態を表示するには**

◆ 次のコマンドを実行します。

```
# vxddmpadm gettune dmp_restore_state  
  
# vxddmpadm gettune dmp_restore_interval  
  
# vxddmpadm gettune dmp_restore_policy
```

## アレイポリシーモジュール (Array Policy Module) の設定

DMP (Dynamic Multi-Pathing) は、アレイと組み合わせて使うための APM (Array Policy Module) を提供します。APM は、次の操作を行うためのアレイ固有の手順とコマンドを定義する、動的にロード可能なカーネルモジュール (プラグイン) です。

- アレイ内のディスクへの複数のパスが有効な場合に I/O パスを選択する。
- パスフェールオーバー機構を選択する。
- パスに障害が発生した場合の代替パスを選択する。
- パスの変更を有効にする。
- SCSI 予約または予約解除要求に応答する。

DMP には、アレイを登録する際、これらの機能のデフォルトが設定されています。APM では、DMP によって提供される、または APM の別のバージョンによって提供される既存の設定値の一部または全部を変更できます。

次のコマンドを使って、システムに設定されたすべての APM を表示できます。

```
# vxdumpadm listapm all
```

このコマンドの出力には各モジュールのファイル名、サポートされるアレイタイプ、APM 名、APM のバージョン、モジュールが現在ロードされ使われているかどうかが含まれます。

各モジュールの詳細情報を確認するには、モジュール名をコマンドの引数として指定します。

```
# vxdumpadm listapm module_name
```

APM を追加し、設定するには、次のコマンドを使います。

```
# vxdumpadm -a cfgapm module_name [attr1=value1 ¥  
    [attr2=value2 ...]]
```

オプションの設定属性およびその値は、各アレイの APM に特有のもので、詳しくは、アレイベンダーが提供するドキュメントを参照してください。

---

**メモ:** デフォルトでは、DMP は最新の APM を使います。DMP で以前の APM のバージョンを強制的に使うには、`-a` オプションの代わりに `-u` オプションを指定します。APM の最新バージョンが使用中でなければ、以前のバージョンに切り替わります。

---

`-r` オプションを指定すると、現在ロードされていない APM を削除できます。

```
# vxdumpadm -r cfgapm module_name
```

`vxdumpadm(1M)` のマニュアルページを参照してください。

# ディスクの管理

この章では以下の項目について説明しています。

- [ディスク管理について](#)
- [新しく追加されたディスクデバイスの検出と設定](#)
- [ディスクデバイスの名前の付け方の変更](#)
- [エンクロージャに基づくディスク名と OS に基づくディスク名の関連付けの検出](#)

## ディスク管理について

DMP (Dynamic Multi-Pathing) はマルチポートディスクアレイを管理するために使われます。

p.9 の「[DMP の動作方法](#)」を参照してください。

DMP はディスクアレイのデバイスの検出と設定の処理にデバイス検出層 (DDL) を使います。DDL は、DMP の操作に必要なディスクとその属性を検出します。DDL の管理には `vxddladm` コーティリティを使います。

p.90 の「[デバイス検出層の管理方法](#)」を参照してください。

## 新しく追加されたディスクデバイスの検出と設定

新しいディスクをホストに物理的に接続する場合、または新しいファイバーチャネルデバイスをホストにゾーンする場合、`vxctl enable` コマンドを使って、ボリュームデバイスノードディレクトリを再構築し、DMP (Dynamic Multi-Pathing) の内部データベースを更新してシステムの新しい状態を反映することができます。

DMP データベースを再構築するには、まず **Linux** に新しいディスクを認識させてから、`vxctl enable` コマンドを呼び出します。

`vxdisk scandisks` コマンドを使って、オペレーティングシステムのデバイスツリー内のデバイスをスキャンし、マルチパス化されたディスクの動的再設定を開始することもできます。

システムに追加された新しいデバイスのみを **DMP** でスキャンし、有効または無効にされたデバイスはスキャンしない場合は、次に示すように、どちらかのコマンドに `-f` オプションを指定します。

```
# vxdctl -f enable
# vxdisk -f scandisks
```

ただし、次の構成要素に変更がありシステム構成が修正になった場合は、完全なスキャンを開始します。

- インストール済み **ASL (Array Support Library)**。
- **VxVM** による使用から除外になっているデバイスの一覧。
- **DISKS (JBOD)**、**SCSI3**、外部デバイスの定義。

`vxctl(1M)` マニュアルページを参照してください。

`vxdisk(1M)` マニュアルページを参照してください。

## 部分的なデバイス検出

**DMP (Dynamic Multi-Pathing)** では部分的なデバイス検出をサポートしており、検出プロセスから物理ディスクへのパスを有効または無効にすることができます。

`vxdisk scandisks` コマンドは、**OS** デバイスツリー内のデバイスを再スキャンして、**DMP** 再設定を実行します。`vxdisk scandisks` コマンドにパラメータを指定すると、部分的なデバイス検出を実行できます。たとえば、次のコマンドを指定すると、**DMP** は以前には認識されなかった新しく追加されたデバイスを検出します。

```
# vxdisk scandisks new
```

次の例では、ファブリックデバイスを検出します。

```
# vxdisk scandisks fabric
```

次のコマンドでは、デバイス `sdm` と `sdn` をスキャンします。

```
# vxdisk scandisks device=sdm, sdn
```

または、接頭辞 **!** を使って指定したデバイス以外をすべてスキャンすることもできます。

---

**メモ:!** 文字は一部のシェルでの特殊文字です。次の例では、**bash** シェルでこの文字をエスケープ処理する方法を示します。

---

```
# vxdisk scandisks ¥!device=sdm, sdn
```

また、論理コントローラまたは物理コントローラの一覧に接続している（または接続していない）デバイスをスキャンすることもできます。たとえば、次のコマンドを実行すると、指定した論理コントローラに接続しているデバイスを除く、すべてのデバイスを検出して設定できます。

```
# vxdisk scandisks ¥!ctlr=c1,c2
```

次のコマンドを実行すると、指定した物理コントローラに接続しているデバイスのみを検出できます。

```
# vxdisk scandisks pctlr=c1+c2
```

物理コントローラの一覧の各項目は、+ 文字で区切られます。

`vxmpadm getctlr all` コマンドを使うと、物理コントローラの一覧を取得できます。

`vxdisk scandisks` コマンドに指定する引数は 1 つのみにする必要があります。複数のオプションを指定すると、エラーになります。

`vxdisk(1M)` マニュアルページを参照してください。

## ディスクの検出とディスクアレイの動的な追加について

DMP (Dynamic Multi-Pathing) は ASL (Array Support Library) を使って、アレイ固有のマルチパスサポートを提供します。ASL は、動的にロード可能な共有ライブラリ (DDL のプラグイン) です。ASL は、デバイス検出中にデバイス属性を検出するためのハードウェア固有のロジックを実装します。DMP は、各ディスクアレイに関連付ける必要のある ASL を決定するためのデバイス検出層 (DDL) を提供します。

場合によって、DMP は、LUN をディスク (JBOD) として処理することで、基本的なマルチパスとフェールオーバーの機能性を提供することもできます。

### DMP がデバイスを要求する方法

あらゆるアレイに対して完全に最適化されたサポートを提供したり、より複雑なアレイタイプをサポートするために、DMP (Dynamic Multi-Pathing) はアレイ固有の ASL (Array Support Library) の使用を要求します。ASL は APM (Array Policy Module) とともに提供される場合があります。ASL と APM は、事実上、特定のアレイモデルと DMP の抱き合わせ販売を可能にするアレイ固有のプラグインです。

サポート対象のアレイの一覧については、ハードウェア互換性リストを参照してください。

[https://www.veritas.com/support/en\\_US/article.000107677](https://www.veritas.com/support/en_US/article.000107677)

デバイス検出中に、DDL はインストールされている各デバイスの ASL を調べて、どの ASL がデバイスを要求するかを見つけてます。

デバイスを要求する ASL がなければ、DDL は対応する JBOD 定義があるかどうかを調べます。サポートされていないアレイの JBOD 定義を追加すれば、DMP はそのアレイに対してマルチパス機能を提供できるようになります。JBOD 定義があれば、DDL は DISKS カテゴリでデバイスを要求します。これにより、DMP が使う JBOD (物理ディスク) デバイスのリストに LUN が追加されます。JBOD 定義にキャビネット番号が含まれていれば、DDL はそのキャビネット番号を使って LUN をエンクロージャに分類します。

p.98 の「DISKS カテゴリへのサポートされていないディスクアレイの追加」を参照してください。

DMP は、ASL または JBOD 定義がなくても、非対称論理ユニットアクセス (ALUA) 対応アレイに対して基本的なマルチパス機能を提供できます。DDL は ALUA ディスクのエンクロージャの一部として LUN を要求します。アレイタイプは ALUA として示されます。JBOD 定義を追加すれば、LUN をエンクロージャに分類することもできます。

## ディスクカテゴリ

DMP (Dynamic Multi-Pathing) を使用して認識されるディスクアレイは ASL (Array Support Library) によってサポートされており、ディスクから返されるベンダー ID 文字列 (HITACHI など) に応じて分類されます。

DMP によってマルチパス化できる JBOD のディスクは DISKS カテゴリに分類されます。サポートされないアレイにあるディスクも、DISKS カテゴリに分類されることがあります。

p.98 の「DISKS カテゴリへのサポートされていないディスクアレイの追加」を参照してください。

サポートされるどのカテゴリにも属さず、DMP によるマルチパス化もできない JBOD のディスクは、OTHER\_DISKS カテゴリに分類されます。

## 新しいディスクアレイの DMP サポートの追加

新しいタイプのディスクアレイに対するサポートパッケージを動的に追加できます。ベリタスが開発したサポートパッケージは ASL (アレイサポートライブラリ) の形式で提供されます。ベリタスは VRTSaslapm rpm に対する更新を通して新しいディスクアレイのサポートを提供します。更新済みの VRTSaslapm rpm がダウンロード可能かどうかを判断するには、ハードウェア互換性リストの TechNote を参照してください。ハードウェア互換性リストには、VRTSaslapm rpm をインストールするための、最新のダウンロード用 rpm と手順へのリンクが記載されています。システムがオンラインの間に VRTSaslapm rpm をアップグレードできます。アプリケーションを停止する必要はありません。

ハードウェア互換性リストにアクセスするには、次の URL に移動します。

[https://www.veritas.com/support/en\\_US/article.000107677](https://www.veritas.com/support/en_US/article.000107677)

各 VRTSaslapm rpm は、Dynamic Multi-Pathing のバージョンごとに固有です。インストールしたバージョンの Dynamic Multi-Pathing をサポートする VRTSaslapm rpm をインストールする必要があります。

VRTSaslapm rpm のインストール時に、新しいディスクアレイがシステムに接続されている必要はありません。

p.110 の「[Dynamic Reconfiguration ツールでのターゲット ID への新しい LUN の動的な追加](#)」を参照してください。

最新の VRTSaslapm rpm を削除する必要がある場合は、以前にインストールしたバージョンに戻せます。手順については、『Veritas InfoScale トラブルシューティングガイド』を参照してください。

## 新しいディスクアレイ検出の有効化

vxctl enable コマンドは、すべてのディスクデバイスとその属性をスキャンし、DMP のデバイスリストを更新し、新しいデバイスデータベースを使って DMP を再設定します。ホストを再ブートする必要はありません。

---

**警告:** このコマンドによって、DMP がアレイについて正しく設定されます。この操作を行わないと、VxVM でディスクへの独立したバスが別々のデバイスとして処理され、データの破損を引き起こす可能性があります。

---

新しいディスクアレイ検出を有効化するには、次の手順を実行します。

- ◆ 次のようにコマンドを入力します。

```
# vxctl enable
```

## サードパーティドライバの共存について

Dynamic Multi-Pathing (DMP) のサードパーティ製ドライバ (TPD) 共存機能を使うと、DMP の監視する能力を保持したまま、いくつかのサードパーティ製マルチパス化ドライバによって制御される I/O に DMP (Dynamic Multi-Pathing) をバイパスさせることができます。適切な ASL (Array Support Library) が使用可能でインストールされている場合は、仕様ファイルの設定や特殊コマンドの実行を行うことなく、TPD を使うデバイスを検出できます。DMP の TPD 共存機能では、サードパーティ製マルチパス化ドライバを一切変更せずに共存させることが可能です。

p.105 の「[サードパーティ製ドライバ制御のエンクロージャに対するデバイスの命名の変更](#)」を参照してください。

p.55 の「[サードパーティ製のドライバにより制御されるデバイスに関する情報の表示](#)」を参照してください。

## デバイス検出層の管理方法

デバイス検出層 (DDL) を使うと、ディスクアレイの動的な追加が可能になります。DDL は、Dynamic Multi-Pathing (DMP) の操作に必要なディスクとその属性を検出します。DDL は、次のタスクを実行する `vxddladm` ユーティリティを使って管理します。

- iSCSI デバイスなどの DDL で検出したすべてのデバイスの階層の一覧表示
- iSCSI を含むすべてのホストバスアダプタの一覧表示.
- ホストバスアダプタ上で設定されたポートの一覧表示.
- ホストバスアダプタから設定されたターゲットの一覧表示
- ホストバスアダプタから設定されたデバイスの一覧表示
- iSCSI 操作パラメータの取得または設定.
- サポートされているアレイタイプの一覧表示
- アレイのサポートの DDL への追加
- アレイのサポートの DDL からの削除
- 無効にされたディスクアレイに関する情報の一覧表示
- DISKS (JBOD) カテゴリ内の認識されているディスクの一覧表示
- DISKS カテゴリへの特定ベンダーのディスクの追加
- DISKS カテゴリからのディスクの削除
- 外部デバイスとしてのディスクの追加

次の項で、これらのタスクの詳細を説明します。

`vxddladm (1M)` マニュアルページを参照してください。

### iSCSI を含むすべてのデバイスの一覧表示

iSCSI デバイスなどの DDL で検出したすべてのデバイスの階層を一覧表示できます。

**iSCSI を含むすべてのデバイスを一覧表示するには**

- ◆ 次のようにコマンドを入力します。

```
# vxddladm list
```

出力例を次に示します。

```
HBA fscsi0 (20:00:00:E0:8B:19:77:BE)
  Port fscsi0_p0 (50:0A:09:80:85:84:9D:84)
    Target fscsi0_p0_t0 (50:0A:09:81:85:84:9D:84)
      Device sda
  . . .
HBA iscsi0 (iqn.1986-03.com.sun:01:0003ba8ed1b5.45220f80)
  Port iscsi0_p0 (10.216.130.10:3260)
    Target iscsi0_p0_t0 (iqn.1992-08.com.netapp:sn.84188548)
      Device sdb
      Device sdc
    Target iscsi0_p0_t1 (iqn.1992-08.com.netapp:sn.84190939)
  . . .
```

**iSCSI を含むすべてのホストバスアダプタの一覧表示**

iSCSI のアダプタを含めて、システムで設定されたすべてのホストバスアダプタ (HBA) についての情報を入手できます。

表 4-1 は HBA の情報を示します。

**表 4-1 HBA の情報**

フィールド	説明
ドライバ	HBA を制御するドライバ。
ファームウェア	ファームウェアのバージョン
検出	対象の検出で採用された方法。
状態 (State)	デバイスがオンラインまたはオフラインのどちらであるか。
アドレス (Address)	ハードウェアアドレス。

**iSCSI** を含むすべてのホストバスアダプタを一覧表示するには

- ◆ 次のコマンドを使って、iSCSI デバイスを含む、システムで設定されたすべての HBA を一覧表示します。

```
# vxddladm list hbas
```

## ホストバスアダプタ上で設定されたポートの一覧表示

HBA に設定されたすべてのポートについての情報を取得できます。画面に次の情報が表示されます。

HBA-ID                      親 HBA。  
 状態(State)                デバイスがオンラインまたはオフラインのどちらであるか。  
 アドレス(Address)         ハードウェアアドレス。

ホストバスアダプタ上で設定されたポートを一覧表示するには

- ◆ 次のコマンドを使って、HBA で設定されたポートを取得します。

```
# vxddladm list ports
```

```
PORT-ID    HBA-ID        STATE        ADDRESS
-----
c2_p0      c2            Online       50:0A:09:80:85:84:9D:84
c3_p0      c3            Online       10.216.130.10:3260
```

## ホストバスアダプタまたはポートから設定されたターゲットの一覧表示

ホストバスアダプタまたはポートから設定されたすべてのターゲットについての情報を取得できます。

表 4-2 はターゲットの情報を示します。

**表 4-2**                      ターゲットの情報

フィールド	説明
エイリアス	エイリアス名 (設定されている場合)。
HBA-ID	親 HBA またはポート。
状態	デバイスがオンラインまたはオフラインのどちらであるか。

フィールド	説明
アドレス(Address)	ハードウェアアドレス。

### ターゲットを一覧表示するには

- ◆ すべてのターゲットを一覧表示するには、次のコマンドを使います。

```
# vxddladm list targets
```

出力例を次に示します。

```
TARGET-ID  ALIAS  HBA-ID  STATE  ADDRESS
-----
c2_p0_t0   -       c2      Online 50:0A:09:80:85:84:9D:84
c3_p0_t1   -       c3      Online iqn.1992-08.com.netapp:sn.84190939
```

### ホストバスアダプタまたはポートから設定されたターゲットを一覧表示するには

- ◆ 次のコマンドを使うと、HBA またはポートに基づいてフィルタ処理できます。

```
# vxddladm list targets [hba=hba_name|port=port_name]
```

たとえば、特定の HBA から設定されたターゲットを取得するには、次のコマンドを使います。

```
# vxddladm list targets hba=c2
```

```
TARGET-ID  ALIAS  HBA-ID  STATE  ADDRESS
-----
c2_p0_t0   -       c2      Online 50:0A:09:80:85:84:9D:84
```

## ホストバスアダプタとターゲットから設定されたデバイスの一覧表示

ホストバスアダプタから設定されたすべてのデバイスについての情報を取得できます。

表 4-3 はデバイスの情報を示します。

表 4-3 デバイスの情報

フィールド	説明
デバイス	デバイス名。
Target-ID	親ターゲット。
状態	デバイスがオンラインまたはオフラインのどちらであるか。

フィールド	説明
DDL の状況	デバイスが DDL によって要求されるかどうか。要求される場合、出力には ASL 名も表示されます。

ホストバスアダプタから設定されたデバイスを一覧表示するには

- ◆ 設定されたデバイスを取得するには、次のコマンドを使います。

```
# vxddladm list devices

Device      Target-ID   State      DDL status (ASL)
-----
sda         fscsi0_p0_t0 Online     CLAIMED (libvxemc.so)
sdb         fscsi0_p0_t0 Online     SKIPPED (libvxemc.so)
sdc         fscsi0_p0_t0 Offline    ERROR
sdd         fscsi0_p0_t0 Online     EXCLUDED
sde         fscsi0_p0_t0 Offline    MASKED
```

ホストバスアダプタとターゲットから設定されたデバイスを一覧表示するには

- ◆ 特定の HBA とターゲットから設定されたデバイスを取得するには、次のコマンドを使います。

```
# vxddladm list devices target=target_name
```

## iSCSI 操作パラメータの取得または設定

DDL は、iSCSI デバイスバスのパフォーマンスに影響を与える特定のパラメータを設定および表示するためのインターフェースを提供します。ただし、基盤となる OS フレームワークがこの値の設定をサポートする必要があります。OS のサポートがない場合、vxddladm set コマンドはエラーを返します。

表 4-4 iSCSI のデバイスのパラメータ

パラメータ	デフォルト値	最小値	最大値
DataPDUInOrder	yes	no	yes
DataSequenceInOrder	yes	no	yes
DefaultTime2Retain	20	0	3600
DefaultTime2Wait	2	0	3600
ErrorRecoveryLevel	0	0	2

パラメータ	デフォルト値	最小値	最大値
FirstBurstLength	65535	512	16777215
InitialR2T	yes	no	yes
ImmediateData	yes	no	yes
MaxBurstLength	262144	512	16777215
MaxConnections	1	1	65535
MaxOutStandingR2T	1	1	65535
MaxRecvDataSegmentLength	8182	512	16777215

特定の iSCSI ターゲットのイニシエータ上の iSCSI 操作パラメータを取得するには

- ◆ 次のコマンドを入力します。

```
# vxddladm getiscsi target=tgt-id {all | parameter}
```

このコマンドを使えば、すべての iSCSI 操作パラメータを取得できます。

```
# vxddladm getiscsi target=c2_p2_t0
```

出力例を次に示します。

PARAMETER	CURRENT	DEFAULT	MIN	MAX
DataPDUInOrder	yes	yes	no	yes
DataSequenceInOrder	yes	yes	no	yes
DefaultTime2Retain	20	20	0	3600
DefaultTime2Wait	2	2	0	3600
ErrorRecoveryLevel	0	0	0	2
FirstBurstLength	65535	65535	512	16777215
InitialR2T	yes	yes	no	yes
ImmediateData	yes	yes	no	yes
MaxBurstLength	262144	262144	512	16777215
MaxConnections	1	1	1	65535
MaxOutStandingR2T	1	1	1	65535
MaxRecvDataSegmentLength	8192	8182	512	16777215

特定の iSCSI ターゲットのイニシエータ上で iSCSI 操作パラメータを設定するには

- ◆ 次のようにコマンドを入力します。

```
# vxddladm setiscsi target=tgt-idparameter=value
```

## サポートされているすべてのディスクアレイの一覧表示

この手順を使って、vxddladm コマンドの他の形とともに使う vid 属性と pid 属性の値を取得します。

サポートされているすべてのディスクアレイを一覧表示するには、次の作業を実行します。

- ◆ 次のコマンドを実行します。

```
# vxddladm listsupport all
```

## ディスクアレイライブラリのサポートの無効化

特定のディスクアレイライブラリに依存するディスクアレイのサポートを無効にすることができます。また、特定のベンダーからディスクアレイのサポートを無効にすることもできます。

ディスクアレイライブラリのサポートを無効化するには

- ◆ ディスクアレイライブラリのサポートを無効化するには、次のコマンドでアレイライブラリを指定します。

```
# vxddladm excludearray libname=libname
```

この例に示すように、特定ベンダーのディスクアレイのサポートを無効にすることもできます。

```
# vxddladm excludearray vid=ACME pid=X1
```

```
# vxdisk scandisks
```

## 無効にされたディスクアレイライブラリのサポートの有効化

特定のディスクアレイライブラリに依存するすべてのアレイのサポートを以前に無効にした場合は、この手順を使ってそれらのアレイのサポートを有効にします。この手順では、エクスクルーードリストからライブラリを削除します。

無効にされたディスクアレイライブラリのサポートを有効化するには

- ◆ 特定のディスクアレイライブラリに依存するすべてのアレイのサポートを無効にした場合は、includearray キーワードを使って、除外ファイルリストのエントリを削除できます。

```
# vxddladm includearray libname=libname
```

このコマンドは、アレイライブラリをデータベースに追加し、そのライブラリをデバイスの検出に使えるようにします。

```
# vxdisk scandisks
```

## 無効にされたディスクアレイの一覧表示

現在 **Veritas Volume Manager (VxVM)** による使用が無効化されているすべてのディスクアレイを一覧表示するには

- ◆ 次のようにコマンドを入力します。

```
# vxddladm listexclude
```

## DISKS カテゴリで認識されているディスクの一覧表示

**DISKS (JBOD)** カテゴリ内の認識されているディスクを一覧表示するには

- ◆ 次のようにコマンドを入力します。

```
# vxddladm listjbod
```

## Array Support Library (ASL) の詳細の表示

**DMP (Dynamic Multi-Pathing)** により、**Array Support Library (ASL)** の詳細を表示できます。

**Array Support Library (ASL)** の詳細を表示するには

- ◆ 次のようにコマンドを入力します。

```
# vxddladm listsupport libname=library_name.so
```

このコマンドを実行すると、ベンダー ID (VID)、アレイの製品 ID (PID)、アレイタイプ (A/A または A/P など) とアレイの名前が表示されます。出力例を次に示します。

```
# vxddladm listsupport libname=libvxfujitsu.so
ATTR_NAME          ATTR_VALUE
=====
LIBNAME            libvxfujitsu.so
VID                vendor
PID                GR710, GR720, GR730
                   GR740, GR820, GR840
ARRAY_TYPE         A/A, A/P
ARRAY_NAME         FJ_GR710, FJ_GR720, FJ_GR730
                   FJ_GR740, FJ_GR820, FJ_GR840
```

## DISKS カテゴリへのサポートされていないディスクアレイの追加

アレイで利用できる **Array Support Library (ASL)** がない場合、JBOD デバイスとしてディスクアレイを追加する必要があります。

JBOD は、指定されていないかぎり、アクティブ/アクティブ (A/A) であると見なされます。適切な ASL が利用できない場合、A/A-A、A/P または A/PF アレイは、パス遅延や I/O エラーを避けるため、アクティブ/パッシブ (A/P) JBOD として認識される必要があります。JBOD が ALUA 対応であれば、JBOD は ALUA アレイとして追加されます。

p.9 の「[DMP の動作方法](#)」を参照してください。

---

**警告:** この手順は、VxVM (Veritas Volume Manager) ではサポートされていないアレイに DMP を正しく設定するためのものです。この操作を行わないと、VxVM でディスクへの独立したパスが別々のデバイスとして処理され、データの破損を引き起こす可能性があります。

---

**DISKS** カテゴリへのサポートされていないディスクアレイを追加するには、次の手順を実行します。

- 1 次のコマンドを使って、アレイ内にあるディスクのベンダー ID とプロダクト ID を確認します。

```
# /etc/vx/diag.d/vxscsiinq device_name
```

**device\_name** には、アレイ内にあるいずれかのディスクのデバイス名を指定します。このコマンドで出力されるベンダー ID (VID) とプロダクト ID (PID) の値を書き留めてください。富士通製のディスクの場合は、表示されるシリアル番号の文字数も書き留めておきます。

次の出力例では、ベンダー ID は SEAGATE、プロダクト ID は ST318404LSUN18G です。

```
Vendor id (VID)      : SEAGATE
Product id (PID)    : ST318404LSUN18G
Revision            : 8507
Serial Number       : 0025T0LA3H
```

- 2 データベースなど、アレイ上で設定された VxVM ボリュームにアクセスしているアプリケーションをすべて停止し、アレイに設定されているすべてのファイルシステムと Storage Checkpoint のマウントを解除します。
- 3 アレイがタイプ A/A-A、A/P または A/PF の場合、auto-trespass モードで設定する必要があります。

- 4 次のコマンドを入力して、新しい JBOD カテゴリを追加します。

```
# vxddladm addjbod vid=vendorid [pid=productid] ¥
[serialnum=opcode/pagecode/offset/length] ¥
[cabinetnum=opcode/pagecode/offset/length] policy={aa|ap}]
```

*vendorid* および *productid* には、前の手順で検出した VID および PID を指定します。たとえば *vendorid* は、FUJITSU、IBM または SEAGATE になります。富士通製のデバイスの場合は、length 引数に対してシリアル番号の文字数(10 など)も指定します。アレイがタイプ A/A-A、A/P または A/PF の場合、policy=ap 属性も指定する必要があります。

前述の例の場合、このタイプのディスクアレイを JBOD として定義するコマンドは次のようになります。

```
# vxddladm addjbod vid=SEAGATE pid=ST318404LSUN18G
```

- 5 アレイを VxVM 制御下に置くには、vxdctl enable コマンドを使います。

```
# vxdctl enable
```

p.89 の「[新しいディスクアレイ検出の有効化](#)」を参照してください。

- 6 アレイがサポートされたことを確認するには、次のコマンドを入力します。

```
# vxddladm listjbod
```

前述の例のアレイに対する、このコマンドの出力例を次に示します。

VID	PID	SerialNum (Cmd/PageCode/off/len)	CabinetNum (Cmd/PageCode/off/len)	Policy
SEAGATE	ALL PIDs	18/-1/36/12	18/-1/10/11	Disk
SUN	SESS01	18/-1/36/12	18/-1/12/11	Disk

- 7 アレイが認識されたことを確認するには、次のように `vxddmpadm listenclosure` コマンドを使います。前述のアレイの場合の出力例は次のとおりです。

```
# vxddmpadm listenclosure

ENCLR_NAME ENCLR_TYPE ENCLR_SNO STATUS ARRAY_TYPE LUN_COUNT FIRMWARE
=====
Disk        Disk        DISKS        CONNECTED Disk        2        -
```

アレイのエンクロージャ名およびエンクロージャタイプが、どちらも「Disk」に設定されていることを確認できます。アレイ内にあるディスクを表示するには、`vxddisk list` コマンドを使います。

```
# vxddisk list

DEVICE          TYPE          DISK          GROUP          STATUS
punr710vm04_disk_1 auto:none    -            -            online invalid
punr710vm04_disk_2 auto:none    -            -            online invalid
punr710vm04_disk_3 auto:none    -            -            online invalid
punr710vm04_disk_4 auto:none    -            -            online invalid
sda              auto:none    -            -            online invalid
xiv0_9148        auto:none    -            -            online invalid thinrclm
...
```

- 8 DMP のパスが認識されたことを確認するには、`vxddmpadm getdmpnode enclosure=Disk` コマンドを使います。前述のアレイの場合の出力例は次のとおりです。

```
# vxddmpadm getdmpnode enclosure=Disk

NAME          STATE          ENCLR-TYPE    PATHS  ENBL  DSBL  ENCLR-NAME
=====
punr710vm04_disk_1 ENABLED        Disk          1      1     0     disk
punr710vm04_disk_2 ENABLED        Disk          1      1     0     disk
punr710vm04_disk_3 ENABLED        Disk          1      1     0     disk
punr710vm04_disk_4 ENABLED        Disk          1      1     0     disk
sda           ENABLED        Disk          1      1     0     disk
...
```

この例の出力では、アレイ内のディスクへのパスが 2 つあることが示されています。

詳しくは、`vxddladm help addjbod` コマンドを入力してください。

`vxddladm(1M)` マニュアルページを参照してください。

`vxddmpadm(1M)` マニュアルページを参照してください。

## DISKS カテゴリからのディスクの削除

DISKS カテゴリからディスクを削除するには、このセクションの手順を使います。

DISKS カテゴリからディスクを削除するには、次の手順を実行します。

- ◆ `vxddladm` コマンドに `rmjbod` キーワードを付けて使います。次の例は、ベンダー ID が SEAGATE のディスクを削除するコマンドを示しています。

```
# vxddladm rmjbod vid=SEAGATE
```

## 外部デバイス

DDL (Device Discovery Layer) では、RAM ディスクなど、自動検出不能な一部のデバイスについては検出できないことがあります。このような外部デバイスは、`vxddladm addforeign` コマンドを使うことにより、VxVM (Veritas Volume Manager) で simple ディスクとして使えるようになります。このコマンドを使うと、I/O 処理に DMP を使う必要もなくなります。次の例は、指定したディレクトリにブロックデバイスおよびキャラクタデバイスのエントリを追加する方法を示しています。

```
# vxddladm addforeign blockdir=/dev/foo/dsk chardir=/dev/foo/rdisk
```

ブロックデバイスまたはキャラクタデバイスがドライバによってサポートされていない場合は、次のようにコマンドから省略できます。

```
# vxddladm addforeign blockdir=/dev/foo/dsk
```

このコマンドを実行すると、デフォルトでは、OS によって管理されるデバイスツリー内の、自動検出機構で検出されたデバイスと一致するエントリがすべて無効になります。この動作は、`vxddladm(1M)` マニュアルページの説明に従い、`-f` オプションおよび `-n` オプションを使って上書きできます。

エントリを追加した外部デバイスは、`vxdisk scandisks` コマンドまたは `vxctl enable` コマンドのいずれかを使うと simple ディスクとして検出されるようになります。検出されたこれらのディスクは、自動設定されたディスクと同じ方法で使えます。

外部デバイスのサポートには、次の制限があります。

- 外部デバイスは常に単一パスのディスクと見なされます。自動検出ディスクとは異なり、DMP ノードがありません。
- クラスタ環境の共有ディスクグループについてはサポートされていません。スタンドアロンホストシステムのみがサポートされています。
- PGR (Persistent Group Reservation) 操作についてはサポートされていません。
- DMP の制御下にならないため、障害が発生したディスクを自動的に有効にすること、および DMP 管理コマンドを使うことができません。
- エンクロージャ情報を VxVM から確認できません。これらのデバイスを使って作成されたディスクグループは可用性が低くなります。

- I/O フェンシング機能とクラスタファイルシステム機能は、外部デバイスではサポートされません。

## ディスクデバイスの名前の付け方の変更

ディスクには、エンクロージャに基づく名前の付け方、またはオペレーティングシステムの名前の付け方を使えます。DMP コマンドでは、現在の名前の付け方によってデバイス名が表示されます。

デフォルトの名前の付け方は、エンクロージャに基づく命名 (EBN) です。

ネイティブボリュームで DMP (Dynamic Multi-Pathing) を使う場合、ディスクの名前の付け方は EBN にする必要があり、`use_avid` 属性を `yes`、永続性の属性を `yes` にする必要があります。

ディスクの名前の付け方を変更するには、次の作業を実行します。

- ◆ `vxddladm` のメインメニューから[ディスクの名前の付け方の変更(Change the disk naming scheme)]を選択して、**DMP** で使うディスクの名前の付け方を、希望のものに変更します。このオプションを選択すると、次の画面が表示されます。名前の付け方を変更するには、`y`を入力してください。

または

コマンドラインから名前の付け方を変更します。エンクロージャに基づく名前の付け方を選択するには、次のコマンドを使います。

```
# vxddladm set namingscheme=ebn [persistence={yes|no}] ¥  
[use_avid={yes|no}] [lowercase={yes|no}]
```

オペレーティングシステムに基づく名前の付け方を選択するには、次のコマンドを使います。

```
# vxddladm set namingscheme=osn [persistence={yes|no}] ¥  
[lowercase=yes|no]
```

オプションの `persistence` 引数を使うと、ディスクハードウェアを再設定しシステムを再ブートした後に、**DMP** で表示するディスクデバイスの名前を変更しないでおくかどうかを選択できます。デフォルトでは、エンクロージャに基づく名前の付け方は永続的です。オペレーティングシステムに基づく名前の付け方はデフォルトでは永続的ではありません。

名前の付け方を変更しないで名前の永続性のみ変更するには、現在の名前の付け方の `vxddladm set namingscheme` コマンドを実行し、永続性の属性を指定します。

デフォルトでは、**ASL** によって指定された名前に大文字が含まれていても、エンクロージャ名は小文字に変換されます。したがって、エンクロージャに基づくデバイス名は小文字になります。小文字への変換を無効にするには `lowercase=no` オプションを設定します。

エンクロージャに基づく名前付けでは、`use_avid` オプションを使って、デバイス名のインデックス番号にアレイのボリューム ID を使うかどうかを指定します。デフォルトは `use_avid=yes` です。これにより、デバイス名は `enclosure_avid` のように設定されます。`use_avid` を `no` に設定すると、**DMP** デバイス名は `enclosure_index` のように設定されます。インデックス番号は、デバイスが LUN シリアル番号でソートされた後に割り当てられます。

どちらの方法でも変更は即座に反映されます。

p.104 の「永続的なデバイス名の再生成」を参照してください。

## ディスクの名前の付け方の表示

DMP (Dynamic Multi-Pathing) でのディスクの名前の付け方は、オペレーティングシステムに基づく名前の付け方またはエンクロージャに基づく名前の付け方に設定できます。

このコマンドは DMP ディスクの名前の付け方が現在設定されているかどうかを表示します。また、永続性が有効になっているかどうかなど、ディスクの名前の付け方に関する属性が表示されます。

現在のディスクの名前の付け方とその操作モードを表示するには、次のコマンドを使います。

```
# vxddladm get namingscheme
NAMING_SCHEME      PERSISTENCE LOWERCASE USE_AVID
=====
Enclosure Based   Yes           Yes           Yes
```

p.18 の「[DMP でのディスクデバイスの命名](#)」を参照してください。

## 永続的なデバイス名の再生成

永続的なデバイス命名機能では、ディスクデバイスの名前はシステムの再ブート後も維持されます。デバイス検出層 (DDL) は永続的なデバイス名のデータベースに従ってデバイス名を割り当てます。

オペレーティングシステムに基づく名前の付け方を選択した場合、各ディスク名には通常、ディスクへのパスのうちいずれかの名前が付けられます。ハードウェアを再構成して再ブートすると、再構成前と異なる名前がディスクへのパスに対して生成されます。従って、永続的なデバイス名は実際のパスに対応しなくなる場合があります。これにより、ディスクが使用不可になることはありませんが、ディスク名がそのパスのいずれかと関連することはなくなります。

同様にエンクロージャベースの命名法を選択すると、デバイス名はエンクロージャ名とインデックス番号によって決まります。アレイによって開示される LUN の順序がハードウェア構成により変更される場合、永続的なデバイス名は現在のインデックスを反映しないことがあります。

永続的なデバイス名を再生成するには、次の手順を実行します。

- ◆ 永続的な名前のリポジトリを再生成するには、次のコマンドを使います。

```
# vxddladm [-c] assign names
```

-c オプションは、ユーザーが指定したすべての名前を消去し、自動生成された名前と置換します。

-c オプションを指定しない場合、ユーザーが指定した既存の名前は維持されますが、オペレーティングシステムベースの名前とエンクロージャベースの名前は再生成されます。

## サードパーティ製ドライバ制御のエンクロージャに対するデバイスの命名の変更

デフォルトでは、サードパーティ製ドライバ (TPD) 制御のエンクロージャは TPD が割り当てたノード名に基づいた疑似デバイス名を使います。デバイスの命名をネイティブに変更すると、デバイスには他の **Dynamic Multi-Pathing (DMP)** デバイスと同じ形式で名前が付けられます。デバイスは命名規則がどちらに設定されているかに応じて、オペレーティングシステムの名前 (**OSN**) またはエンクロージャに基づく名前 (**EBN**) を使います。

p.104 の「[ディスクの名前の付け方の表示](#)」を参照してください。

### TPD 制御のエンクロージャに対するデバイスの命名を変更するには

- ◆ ディスクエンクロージャを制御するサードパーティ製ドライバ (TPD) が、適切な **ASL (Array Support Library)** によって共存をサポートしている場合、デフォルトの動作では、TPD が割り当てたノード名に基づいてデバイス名が割り当てられます。vxdmpadm コマンドを使うと、TPD 割り当ての名前とオペレーティングシステムが認識しているデバイス名を切り替えることができます。

```
# vxdmpadm setattr enclosure enclosure_name tpdmode=native|pseudo

tpdmode 属性の引数では、オペレーティングシステムが使う名前 (native) または TPD 割り当てのノード名 (pseudo) を指定します。
```

次の例は、pp\_emc\_clariion0 という名前のエンクロージャに対して、TPD に基づく命名とオペレーティングシステムに基づく命名を切り替えるときのコマンドの使い方を示しています。この例では、デバイスの名前の付け方は **OSN** に設定されます。

```
# vxdisk list
```

DEVICE	TYPE	DISK	GROUP	STATUS
emcpowerp	auto:cdsdisk	-	-	online
emcpowerq	auto:cdsdisk	-	-	online
emcpowerrr	auto:cdsdisk	-	-	online
emcpowers	auto:cdsdisk	-	-	online
emcpowert	auto:cdsdisk	-	-	online

```
# vxdmpadm setattr enclosure pp_emc_clariion0 tpdmode=native
```

```
# vxdisk list
```

DEVICE	TYPE	DISK	GROUP	STATUS
sde	auto:cdsdisk	-	-	online
sdf	auto:cdsdisk	-	-	online
sdg	auto:cdsdisk	-	-	online
sdh	auto:cdsdisk	-	-	online
sdi	auto:cdsdisk	-	-	online

tpdmode が native に設定されると、最小のデバイス番号のパスが表示されます。

## エンクロージャに基づくディスク名と OS に基づくディスク名の関連付けの検出

エンクロージャに基づく名前の付け方を有効にした場合、vxprint コマンドでは、OS に基づいた名前ではなく、エンクロージャに基づいたディスクデバイス名 (ディスクアクセス名) を使用してボリュームの構造が表示されます。

エンクロージャに基づくディスク名と OS に基づくディスク名の関連付けを検出するには

- ◆ 任意のエンクロージャに基づくディスク名に関連付けられたオペレーティングシステムに基づく名前を検出するには、次のいずれかのコマンドを使います。

```
# vxdisk list enclosure-based_name  
# vxdmpadm getsubpaths dmpnodename=enclosure-based_name
```

たとえば、ディスク ENC0\_21 に関連する物理デバイスを検索する適切なコマンドは、次のとおりです。

```
# vxdisk list ENC0_21  
# vxdmpadm getsubpaths dmpnodename=ENC0_21
```

これらのコマンドを実行して、ブロックディスクデバイスおよびキャラクタディスクデバイスのフルパス名を取得するには、表示されたデバイス名の接頭辞として /dev/vx/dmp/ または /dev/vx/rdmp/ を追加します。

# デバイスの動的再設定

この章では以下の項目について説明しています。

- [オンラインの Dynamic Reconfiguration](#) について
- [Dynamic Reconfiguration](#) ツールでの [DMP](#) の制御下にある [LUN](#) のオンラインでの再設定
- [DMP](#) の制御下にある [LUN](#) のオンラインでの手動での再設定
- [アレイ側からの LUN](#) の特性の変更
- [Dynamic Reconfiguration](#) ツールでの [ホストバスアダプタ](#) のオンラインでの交換
- [アレイコントローラファームウェア](#) のオンラインでのアップグレード

## オンラインの Dynamic Reconfiguration について

場合により、システム管理者やストレージ管理者はサーバーにプロビジョニングされた [LUN](#) のセットを変更する必要があります。ホストで再設定の再ブートを実行しないで、[LUN](#) 設定を動的に変更できます。

次のタイプのオンラインの [Dynamic Reconfiguration](#) を実行できます。

- [DMP \(Dynamic Multi-Pathing\)](#) の制御下にある [LUN](#) のオンラインでの再設定  
p.108 の「[Dynamic Reconfiguration](#) ツールでの [DMP](#) の制御下にある [LUN](#) のオンラインでの再設定」を参照してください。  
p.112 の「[DMP](#) の制御下にある [LUN](#) のオンラインでの手動での再設定」を参照してください。
- [ホストバスアダプタ \(HBA\)](#) のオンラインでの交換  
p.119 の「[Dynamic Reconfiguration](#) ツールでの [ホストバスアダプタ](#) のオンラインでの交換」を参照してください。
- [アレイコントローラファームウェア](#) の更新(無停止アップグレードとも呼ばれる)

p.120 の「[アレイコントローラファームウェアのオンラインでのアップグレード](#)」を参照してください。

## Dynamic Reconfiguration ツールでの DMP の制御下にある LUN のオンラインでの再設定

場合により、システム管理者やストレージ管理者はサーバーにプロビジョニングされた LUN のセットを変更する必要があります。ホストで再設定の再ブートを実行しないで、LUN 設定を動的に変更できます。

実行する操作は次のとおりです。

- 既存のターゲット ID からの LUN の動的削除  
p.108 の「[Dynamic Reconfiguration ツールでの既存のターゲット ID からの LUN の動的な削除](#)」を参照してください。
- 新しいターゲット ID への新しい LUN の動的追加  
p.110 の「[Dynamic Reconfiguration ツールでのターゲット ID への新しい LUN の動的な追加](#)」を参照してください。
- 既存のターゲット ID の LUN の置き換え  
p.111 の「[Dynamic Reconfiguration ツールでの既存のターゲット ID からの LUN の動的な交換](#)」を参照してください。
- LUN の特性の変更  
p.118 の「[アレイ側からの LUN の特性の変更](#)」を参照してください。

## Dynamic Reconfiguration ツールでの既存のターゲット ID からの LUN の動的な削除

DMP ( Dynamic Multi-Pathing ) には、既存のターゲット ID からの LUN の削除を簡素化する DR ( Dynamic Reconfiguration ) ツールがあります。各 LUN はホストからマッピングが解除されます。DMP はオペレーティングシステムのデバイススキャンを実行し、オペレーティングシステムのデバイスのツリーをクリーンアップします。

---

**警告:** Dynamic Reconfiguration ツールの外部のデバイス検出操作は、デバイス操作が完了するまで実行しないでください。

---

クラスタのすべてのノードに手順を実行します。

既存のターゲット ID から LUN を動的に削除するには

- 1 削除される LUN にホストされているすべてのアプリケーションとボリュームを停止します。

DMP デバイス上で Linux LVM を使っている LUN の場合は、LVM ボリュームグループからデバイスを削除します。

```
# vgreduce vgnamedevicepath
```

- 2 vxdiskadm ユーティリティを開始します。

```
# vxdiskadm
```

- 3 vxdiskadm メニューから [Dynamic Reconfiguration 機能] オプションを選択します。

- 4 [LUN を削除 (Remove LUNs)] オプションを選択します。

- 5 list と入力するか、Return キーを押し、削除できる LUN のリストを表示します。使用中でない LUN は削除できます。

出力例を次に示します。

```
Select disk devices to remove: [<pattern-list>,all,list]: list
LUN(s) available for removal:
eva4k6k0_0
eva4k6k0_1
eva4k6k0_2
eva4k6k0_3
eva4k6k0_4
emc0_017e
```

- 6 LUN の名前、カンマ(,)で区切った LUN のリスト、正規表現を入力し、削除する LUN を指定します。

たとえば、emc0\_017e と入力します。

- 7 プロンプトで LUN の選択を確認します。

DMP は VxVM による使用から LUN を削除します。

## Dynamic Reconfiguration ツールでの DMP の制御下にある LUN のオンラインでの再設定

- 8 次のプロンプトで、アレイターゲットから LUN を削除します。

```
Enclosure=emc0 AVID=017E
Device=emc0_017e Serial=830017E000
-----
PATH=sda ctrlr=c15 port=7e-a [50:01:43:80:12:08:3c:26]
PATH=sdg ctrlr=c17 port=7e-a [50:01:43:80:12:08:3a:76]
-----
Please remove LUNs with above details from array and
press 'y' to continue removal (default:y):
```

- 9 削除処理を続行するために Dynamic Reconfiguration のツールに戻って[y]を選択します。

DMP は VxVM による使用からのデバイスの削除を完了します。次のような出力が表示されます。

```
Luns Removed
-----
emc0_017e
```

DMP はオペレーティングシステムのデバイスツリーと VxVM デバイスツリーを更新します。

- 10 [終了]を選択して Dynamic Reconfiguration ツールを終了します。

## Dynamic Reconfiguration ツールでのターゲット ID への新しい LUN の動的な追加

DMP (Dynamic Multi-Pathing) には、新しいまたは既存のターゲット ID への新しい LUN の追加を簡素化する DR (Dynamic Reconfiguration) ツールがあります。1 つ以上の新しい LUN が複数の HBA ポートを介してホストにマップされます。認識する LUN に対してオペレーティングシステムのデバイススキャンが実行され、DMP 制御に追加されます。

---

**警告:** Dynamic Reconfiguration ツールの外部のデバイス検出操作は、デバイス操作が完了するまで実行しないでください。

---

クラスタの場合、クラスタ内のすべてのノードでこの手順を実行します。

ターゲット ID に新しい LUN を動的に追加するには

- 1 vxdiskadm ユーティリティを開始します。

```
# vxdiskadm
```

- 2 vxdiskadm メニューから [Dynamic Reconfiguration 機能] オプションを選択します。

- 3 [LUN を追加 (Add LUNs)] オプションを選択します。

- 4 プロンプトが表示されたら、アレイ/ターゲットから LUN を追加します。

- 5 [y] を選択して DMP への LUN の追加を続行します。

DMP はオペレーティングシステムのデバイスツリーと VxVM デバイスツリーを更新します。新しく検出されたデバイスが認識可能になります。

```
Luns Added
```

```
-----  
Enclosure=emc0 AVID=017E
```

```
Device=emc0_017e Serial=830017E000
```

```
PATH=c15t0d6 ctrlr=c15 port=7e-a [50:01:43:80:12:08:3c:26]
```

```
PATH=c17t0d6 ctrlr=c17 port=7e-a [50:01:43:80:12:08:3a:76]
```

- 6 [終了] を選択して Dynamic Reconfiguration ツールを終了します。

## Dynamic Reconfiguration ツールでの既存のターゲット ID からの LUN の動的な交換

DMP (Dynamic Multi-Pathing) には、既存のターゲット ID からの新しい LUN の交換を簡素化する DR (Dynamic Reconfiguration) ツールがあります。各 LUN はホストからマッピングが解除されます。DMP はオペレーティングシステムのデバイススキャンを実行し、オペレーティングシステムのデバイスのツリーをクリーンアップします。

---

**警告:** Dynamic Reconfiguration ツールの外部のデバイス検出操作は、デバイス操作が完了するまで実行しないでください。

---

クラスタの場合、クラスタ内のすべてのノードでこの手順を実行します。

既存のターゲット ID から LUN を動的に置換するには

- 1 削除される LUN にホストされているすべてのアプリケーションとボリュームを停止します。

DMP デバイス上で Linux LVM を使っている LUN の場合は、LVM ボリュームグループからデバイスを削除します。

```
# vgreduce vgnamedevicepath
```

- 2 vxdiskadm ユーティリティを開始します。

```
# vxdiskadm
```

- 3 vxdiskadm メニューから [Dynamic Reconfiguration 機能] オプションを選択します。

- 4 [LUN を置換 (Replace LUNs)] オプションを選択します。

出力が置換可能な LUN の一覧を表示します。LUN 上に開いているものがなく、状態がオンラインまたは nolabel の場合は、その LUN を交換に利用できます。

- 5 置換する LUN を 1 つ以上選択します。

- 6 プロンプトで LUN の選択を確認します。

- 7 アレイ/ターゲットから LUN を削除します。

- 8 削除を続行するために Dynamic Reconfiguration のツールに戻って [y] を選択します。

削除が正常に完了すると、Dynamic Reconfiguration ツールで LUN を追加するためのメッセージが表示されます。

- 9 プロンプトが表示されたら、アレイ/ターゲットから LUN を追加します。

- 10 [y] を選択して LUN の追加を続行します。

DMP はオペレーティングシステムのデバイスツリーと VxVM デバイスツリーを更新します。新しく検出されたデバイスが認識可能になります。

## DMP の制御下にある LUN のオンラインでの手動での再設定

場合により、システム管理者やストレージ管理者はサーバーにプロビジョニングされた LUN のセットを変更する必要があります。ホストで再設定の再ブートを実行しないで、LUN 設定を動的に変更できます。

LUN の動的再設定では、アレイ設定コマンド、OS コマンド、Veritas Volume Manager コマンドが必要です。処理を正確に完了するには、コマンドをホストで正しい順序で実行する必要があります。

実行する操作は次のとおりです。

- 既存のターゲット ID からの LUN の動的削除  
p.113 の「既存のターゲット ID から LUN を動的に手動での削除」を参照してください。
- 新しいターゲット ID への新しい LUN の動的追加  
p.115 の「新しいターゲット ID に新しい LUN を動的に手動での追加」を参照してください。

## 既存のターゲット ID から LUN を動的に手動での削除

この場合、LUN のグループはホスト HBA ポートからマッピングが解除され、オペレーティングシステムのデバイススキャンが実行されます。それ以降 LUN をシームレスに追加するには、追加の手順でオペレーティングシステムのデバイストリーをクリーンアップします。

上位レベルの手順と DMP コマンドは共通です。ただし、オペレーティングシステムのコマンドは Linux のバージョンによって変わることがあります。次の手順では例として Linux Suse10 を使います。

既存のターゲット ID から LUN を動的に削除するには

- 1 Dynamic Reconfiguration の実行前に、`dmp_cache_open` チューニングパラメータが `on` に設定されていることを確認します。この設定がデフォルト値です。

```
# vxddmpadm gettune dmp_cache_open
```

チューニングパラメータが `off` に設定されている場合、`dmp_cache_open` チューニングパラメータを `on` に設定します。

```
# vxddmpadm settune dmp_cache_open=on
```

- 2 ホストから削除する LUN を識別します。次のいずれかを実行します。
  - ストレージアレイ管理を使い、LUN のアレイポリューム ID (AVID) を識別します。
  - アレイから AVID が報告されない場合は、LUN のインデックスを使います。
- 3 VxVM の制御下にある LUN の場合、次の手順を実行します。
  - `vxevac` コマンドを使って、LUN からデータを退避します。  
`vxevac (1M)` のオンラインマニュアルページを参照してください。  
データが待避されたら、次のコマンドを入力し、ディスクグループから LUN を削除します。

```
# vxdg -g diskgroup rmdisk da-name
```

- データが待避されておらず、LUN がサブディスクまたはディスクグループの一部になっている場合は、次のコマンドを入力し、ディスクグループから LUN を削除します。ディスクが共有ディスクグループの一部ならば、-k オプションを使って強制的に削除する必要があります。

```
# vxdg -g diskgroup -k rmdisk da-name
```

- 4 DMP デバイス上で Linux LVM を使っている LUN の場合は、LVM ボリュームグループからデバイスを削除します。

```
# vgreduce vgnamedevicepath
```

- 5 AVID または LUN のインデックスを使う場合は、ストレージレイ管理を使い、手順 2 で識別した LUN のマップやマスクを解除します。

- 6 vxdisk リストから LUN を削除します。クラスタ内のすべてのノードで次のコマンドを入力します。

```
# vxdisk rm da-name
```

この手順は必須です。この手順を実行しない場合は、DMP デバイスツリーにゴーストパスが表示されます。

- 7 手順「6」で削除したデバイスの Linux SCSI デバイスツリーをクリーンアップします。

p.118 の「[LUN の削除後のオペレーティングシステムデバイスツリーの手動でのクリーンアップ](#)」を参照してください。

この手順は必須です。後から新しい LUN をホストに追加する場合は、オペレーティングシステムの SCSI デバイスツリーをクリーンアップして SCSI ターゲット ID を再利用のために解放する必要があります。

- 8 オペレーティングシステムのデバイスツリーをスキャンします。

p.117 の「[LUN の追加または削除後のオペレーティングシステムデバイスツリーのスキャン](#)」を参照してください。

- 9 DMP を使ってデバイススキャンを実行します。この操作はクラスタのすべてのノードで実行する必要があります。次のいずれかのコマンドを入力します。

```
■ # vxdctl enable
```

```
■ # vxdisk scandisks
```

10 次のコマンドを使って DMP のデバイス名データベースを更新します。

```
# vxddladm assign names
```

11 次の質問の回答により、LUN がクリーンに削除されたことを検証します。

- デバイスツリーがクリーンであるかどうか。  
オペレーティングシステムのメタノードが /sys/block ディレクトリから削除されていることを確認します。
- すべての該当する LUN が削除されたかどうか。  
vxdisk list コマンド出力などの DMP ディスクレポート作成ツールで、LUN が正常にクリーンアップされたかどうかを判断します。
- vxdisk list 出力は正しいかどうか。  
vxdisk list の出力に正しい数のパスが表示されていて、ゴーストディスクが含まれていないことを確認します。

これらの質問の回答に「いいえ」がある場合は、手順 5 に戻り、必要な手順を実行します。

すべての質問の回答が「はい」である場合は、LUN は正常に削除されています。

## 新しいターゲット ID に新しい LUN を動的に手動での追加

この場合、新しい LUN グループが複数の HBA ポートを介してホストにマップされます。認識する LUN に対してオペレーティングシステムのデバイススキャンが実行され、DMP 制御に追加されます。

上位レベルの手順と DMP コマンドは共通です。ただし、オペレーティングシステムのコマンドは Linux のバージョンによって変わることがあります。次の手順では例として Linux Suse10 を使います。

新しいターゲット ID に新しい LUN を動的に追加するには

1 Dynamic Reconfiguration の実行前に、dmp\_cache\_open チューニングパラメータが on に設定されていることを確認します。この設定がデフォルト値です。

```
# vxddmpadm gettune dmp_cache_open
```

チューニングパラメータが off に設定されている場合、dmp\_cache\_open チューニングパラメータを on に設定します。

```
# vxddmpadm settune dmp_cache_open=on
```

2 ホストに追加する LUN を識別します。次のいずれかを実行します。

- ストレージアレイ管理を使い、LUN のアレイボリューム ID (AVID) を識別します。

- アレイから AVID が報告されない場合は、LUN のインデックスを使います。
- 3 複数のホスト上の新しいターゲット ID に対して、LUN をマップするかマスクします。
  - 4 オペレーティングシステムのデバイスをスキャンします。  
p.117 の「LUN の追加または削除後のオペレーティングシステムデバイスツリーのスキャン」を参照してください。

すべての LUN が追加されたことが確認されるまで、手順 2 と手順 3 を繰り返します。

- 5 DMP を使ってデバイススキャンを実行します。この操作はクラスタのすべてのノードで実行する必要があります。次のいずれかのコマンドを入力します。

- # vxctl enable
- # vxdisk scandisks

- 6 次のコマンドを使って DMP のデバイス名データベースを更新します。

```
# vxddladm assign names
```

- 7 次の質問の回答により、LUN が正しく追加されたことを検証します。
  - 新しくプロビジョニングした LUN が vxdisk list の出力に表示されるかどうか。
  - 各 LUN に設定されたパスがあるかどうか。

これらの質問の回答に「いいえ」がある場合は、手順 2 に戻り、もう一度手順を開始します。

すべての質問の回答が「はい」である場合は、LUN は正常に追加されています。ここで LUN をディスクグループに追加し、新しいボリュームを作成または既存のボリュームを拡張できます。

dmp\_native\_support チューニングパラメータが ON に設定されていて、新しい LUN に VxVM ラベルがない場合、または新しい LUN が TPD ドライバによって要求されていない場合、LUN は LVM で使用できます。

## オペレーティングシステムのデバイスツリーがクリーンアップされていない場合のターゲット ID 再利用の検出について

以前有効であったオペレーティングシステムのデバイスエントリがクリーンアップされていない LUN または LUN セットのプロビジョニングを再度試みると、次のメッセージが表示されます。また、DMP デバイススキャンと DMP の再設定の間には、DMP の再設定が一時的に禁止されます。

p.118 の「LUN の削除後のオペレーティングシステムデバイスツリーの手動でのクリーンアップ」を参照してください。

```
VxVM vxdisk ERROR V-5-1-14519 Data Corruption Protection Activated  
- User Corrective Action Needed
```

```
VxVM vxdisk INFO V-5-1-14521 To recover, first ensure that the OS  
device tree is up to date (requires OS specific commands).
```

```
VxVM vxdisk INFO V-5-1-14520 Then, execute 'vxdisk rm' on the  
following devices before reinitiating device discovery. <DA names>
```

前のメッセージの意味は新しい LUN が古い LUN のターゲット ID の再利用を試みるということです。デバイスエントリはクリーンされていません。したがって新しい LUN はターゲット ID を利用できません。DMP ではオペレーティングシステムのデバイスツリーがクリーンアップされるまでこの処理は実行されません。

## LUN の追加または削除後のオペレーティングシステムデバイスツリーのスキャン

LUN を追加または削除したら、オペレーティングシステムのデバイスツリーをスキャンし、操作が正常に完了したことを確認します。

Linux には、SCSI バスを再スキャンし、バスにマップされているデバイスを識別するための方法が複数用意されています。これらには次の方法が含まれます。

- /sys ディレクトリでの SCSI スキャン機能
- HBA ベンダーのユーティリティ

### SCSI スキャン機能を使ってスキャンするには

- ◆ 次のコマンドを入力します。

```
# echo '- - -' > /sys/class/scsi_host/host$i/scan
```

3 つのハイフンはチャンネル、ターゲット、LUN 番号を指し、*host\$i* はホストバスアダプタのインスタンスです。この例では、このホストバスアダプタのインスタンス経由で参照できるすべてのチャンネル、ターゲット、LUN をスキャンします。

### HBA ベンダーのユーティリティを使ってスキャンするには

- ◆ HBA ユーティリティのベンダーの指示に従います。次のような例があります。
  - QLogic 社は新しく追加された LUN を動的にスキャンするスクリプトを提供しています。このスクリプトは QLogic 社の Web サイトからダウンロードできます。スクリプトを実行するには、次のコマンドを入力します。

```
# ./ql-dynamic-tgt-lun-disc.sh
```

- Emulex 社は HBAAnywhere スクリプトを提供しています。このスクリプトは Emulex 社の Web サイトからダウンロードできます。このスクリプトには、新しく追加された

LUN を動的にスキャンする LUN スキャンユーティリティが含まれます。ユーティリティを実行するには、次のコマンドを入力します。

```
# lun_scan all
```

## LUN の削除後のオペレーティングシステムデバイスツリーの手動でのクリーンアップ

LUN を削除した後は、オペレーティングシステムのデバイスツリーをクリーンアップする必要があります。

オペレーティングシステムのコマンドは Linux のバージョンによって変わることがあります。次の手順では SUSE 10 を使います。これらの手順のいずれかで必要な結果が得られなければ、Novell 社のサポートに連絡してください。

**LUN の削除後にオペレーティングシステムのデバイスツリーをクリーンアップするには**

- 1 オペレーティングシステムのデータベースからデバイスを削除します。次のコマンドを入力します。

```
# echo 1 > /sys/block/$PATH_SYS/device/delete
```

*PATH\_SYS* は削除するデバイスの名前です。

- 2 次のコマンドを入力したときにデバイスが表示されてはいけません。この手順で、LUN が削除されたことを確認します。

```
# lsscsi | grep PATH_SYS
```

- 3 LUN を削除したら、デバイスをクリーンアップします。次のコマンドを入力します。

```
# echo "-- --" > /sys/class/scsi_host/host$I/scan
```

3 つのハイフンはチャンネル、ターゲット、LUN 番号を指し、*host\$I* はホストバスアダプタのインスタンスです。この例では、このホストバスアダプタのインスタンス経由で参照できるすべてのチャンネル、ターゲット、LUN をクリーンアップします。

## アレイ側からの LUN の特性の変更

アレイで LUN の特性を変更できる場合があります。ほとんどの場合、デバイスに変更された性質が適用される前に、完全にデバイスの使用を停止する必要があります。LUN のプロパティを変更する前にデバイスをオフラインにし、設定後にデバイスを再びオンラインにすることをお勧めします。

EMC BCV や SRDF 操作など特定の場、処理中にデバイスをオンラインの状態に保つことができます。

クラスタの場合、クラスタ内のすべてのノードでこの手順を実行します。

#### LUN のプロパティを変更するには

- 1 デバイスでホストされているすべてのアプリケーションおよびボリュームを停止します。  
デバイスが Veritas Volume Manager (VxVM) によって使用中の場合は、次の手順を実行します。

DMP デバイス上で Linux LVM を使っている LUN の場合は、LVM ボリュームグループからデバイスを削除します。

```
# vgreduce vgnamedevicepath
```

- 2 LUN の特性を変更します。
- 3 デバイスをオンラインにします。

Veritas Volume Manager ディスクの場合:

DMP デバイス上で Linux LVM を使っている LUN の場合は、LVM ボリュームグループにデバイスを再び追加します。

```
# vgreduce vgnamedevicepath
```

- 4 DMP を使ってデバイススキャンを実行します。  
クラスタ内のすべてのノードでこの手順を実行します。

```
# vxdisk scandisks
```

## Dynamic Reconfiguration ツールでのホストバスアダプタのオンラインでの交換

DMP (Dynamic Multi-Pathing) には、既存のシステムからのホストバスアダプタの削除を簡素化する DR (Dynamic Reconfiguration) ツールがあります。

ホストバスアダプタをオンラインで交換するには

- 1 `vxdiskadm` ユーティリティを開始します。  

```
# vxdiskadm
```
- 2 `vxdiskadm` メニューから [Dynamic Reconfiguration 機能] オプションを選択します。
- 3 [HBA を追加 (Replace HBAs)] オプションを選択します。  
出力が DMP に利用可能な HBA の一覧を表示します。

- 4 交換する HBA を 1 つ以上選択します。
- 5 プロンプトで HBA の選択を確認します。
- 6 ホストバスアダプタを交換します。
- 7 置換処理を続行するために **Dynamic Reconfiguration** のツールに戻って [y] を選択します。

DMP がオペレーティングシステムのデバイスツリーを更新します。

## アレイコントローラファームウェアのオンラインでのアップグレード

ストレージアレイサブシステムには、修正、パッチ、機能のアップグレードとして、コードのアップグレードが必要です。ファイルシステムがマウントされ、I/O サービスがストレージに提供されている場合は、これらのアップグレードをオンラインで実行できます。

ストレージサブシステムには、冗長性用に複数のコントローラが搭載されています。オンラインアップグレードは一度に 1 つのコントローラで行われます。コントローラのいずれかでオンラインコントローラアップグレードを実行している間は、DMP (**Dynamic Multi-Pathing**) がすべての I/O を代替コントローラにフェールオーバーします。最初のコントローラが完全にコードを展開した後、新しいバージョンのコードを使って再ブート、リセットするとオンラインになります。他のコントローラは同じ処理を実行し、I/O を代替コントローラにフェールオーバーします。

---

**メモ:** この処理を通して、アプリケーションの I/O は影響を受けません。

---

アレイベンダーはこの処理にさまざまな名前を付けています。たとえば、EMC は **CLARiiON** アレイの無停止アップグレード (NDU) と呼んでいます。

A/A タイプのアレイでは、このオンラインアップグレード処理の間に特別な処理は必要ありません。A/P、A/PF、ALUA タイプのアレイでは、オンラインのコントローラコードアップグレード中に、ベンダ固有のアレイポリシーモジュール (APM) を通して、DMP がアレイ固有の処理を実行します。

コントローラがコードアップグレード時にリセットされ、再ブートされるときに、DMP は SCSI 状態を通してこの状態を検出します。DMP はすべての I/O をすぐに次のコントローラにフェールオーバーします。

アレイが完全に NDU をサポートしていない場合は、コントローラへのすべてのパスが短い間 I/O 用に使用できなくなる場合があります。アップグレードを始める前に、`dmp_lun_retry_timeout` チューニングパラメータを、I/O 用にコントローラを使用できなくなると予想される時間よりも長い時間に設定します。DMP では `dmp_lun_retry_timeout` で指定した時間が終わるまで、または I/O が成功するまで、

どちらかの状態になるまでは I/O はエラーになりません。したがって、アプリケーション I/O を中断しないでファームウェアのアップグレードを実行できます。

たとえば、I/O のためにパスを 300 秒間利用できないと予想している場合は、次のコマンドを使います。

```
# vxddmpadm settune dmp_lun_retry_timeout=300
```

DMP は I/O の再試行を 300 秒間、または I/O が成功するまで実行します。

オンラインコントローラアップグレードまたは NDU をサポートしているアレイドを確認するには、次の URL にあるハードウェア互換性リスト(HCL)を参照してください。

[https://www.veritas.com/support/en\\_US/article.000107677](https://www.veritas.com/support/en_US/article.000107677)

# イベント監視

この章では以下の項目について説明しています。

- [Dynamic Multi-Pathing \(DMP\) のイベントソースデーモン \(vxesd\) について](#)
- [ファブリック監視と予防的なエラー検出](#)
- [Dynamic Multi-Pathing \(DMP\) の iSCSI および SAN ファイバーチャネルトポロジーの検出](#)
- [DMP イベントログ](#)
- [Dynamic Multi-Pathing \(DMP\) のイベントソースデーモンの起動と停止](#)

## Dynamic Multi-Pathing (DMP) のイベントソースデーモン (vxesd) について

イベントソースデーモン (vxesd) は Dynamic Multi-Pathing (DMP) コンポーネントプロセスで、適切な処理を実行するために使われるデバイス関連イベントの通知を受信します。vxesd のメリットには次のものがあります。

- SAN ファブリックイベントの監視と予防的なエラー検出 (SAN イベント)  
p.123 の「[ファブリック監視と予防的なエラー検出](#)」を参照してください。
- トラブルシューティングのために DMP イベントをログに記録 (DMP イベント)  
p.124 の「[DMP イベントログ](#)」を参照してください。
- 自動デバイス検出 (OS イベント)
- SAN コンポーネントと HBA アレイポートの接続性の検出 (ファイバーチャネルと iSCSI)  
p.124 の「[Dynamic Multi-Pathing \(DMP\) の iSCSI および SAN ファイバーチャネルトポロジーの検出](#)」を参照してください。

p.125 の「[Dynamic Multi-Pathing \(DMP\) のイベントソースデーモンの起動と停止](#)」を参照してください。

## ファブリック監視と予防的なエラー検出

DMP は、パスのエラー検出で予防的な対応を取ります。

DMP イベントソースデーモン `vxesd` は、SNIA (Storage Networking Industry Association) HBA API ライブラリを使い、SAN ファブリックイベントを HBA から受信します。

DMP は、アクティブな I/O が無い場合でも、この情報に基づいて問題が疑われるデバイスを SAN イベントからチェックします。DMP が疑わしいデバイスを検証する間、新しい I/O は正常なパスに転送されます。

起動中、`vxesd` は (SNIA ライブラリ経由で) HBA をクエリーして SAN トポロジーを取得します。`vxesd` デーモンは、オペレーティングシステムが認識する個々のデバイスパスに対応する PWWN (Port World Wide Name) を確認します。`vxesd` デーモンがトポロジーを取得した後に、`vxesd` は SAN イベントの通知を受けるために HBA に登録します。LUN が SAN から切断されると、HBA は `vxesd` に SAN イベントを通知し、影響を受ける PWWN を指示します。`vxesd` デーモンはこのイベント情報を以前のトポロジー情報と関連付けて、どのデバイスパスの集合が影響を受けたかを特定します。

`vxesd` デーモンは、影響を受けたパスの集合を `vxconfigd` デーモン (DDL) に送信し、デバイスパスを疑わしい状態としてマーク付けできるようにします。

パスが疑わしい状態としてマーク付けされると、そのパスがデバイスへの最終パスである場合を除き、DMP は新しい I/O をそのパスに送信しません。バックグラウンドでは、DMP リストアタスクが SCSI 照会プローブを使って次の定期サイクルでパスのアクセシビリティを確認します。SCSI 照会に失敗した場合、DMP は影響を受けた LUN へのパスを無効にし、そのパスはイベントログにも記録されます。

後から LUN が再接続された場合、HBA は `vxesd` に SAN イベントを通知します。DMP リストアタスクが次のテストサイクルを実行する際に、無効にされたパスが SCSI プローブによって確認され、成功した場合はパスが再び有効化されます。

---

**メモ:** `vxesd` が HBA LINK UP イベントを受信すると、DMP リストアタスクが再開され、次の定期サイクルを待たずに SCSI プローブがすぐに実行されます。DMP リストアタスクが再開されると、このタスクによって新しい定期サイクルが開始されます。無効化されたパスが最初の SCSI プローブの時点でアクセス不能な場合、次のサイクル (デフォルトの周期は 300 秒) でそのパスが再度テストされます。

---

ファブリック監視機能はデフォルトで有効です。`dmp_monitor_fabric` チューニングパラメータの値は、再起動の前後で変化しません。

`dmp_monitor_fabric` チューニングパラメータの現在値を表示するには、次のコマンドを使います。

```
# vxddmpadm gettune dmp_monitor_fabric
```

ファブリック監視機能を無効にするには、次のコマンドを使います。

```
# vxddmpadm settune dmp_monitor_fabric=off
```

ファブリック監視機能を有効にするには、次のコマンドを使います。

```
# vxddmpadm settune dmp_monitor_fabric=on
```

## Dynamic Multi-Pathing (DMP) の iSCSI および SAN ファイバーチャネルトポロジーの検出

`vxesd` は、ホストが認識する iSCSI および FC (Fibre Channel の略でファイバーチャネルデバイスの意味) のトポロジーを構築します。`vxesd` デーモンは、SNIA Fibre Channel HBA API を使って SAN トポロジーを取得します。IMA が利用できない場合、iSCSI SAN トポロジーを取得するために iSCSI 管理 CLI が使われます。

ファイバーチャネルおよび iSCSI デバイスの階層リストを表示するには、次のコマンドを使います。

```
# vxddladm list
```

`vxddladm(1M)` のマニュアルページを参照してください。

## DMP イベントログ

イベントソースデーモン (`vxesd`) は Dynamic Multi-Pathing (DMP) コンポーネントプロセスで、適切な処理を実行するために使われるデバイス関連イベントの通知を受信します。

DMP はメジャーイベントを `vxesd` に通知し、`vxesd` はイベントをログファイルに記録します。記録されるイベントには次のものがあります。

- パスまたは `dmnode` を有効としてマーク付け
- パスまたは `dmnode` を無効としてマーク付け
- パスの調整
- I/O エラー解析
- HBA および SAN イベント

DMP イベントについてシステムログまたはコンソールログに表示される詳細レベルを変更できます。チューニングパラメータ `dmp_log_level` を使用します。有効な値は 1 から 9 までです。デフォルトレベルは 1 です。

```
# vxddladm settune dmp_log_level=X
```

`dmp_log_level` の現在の値は次のようにして表示できます。

```
# vxddladm gettune dmp_log_level
```

各種のログレベルについて詳しくは、`vxddladm(1M)` のマニュアルページを参照してください。

## Dynamic Multi-Pathing (DMP) のイベントソースデーモンの起動と停止

デフォルトでは、DMP (Dynamic Multi-Pathing) はブート時にイベントソースデーモン `vxesd` を起動します。

`vxesd` デーモンを停止するには、次のように `vxddladm` ユーティリティを使います。

```
# vxddladm stop eventsource
```

`vxesd` デーモンを起動するには、次のように `vxddladm` ユーティリティを使います。

```
# vxddladm start eventsource [logfile=logfilename]
```

`vxesd` デーモンの状態を表示するには、`vxddladm` ユーティリティを使います。

```
# vxddladm status eventsource
```

# 処理効率の監視とチューニング

この章では以下の項目について説明しています。

- [テンプレートを使った DMP \(Dynamic Multi-Pathing\) のチューニングについて](#)
- [DMP チューニングテンプレート](#)
- [DMP チューニングテンプレートの例](#)
- [設定属性テンプレートを使った DMP のホストのチューニング](#)
- [DMP 設定ファイルの管理](#)
- [DMP のチューニングパラメータと属性のデフォルト値へのリセット](#)
- [テンプレートでサポートされる DMP のチューニングパラメータと属性](#)
- [DMP チューニングパラメータ](#)

## テンプレートを使った DMP (Dynamic Multi-Pathing) のチューニングについて

Dynamic Multi-Pathing には、処理効率を最適化するために設定できる複数のチューニングパラメータや属性があります。DMP は、1 回の操作で複数のチューニングパラメータと属性を更新できるテンプレートメソッドを提供します。テンプレートは DMP 設定の全部か一部を表し、ホストのパラメータと属性の値を示します。

チューニングパラメータの表示や操作を行うには、ファイルに DMP チューニングパラメータの設定値をダンプできます。必要に応じて、パラメータと属性を編集します。その後、ホストにテンプレートファイルをロードして、1 回の操作ですべての値を更新します。

設定ファイルは、同じホストにロードすることも、別の同様のホストにロードすることもできます。テンプレートメソッドは、次のシナリオに有用です。

- 処理効率を最適化するチューニング値を使って、複数の同様のホストを設定する。  
1つのホストを設定して、処理効率を最適化します。ホストの設定後、テンプレートファイルにチューニングパラメータと属性をダンプします。同様の要件を持つ別のホストに、そのテンプレートファイルをロードできます。ベリタスは同じ設定テンプレートを使うホストは、オペレーティングシステムが同じで、I/O 要件も同様であることを推奨します。
- 複数の特殊なテンプレートを定義して、異なる I/O 負荷要件を処理する。  
ホストの負荷が変更されたときに、異なるテンプレートをロードして処理効率の最適化を図ることができます。この戦略は、予測可能な一時的な I/O 負荷の変更に対応する場合に適切です。システム管理者は、システムの I/O 負荷の動作を定義した後で、特定の負荷に合わせてチューニングテンプレートをカスタマイズできます。スクリプトか cron ジョブで使うことができる単一のロードコマンドがあるため、チューニングを自動化できます。

いつでも、設定はリセットできます。その場合、チューニングパラメータと属性の値は、DMP のデフォルト値に復帰します。

`vxddmpadm config` コマンドを使って、DMP 設定ファイルを管理できます。

`vxddmpadm (1M)` マニュアルページを参照してください。

## DMP チューニングテンプレート

テンプレートの機構は、ファイルまたは標準出力に設定値をダンプすることで、DMP のパラメータと属性のチューニングを可能にします。

DMP では、次のタイプの情報について、テンプレートファイルを使ったチューニングがサポートされます。

- DMP チューニングパラメータ。
- エンクロージャ、アレイ名、またはアレイタイプに定義する DMP 属性。
- Veritas の名前前の付け方のパラメータ。

テンプレートファイルは、次のようなセクションに分かれています。

DMP Tunables	すべてのエンクロージャとアレイに適用されます。
Namingscheme	すべてのエンクロージャとアレイに適用されます。
Arraytype	アレイタイプのカスタマイズに使用します。指定したアレイタイプのすべてのエンクロージャに適用されます。

Arrayname	特定のアレイでカスタマイズが必要な場合、つまり、チューニングパラメータがアレイタイプに適用されるものと異なる場合に使用します。  このセクションの属性は、指定したアレイ名のすべてのエンクロージャ適用されます。
Enclosurename	指定した Cab シリアル番号とアレイ名のエンクロージャに適用されます。  特定のエンクロージャでカスタマイズが必要な場合、つまり、チューニングパラメータがアレイタイプとアレイ名に適用されるものと異なる場合に使用します。

ロードはセクション単位で実行されます。DMP は、各セクションにつき、すべての属性が有効な場合のみ、そのセクションをロードします。すべてのセクションが処理されたら、DMP はエラーと警告の一覧をユーザーに報告します。DMP は部分的なロールバックをサポートしません。DMP はロード処理中にチューニングパラメータと属性を検証します。ただし、Veritas はファイルのロード前に、設定テンプレートファイルを調べることを推奨します。設定ファイルが正しいことが確認されるまで、必要に応じて修正してください。

テンプレートにロード時に、各属性は次の順序の優先度が割り当てられます。

エンクロージャセクション > アレイ名セクション > アレイタイプセクション

同じアレイタイプのすべてのエンクロージャに同じ設定が必要な場合は、テンプレートから対応するアレイ名セクションとエンクロージャ名セクションを削除します。アレイタイプセクションの設定のみを定義します。一部のエンクロージャがアレイ名で設定をカスタマイズする必要がある場合は、アレイ名またはエンクロージャの属性セクションを保持します。アレイタイプに定義した設定と同じ設定を使う場合は、エンクロージャまたはアレイ名のエントリを削除できます。

ホストから設定ファイルをダンプするときに、そのホストには他のホストでは表示されないアレイが含まれていることがあります。そのようなエンクロージャ、アレイタイプ、またはアレイ名を含まないチューニング先のホストにテンプレートをロードすると、DMP は該当セクションを無視します。

チューニング先ホストの非共有のアレイやホスト固有のアレイに設定を適用しない場合は、該当する各アレイのエンクロージャセクションをテンプレートに必ず定義してください。チューニング先のホストにテンプレートファイルをロードするときに、エンクロージャセクションによって設定が決めます。定義しないと、DMP は各アレイ名またはアレイタイプのセクションの設定を適用します。

## DMP チューニングテンプレートの例

このセクションでは、DMP チューニングテンプレートの例を示します。

## DMP Tunables

```
dmp_cache_open=on
dmp_daemon_count=10
dmp_delayq_interval=15
dmp_restore_state=enabled
dmp_fast_recovery=on
dmp_health_time=60
dmp_log_level=1
dmp_low_impact_probe=on
dmp_lun_retry_timeout=30
dmp_path_age=300
dmp_pathswitch_blks_shift=9
dmp_probe_idle_lun=on
dmp_probe_threshold=5
dmp_restore_cycles=10
dmp_restore_interval=300
dmp_restore_policy=check_disabled
dmp_retry_count=5
dmp_scsi_timeout=20
dmp_sfg_threshold=1
dmp_stat_interval=1
dmp_monitor_ownership=on
dmp_monitor_fabric=on
dmp_native_support=off
```

## Namingscheme

```
namingscheme=ebn
persistence=yes
lowercase=yes
use_avid=yes
```

## Arraytype

```
arraytype=CLR-A/PF
iopolicy=minimumq
partitionsizesize=512
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
```

## Arraytype

```
arraytype=ALUA
iopolicy=adaptive
partitionsizesize=512
use_all_paths=no
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
```

```
        redundancy=0
Arraytype
    arraytype=Disk
    iopolicy=minimumq
    partitionsize=512
    recoveryoption=nothrottle
    recoveryoption=timebound iotimeout=300
    redundancy=0
Arrayname
    arrayname=EMC_CLARiON
    iopolicy=minimumq
    partitionsize=512
    recoveryoption=nothrottle
    recoveryoption=timebound iotimeout=300
    redundancy=0
Arrayname
    arrayname=EVA4K6K
    iopolicy=adaptive
    partitionsize=512
    use_all_paths=no
    recoveryoption=nothrottle
    recoveryoption=timebound iotimeout=300
    redundancy=0
Arrayname
    arrayname=Disk
    iopolicy=minimumq
    partitionsize=512
    recoveryoption=nothrottle
    recoveryoption=timebound iotimeout=300
    redundancy=0
Enclosure
    serial=CK200051900278
    arrayname=EMC_CLARiON
    arraytype=CLR-A/PF
    iopolicy=minimumq
    partitionsize=512
    recoveryoption=nothrottle
    recoveryoption=timebound iotimeout=300
    redundancy=0
    dmp_lun_retry_timeout=30
Enclosure
    serial=50001FE1500A8F00
    arrayname=EVA4K6K
```

```
arraytype=ALUA
iopolicy=adaptive
partitionsizesize=512
use_all_paths=no
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
dmp_lun_retry_timeout=30

Enclosure
serial=50001FE1500BB690
arrayname=EVA4K6K
arraytype=ALUA
iopolicy=adaptive
partitionsizesize=512
use_all_paths=no
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
dmp_lun_retry_timeout=30

Enclosure
serial=DISKS
arrayname=Disk
arraytype=Disk
iopolicy=minimumq
partitionsizesize=512
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
dmp_lun_retry_timeout=30
```

## 設定属性テンプレートを使った DMP のホストのチューニング

テンプレートファイルを使って、同じホストまたは別の同様のホストに DMP 設定の一連の変更をアップロードできます。

Veritas はチューニングパラメータ値の設定もとのホストと同様のホストに DMP のテンプレートをロードすることを推奨します。

## テンプレートを使ってホストの DMP を設定するには

- 1 ファイルに現在のホストの設定内容をダンプします。

```
# vxddmpadm config dump file=filename
```

- 2 ファイルを編集し、テンプレートのチューニングパラメータを必要に応じて変更します。

チューニング先のホストには、非共有のアレイかホスト固有のアレイが含まれることがあります。このようなアレイでアレイ名またはアレイタイプの設定が更新されるのを回避するには、テンプレートで各アレイのエンクロージャセクションを定義します。チューニング先のホストにテンプレートファイルをロードするときに、エンクロージャセクションによって設定が決められます。定義しないと、DMP は各アレイ名またはアレイタイプのセクションの設定を適用します。

- 3 DMP チューニングパラメータの値を検証します。

```
# vxddmpadm config check file=filename
```

DMP は設定の検査が成功なら出力を表示しません。ファイルにエラーがある場合は、DMP はエラーを表示します。設定ファイルが有効になるまで、必要に応じて修正してください。たとえば、次のようなエラーが表示されることがあります。

```
VxVM vxddmpadm ERROR V-5-1-0 Template file 'error.file' contains following errors:
```

```
Line No: 22  'dmp_daemon_count' can not be set to 0 or less
Line No: 44  Specified value for 'dmp_health_time' contains
non-digits
Line No: 64  Specified value for 'dmp_path_age' is beyond
the limit of its value
Line No: 76  'dmp_probe_idle_lun' can be set to either on or off
Line No: 281 Unknown arraytype
```

- 4 チューニング先のホストにファイルをロードします。

```
# vxddmpadm config load file=filename
```

ロード処理中に、DMP はテンプレートの各セクションを検証します。DMP はすべての有効なセクションをロードします。DMP はエラーがあるセクションはロードしません。

## DMP 設定ファイルの管理

ホストに前回ロードしたテンプレートファイルの名前を表示できます。この情報には、DMP がテンプレートファイルをロードした日時が含まれます。

ホストが現在使っているテンプレートファイルの名前を表示するには

◆ # vxdmppadm config show

```
TEMPLATE_FILE      DATE                TIME
=====
/tmp/myconfig      Feb 09, 2011      11:28:59
```

## DMP のチューニングパラメータと属性のデフォルト値へのリセット

DMP は DMP のチューニングパラメータと属性のデフォルト値を維持します。いつでも、ホストにデフォルト値を復元できます。テンプレートファイルを使ってホストに適用した変更は、破棄されます。

**DMP チューニングパラメータをデフォルト値にリセットするには**

◆ 次のコマンドを実行します。

```
# vxdmppadm config reset
```

## テンプレートでサポートされる DMP のチューニングパラメータと属性

DMP では、次のチューニングパラメータと属性について、設定テンプレートを使ったチューニングがサポートされます。

DMP チューニングパラメータ

p.134 の「[DMP チューニングパラメータ](#)」を参照してください。

エンクロージャ、アレイ名、またはアレイタイプに定義する DMP 属性。

- iopolicy
- partitionsize
- use\_all\_paths
- recoveryoption 属性 (retrycount または iotimeout)
- redundancy
- dmp\_lun\_retry\_timeout

- 名前の付け方の属性
- naming scheme
  - persistence
  - lowercase
  - use\_avid

次のチューニングパラメータは、テンプレートでサポートされません。

- OS のチューニングパラメータ
- TPD モード
- エンクロージャのフェールオーバー属性 (failovermode)

## DMP チューニングパラメータ

DMP は環境をチューニングするのに使うことができる各種パラメータを提供します。

表 7-1 に、チューニングできる DMP パラメータを示します。チューニングパラメータはオンラインで設定できます。再ブートは不要です。

表 7-1 チューニング可能な DMP パラメータ

パラメータ	説明
dmp_cache_open	このパラメータを on に設定した場合、デバイスの最初の起動がキャッシュに保存されます。このキャッシュ処理により、デバイスの以降の起動によって発生するオーバーヘッドが最小化されるため、デバイス検出のパフォーマンスが高まります。このパラメータを off に設定した場合、キャッシュへの保存は行われません。  デフォルト値は on です。
dmp_daemon_count	サービスパスのエラー処理、パスリストア、その他の DMP 管理作業に使用できるカーネルスレッド数。  スレッドのデフォルト数は 10 です。
dmp_delayq_interval	アレイがスタンバイパスにフェールオーバーした後、再試行 I/O までの DMP の待機時間。一部のディスクアレイは、フェールオーバー直後の I/O 要求を受け入れることができません。  デフォルト値は 15 秒です。

パラメータ	説明
dmp_display_alua_states	<p>ALUA アレイの場合、このチューニングパラメータは PATH-TYPE[M] 列の PRIMARY または SECONDARY 状態ではなく、非対称アクセスの状態を表示します。</p> <p>非対称アクセスの状態は次のとおりです。</p> <ul style="list-style-type: none"> <li>■ アクティブ/最適化 (Active/Optimized)</li> <li>■ アクティブ/非最適化 (Active/Non-optimized)</li> <li>■ スタンバイ (Standby)</li> <li>■ 利用不能 (Unavailable)</li> <li>■ 進行中の移行 (TransitionInProgress)</li> <li>■ オフライン (Offline)</li> </ul> <p>デフォルトのチューニングパラメータの値は on です。</p>
dmp_fast_recovery	<p>DMP が、HBA インターフェースから直接 SCSI エラー情報を取得するかどうかを示します。HBA インターフェースがエラー照会機能をサポートしている場合、値を on に設定するとエラーリカバリが潜在的に速くなります。このパラメータを off に設定した場合、HBA インターフェースは使われません。</p> <p>デフォルト設定は on です。</p>
dmp_health_time	<p>DMP は、断続的にエラーになっているパスを検出し、I/O 要求がこれらのパスに送信されないようにします。</p> <p>dmp_health_time の値は、パスが健全であり続けなければならない時間を秒で表します。この時間内にパスの状態が enabled から disabled に戻った場合、DMP はパスに断続的なエラーが発生していると設定し、dmp_path_age に設定されている秒数が経過するまでパスを I/O に対して有効に再設定しません。</p> <p>デフォルト値は 60 秒です。</p> <p>値を 0 に設定すると、DMP は断続的にエラーになっているパスを検出しません。</p>

パラメータ	説明
dmp_log_level	<p>DMP コンソールメッセージで表示される詳細のレベル。次のレベル値が定義されています。</p> <p>1 - 重要な DMP ログメッセージをすべて表示します。</p> <p>2 - レベル 1 のメッセージに加えて、パスまたはディスクの追加または削除、SCSI エラー、I/O エラー、DMP ノードの移行に関するメッセージを表示します。</p> <p>3 - レベル 1 と 2 のメッセージに加えて、パスの調整、障害の可能性のあるパス、アイドル状態のパス、異常なパスのロジックに関するメッセージを表示します。</p> <p>4 - レベル 1、2、3 のメッセージに加えて、パスの属性の設定または変更に関するメッセージと、チューニングパラメータに関連する変更を表示します。</p> <p>5 以上 - レベル 1、2、3、4 のメッセージに加えて、他の詳しいメッセージを表示します。</p> <p>デフォルト値は 1 です。</p>
dmp_low_impact_probe	<p>リストアデーモンによるパスのプロープが最適化されているかどうかを決定します。最適化は on に設定すると有効になり、off に設定すると無効になります。パスのプロープは、リストアポリシーが check_disabled である場合、または check_periodic ポリシーの check_disabled フェーズの間のみ最適化されます。</p> <p>デフォルト値は on です。</p>

パラメータ	説明
dmp_lun_retry_timeout	<p><b>HBA と SCSI ドライバ</b>によって処理されない一時的なエラーを処理するための再試行間隔を指定します。</p> <p>時間は秒数で指定します。</p> <p>通常は、このような特殊な処理は不要です。そのため、<code>dmp_lun_retry_timeout</code> チューニングパラメータのデフォルト値は <b>30</b> です。ディスクのすべてのパスがエラーになった場合、<b>DMP</b> はアプリケーションの <b>I/O</b> をエラーにします。パスでは、接続性が 1 回のみチェックされます。</p> <p><b>DMP</b> による一時的なエラー処理が必要になる特殊な場合では、<b>DMP</b> がアプリケーション <b>I/O</b> をエラーにする処理を短期間延期するように設定します。この期間を指定するには、<code>dmp_lun_retry_timeout</code> チューニングパラメータを <b>0</b> 以外の値に設定します。<b>LUN</b> のすべてのパスがエラーになっても、<b>I/O</b> を提供する必要がある場合、<b>DMP</b> は指定した期間中、<b>5</b> 秒ごとにパスをプローブします。この期間内にパスが復元された場合、<b>DMP</b> はそのことを検出して <b>I/O</b> を再試行します。どちらが先でも、指定した <code>dmp_lun_retry_timeout</code> が経過するか、またはパスの <b>1</b> つで <b>I/O</b> が正常に処理されないかぎり、<b>DMP</b> はエラーが発生したすべてのパスでディスクに <b>I/O</b> を提供する処理をエラーにしません。</p>
dmp_monitor_fabric	<p><b>DMP</b> が <b>SNIA HAB API</b> から <b>HBA</b> イベントに登録するかどうかを指定します。これらのイベントは障害が切迫している <b>I/O</b> パスを避けることによってプロアクティブにフェールオーバーパフォーマンスを改善します。</p> <p>この <b>DDL</b> 機能をサポートするようにパッチがあてられたリリース <b>5.0</b> 以前の場合、デフォルト設定は <code>off</code> です。<b>5.0</b> 以降のリリースでは、デフォルト設定は <code>on</code> です。</p>
dmp_monitor_ownership	<p><b>ALUA</b> のアレイの所有権の監視を有効にするかどうかを決定します。このチューニングパラメータを <code>on</code> に設定すると、<b>DMP</b> はデバイスをポーリングして <b>LUN</b> 所有権の変更を確認します。ポーリング間隔は <code>dmp_restore_interval</code> チューニングパラメータで指定します。デフォルト値は <code>on</code> です。</p> <p><code>dmp_monitor_ownership</code> チューニングパラメータが <code>off</code> のとき、<b>DMP</b> は <b>LUN</b> 所有権の変更を確認するポーリングを行いません。</p>

パラメータ	説明
dmp_native_support	<p>DMP がネイティブデバイスのマルチパスを行うかどうかを決定します。</p> <p>DMP がネイティブデバイスをマルチパス化するには、チューニングパラメータを on に設定します。</p> <p>Dynamic Multi-Pathing が別の Veritas InfoScale 製品のコンポーネントとしてインストールされる場合、デフォルト値は off です。</p> <p>Dynamic Multi-Pathing がスタンドアロンの製品としてインストールされる場合、デフォルト値は on です。</p>
dmp_path_age	<p>断続的にエラーの発生しているパスが継続して健全と判断されなければならない期間を示します。この期間が経過すると、DMP は再度そのパスに I/O 要求のスケジュール設定を試みます。</p> <p>デフォルト値は 300 秒です。</p> <p>値を 0 に設定すると、DMP は断続的にエラーになっているパスを検出しません。</p>
dmp_pathswitch_blks_shift	<p>次に使用可能なパスに切り替わる前に DMP パスでアレイに送信される、連続 I/O ブロックのデフォルト数を示します。この値は 2 の累乗の指数(整数)で指定するようになっており、たとえば 9 は 512 ブロックを表します。</p> <p>デフォルト値は 9 です。この場合、512 ブロック(256 KB)の連続 I/O が切り替え前に DMP パスで送信されます。内部データキャッシュを持つ高機能ディスクアレイの場合、このチューニングパラメータの値を大きくすることで、スループットが向上する可能性があります。たとえば、日立 SANRISE2800 アクティブ/アクティブアレイの場合、シーケンシャルな読み取りまたは書き込みを主として構成される I/O 処理パターンに最適な値は 15 - 17 です。</p> <p>このパラメータの影響を受けるのは、I/O ポリシー balanced 設定時の動作のみです。パラメータ値を 0 にすると、vxddm コマンドでアレイに対し別のパーティションサイズを指定した場合を除き、このポリシーではマルチパス化が無効になります。</p> <p>p.68 の「I/O ポリシーの指定」を参照してください。</p>

パラメータ	説明
dmp_probe_idle_lun	<p>DMP 統計情報の収集が有効になっている場合、DMP パスリストアスレッドがアイドル状態の LUN をプローブするように、このチューニングパラメータを on (デフォルト) に設定します。機能をオフにするには、このチューニングパラメータを off に設定します (アイドル状態の LUN とは、I/O 要求がスケジュール設定されていない VM ディスクを指します。) このチューニングパラメータの値は、DMP 統計の収集が有効になっているときのみ解釈されます。統計の収集をオフにすると、アイドル状態の LUN のプローブも無効になります。</p> <p>デフォルト値は on です。</p>
dmp_probe_threshold	<p>dmp_low_impact_probe が on に設定されている場合、dmp_probe_threshold は同じサブパスフェールオーバーグループに属する他のパスの状態を変更する前に、プローブするパスの数を決定します。</p> <p>デフォルト値は 5 です。</p>
dmp_restore_cycles	<p>DMP リストアポリシーが check_periodic の場合に、check_all ポリシーが呼び出されるまでのサイクル数です。</p> <p>デフォルト値は 10 です。</p> <p>p.81 の「<a href="#">DMP パスリストアポリシーの設定</a>」を参照してください。</p>
dmp_restore_interval	<p>interval 属性値には、パスリストアスレッドがパスを調べる頻度を指定します。時間は秒数で指定します。</p> <p>デフォルト値は 300 です。</p> <p>このチューニングパラメータの値は、vxsdmpadm start restore コマンドを使っても設定できます。</p> <p>p.81 の「<a href="#">DMP パスリストアポリシーの設定</a>」を参照してください。</p>

パラメータ	説明
dmp_restore_policy	<p>次のいずれかの値に設定可能な DMP リストアポリシー。</p> <ul style="list-style-type: none"> <li>■ check_all</li> <li>■ check_alterdate</li> <li>■ check_disabled</li> <li>■ check_periodic</li> </ul> <p>デフォルト値は check_disabled です。</p> <p>このチューニングパラメータの値は、vxdmpadm start restore コマンドを使っても設定できます。</p> <p>p.81 の「<a href="#">DMP パスリストアポリシーの設定</a>」を参照してください。</p>
dmp_restore_state	<p>このパラメータを enabled に設定している場合は、パスリストアスレッドの開始が有効になります。</p> <p>p.81 の「<a href="#">DMP パスリストアポリシーの設定</a>」を参照してください。</p> <p>このパラメータを disabled に設定した場合、パスリストアスレッドは停止し、無効になります。</p> <p>このパラメータを stopped に設定した場合、次のデバイスの検出サイクルまでパスリストアスレッドは停止します。</p> <p>デフォルトは enabled です。</p> <p>p.83 の「<a href="#">DMP パスリストアスレッドの停止</a>」を参照してください。</p>
dmp_scsi_timeout	<p>DMP 経由で送信される SCSI コマンドにタイムアウト値を設定する必要があるかを判別します。タイムアウト時間内にデバイスに送信されたことを示す SCSI コマンドの応答を HBA が受信しない場合、SCSI コマンドは障害エラーコードとともに返されます。</p> <p>デフォルト値は 20 秒です。</p>
dmp_sfg_threshold	<p>DMP が同じフェールオーバーグループに属する他のパスの検討を開始する前に、フェールオーバーグループ内で障害が発生する必要があるパスの最小数を決定します。値が 0 の場合、サブバスフェールオーバーグループに基づくフェールオーバーのロジックが無効になります。</p> <p>デフォルト値は 1 です。</p>
dmp_stat_interval	<p>DMP 統計情報を収集する間隔。</p> <p>最小値は 1 秒で、これがデフォルトです。</p>

# DMP のトラブルシューティング

この付録では以下の項目について説明しています。

- [DMP からパスを除外または DMP にパスを追加するときのエラーのリカバリ](#)
- [アレイサポートのダウングレード](#)

## DMP からパスを除外または DMP にパスを追加するときのエラーのリカバリ

`vxddmpadm exclude` コマンドを使って DMP からパスを除外できます。以前に除外したパスを、`vxddmpadm include` コマンドを使って DMP 制御に戻すことができます。これらのコマンドは除外済みパスを `vxvm.exclude` ファイルに格納します。`vxvm.exclude` ファイルが壊れている場合は、パスの追加と除外の操作を正常に完了できません。

`vxvm.exclude` ファイルが壊れている場合は次のエラーが表示されます。

```
# vxddmpadm exclude ctrlr=c0
VxVM vxddmpadm ERROR V-5-1-3996 File not in correct format
```

DMP は `vxvm.exclude.corrupt` という名前で壊れたファイルを保存します。DMP は新しい `vxvm.exclude` ファイルを作成します。この状況から手動でリカバリする必要があります。

壊れた除外ファイルのリカバリするには

- 1 エラーが表示された `vxddmpadm include` コマンドまたは `vxddmpadm exclude` コマンドを再び実行します。

```
# vxddmpadm exclude ctrl=c0
```

- 2 関連する除外パスのエントリを見つけるには、保存されている `vxvm.exclude.corrupt` ファイルを表示します。

```
# cat /etc/vx/vxvm.exclude.corrupt
exclude_all 0
paths
controllers
c4 /pci@1f,4000/pci@4/scsi@4/fp@0,0
```

- 3 手順 2 で見つけたパスに `vxddmpadm exclude` コマンドを再び実行します。

```
# vxddmpadm exclude ctrl=c4
```

- 4 除外パスが `vxvm.exclude` ファイルにあることを確認します。

```
# cat /etc/vx/vxvm.exclude

exclude_all 0
paths
#
controllers
c0 /pci@1f,4000/scsi@3
c4 /pci@1f,4000/pci@4/scsi@4/fp@0,0
#
product
#

# cat vxvm.exclude

exclude_all 0
paths
#
controllers
c0 c0
c4 c4
#
product
#
```

## アレイサポートのダウングレード

アレイサポートは `VRTSaslapm` という単一 rpm に含まれています。この rpm には、**Array Support Libraries (ASL)** と **Array Policy Modules (APM)** が含まれています。**Dynamic Multi-Pathing** の各メジャーリリースには、サポート対象の `VRTSaslapm rpm` が含まれます。これは、製品インストールの一環としてインストールされます。次のメジャーリリースが出るまでに、`VRTSaslapm rpm` に対する更新を通して、ベリタスから追加のアレイサポートが提供される場合があります。

更新済みの `VRTSaslapm rpm` に何らかの問題があった場合、ベリタスは、**ASL/APM rpm** の前のバージョンにダウングレードするよう推奨することがあります。戻すことができる対象は、インストール済みの **Dynamic Multi-Pathing** のリリースについてサポートされている rpm のみです。システムがオンラインになっている間にダウングレードを実行する場合は、インストールされている rpm を削除しないでください。代わりに、新しい rpm に前のバージョンの rpm を上書きインストールできます。この方法では、`VRTSaslapm rpm` の複数のインスタンスがインストールされることはありません。

次の方法で `VRTSaslapm rpm` をダウングレードします。

オンライン時に **ASL/APM rpm** をダウングレードするには

- ◆ 前のバージョンの `VRTSaslapm rpm` を次のコマンドに指定します。

```
# rpm --force -Uvh VRTSaslapm
```

# 参照

この付録では以下の項目について説明しています。

- [Veritas コマンドに対するコマンド入力補完機能](#)

## Veritas コマンドに対するコマンド入力補完機能

Dynamic Multi-Pathing では、Dynamic Multi-Pathing (DMP) コマンドに対するコマンド入力補完機能がサポートされています。

このリリースでは、コマンド入力補完機能は `bash` シェルでのみサポートされます。シェルは、`bash` バージョン 2.4 以降である必要があります。

この機能を使うには、サポート対象の VxVM コマンドまたは DMP コマンドの入力時に **Tab** キーを押します。コマンドは可能なかぎり補完されます。選択肢があるときに、コマンド入力補完機能によって、コマンドに関する次の有効なオプションが表示されます。表示された値のうちの 1 つを入力します。カッコに囲まれた値は、ユーザー指定の値を示します。

---

**メモ:** プラットフォーム固有のオプションは、コマンド入力補完機能ではサポートされません。

---

デフォルトでは、ログインのたびに `bash` シェルを呼び出すことで、コマンド入力補完機能を使うことができます。コマンド入力補完機能を永続的に有効にするには、次のコマンドを使用します。

```
# vxdctl cmdcompletion enable
```

コマンド入力補完機能を有効にすると、`.bash_profile` ファイルが作成されます (存在しない場合)。

コマンド入力補完機能を永続的に無効にするには、次のコマンドを使用します。

```
# vxdctl cmdcompletion disable
```

vxdtcl(1M) マニュアルページを参照してください。

コマンド入力補完機能をサポートするコマンドは、次のとおりです。

- vxddladm
- vxdisk
- vxdlpadm

# 用語集

<b>bootdg</b>	ブートディスクグループのエイリアスとして予約済みのディスクグループ名。
<b>DCO ボリューム</b>	永続 <b>FastResync</b> 変更マップと <b>DRL</b> を保持するために使う特別なボリューム。「 <b>dirty region logging</b> 」を参照。
<b>DCO(データ変更オブジェクト)</b>	DCO ボリュームの <b>FastResync</b> マップについての情報を管理するために使う <b>VxVM</b> オブジェクト。ボリューム上で永続 <b>FastResync</b> を実行するには、 <b>DCO</b> オブジェクトと <b>DCO</b> ボリュームの両方をそのボリュームに関連付けておく必要があります。
<b>dirty region logging</b>	<b>VxVM</b> がプレックスへの変更を監視し、変更された領域をビットマップとしてログに記録する方法。新しいスタイルの <b>DCO</b> ボリュームがボリュームに関連付けられている場合、 <b>DRL (dirty region logging)</b> は <b>DCO</b> ボリュームに保存されます。それ以外の場合、 <b>DRL</b> はログサブディスクと呼ばれる関連付けられたサブディスクに保存されます。
<b>FastResync</b>	高速再同期化機能。 <b>STALE</b> 状態のミラーを迅速かつ効率的に再同期化し、スナップショット機構の効率性を向上させるために使います。
<b>Fibre Channel</b>	ストレージエリアネットワーク ( <b>SAN</b> ) のセットアップに一般的に使われる光ファイバ技術の総称。
<b>hostid</b>	<b>VxVM</b> でホストを識別するための文字列。ホストのホスト ID は <b>volboot</b> に保存され、ディスクとディスクグループの所有者を定義するときに使われます。
<b>JBOD (Just a Bunch Of Disks)</b>	高機能ではないディスクアレイの一般名。ディスクのホットスワップをサポートする場合としない場合があります。
<b>RAID (Redundant Array of Independent Disks)</b>	組み合わせられたストレージ容量の一部を使って、このアレイに格納されたデータに関する複製情報を保存するディスクアレイです。このディスクアレイにより、ディスクに障害が発生してもデータを再生成することができます。
<b>SAN (storage area network)</b>	コンピュータ、ディスクストレージおよびスイッチ、ハブ、ブリッジなどの相互接続ハードウェアのサブセット間の接続性を簡単に再設定できるネットワークパラダイム。
<b>VM ディスク (VM disk)</b>	<b>VxVM</b> の制御下にあり、ディスクグループに割り当てられているディスク。 <b>VM</b> ディスクは、 <b>VxVM</b> ディスクと呼ぶこともあります。
<b>volboot ファイル</b>	ブートディスクグループ設定情報のコピーを検索するために使う小さなファイル。このファイルには、標準位置に設定コピーを保存したディスクをリストすることも、設定コピーの位置への直接ポインタを保存することもできます。 <b>volboot</b> ファイルの保存場所はシステムによって異なります。
<b>vxconfigd</b>	<b>VxVM</b> 設定デーモン。 <b>VxVM</b> の設定の変更を行います。 <b>VxVM</b> の操作の前に、このデーモンを起動する必要があります。

アクティブ/アクティブディスクアレイ	マルチパス化されたディスクアレイのタイプの 1 つ。処理効率を低下させることなく、ディスクへのすべてのパスを介してディスクアレイ内のディスクへ同時にアクセスすることができます。
アクティブ/パッシブディスクアレイ	マルチパス化されたディスクアレイのタイプの 1 つ。1 つのディスクに 1 つのプライマリパスを指定でき、ディスクへのアクセスに随時使えます。指定されたアクティブパス以外のパスを使うと、一部のディスクアレイでは処理効率が大幅に低下することがあります。
エンクロージャ	「ディスクエンクロージャ」を参照。
エンクロージャに基づく名前の付け方	「デバイス名」を参照。
オブジェクト	VxVM に対して定義され、VxVM 内部で認識されるエンティティ。VxVM オブジェクトには、ボリューム、プレックス、サブディスク、ディスク、ディスクグループがあります。ディスクオブジェクトは、実際には、それぞれディスクの物理的側面と論理的側面を表す 2 種類のオブジェクトに分けられます。
カプセル化	指定されたディスク上の既存のパーティションをボリュームに変換するプロセス。いずれかのパーティションにファイルシステムが含まれる場合、/etc/fstab エントリが修正され、代わりにファイルシステムがボリューム上にマウントされます。
カラム	ストライプ化したプレックス内の 1 つまたは複数のサブディスクセット。ストライプ化は、プレックス内のカラムに交互に均等にデータを割り当てることにより実現されます。
クラスタ	ひとまとまりのディスクを共有するホスト群(それぞれをノードと呼びます)。
クラスタマネージャ	クラスタ内の各ノード上で実行される外部デーモンプロセス。各ノードのクラスタマネージャは相互に通信し、VxVM にクラスタメンバーシップの変更を通知します。
クラスタ共有ディスクグループ	複数ホストによる共有アクセスが可能なディスクで構成されるディスクグループ(共有ディスクグループともいいます)。
ストライプサイズ	ストライプ化するすべてのカラム上に 1 つのストライプを構成するストライプユニットサイズの合計。
ストライプ化	ストライプを使い、複数の物理ディスクにデータを分散させるレイアウト手法。データは各プレックスのサブディスク内のストライプに交互に割り当てられます。
スパープレックス	ボリュームより小さいプレックスまたは空白(バックアップ先のサブディスクを持たないプレックスの領域)が存在するプレックス。
スワップボリューム	スワップ領域として使うために設定された VxVM ボリューム。
スワップ領域	システムページャプロセスによりスワップアウトされたメモリページのコピーを保有するためのディスク領域。
セカンダリパス	アクティブ/パッシブディスクアレイでは、プライマリパス以外のディスクへのパスはセカンダリパスと呼ばれます。プライマリパスに障害が発生しない限り、ディスクへのアクセスにはプライマリパスのみが使われます。プライマリパスに障害が発生すると、ディスクの所有権はセカンダリパスの 1 つに移譲されます。

セクタ	サイズの単位。システムに応じて異なります。セクタサイズはデバイス(ハードドライブ、CD-ROM など)ごとに設定されます。システム内のすべてのデバイスは通常、相互運用性を確保するため同じセクタサイズに設定されていますが、そうでない場合もあります。 通常、1 セクタは 512 バイトです。
ディスク	読み取りおよび書き込みデータブロックの集合。インデックスが付いており、かなり高速にアクセスできます。各ディスクには全世界で一意的識別子が付けられます。
ディスク ID	各ディスクに付けられた全世界で一意的識別子。ディスクを移動した場合でも、この識別子を使ってディスクを識別することができます。
ディスクアクセスレコード	特定のディスクへのアクセスパスを指定するために使う設定レコード。各ディスクアクセスレコードには名前とタイプが記録されており、タイプ固有の情報が保存されている場合もあります。この情報は、VxVM がディスクアクセスレコードで定義されたディスクにアクセスし、それを操作する方法を決定するために使われます。
ディスクアクセス名	デバイス名の別称。
ディスクアレイシリアル番号	ディスクアレイのシリアル番号。通常はディスクアレイキャビネットに印刷されています。あるいは、ディスクアレイ上のディスクにベンダー固有の SCSI コマンドを発行して入手します。この番号は DMP サブシステムがディスクアレイを一意的に識別するために使います。
ディスクアレイ(disk array)	1 つのオブジェクト内に配置されたディスクの論理的集合。アレイ化すると、冗長性が高まり、処理効率が向上するなどの利点があります。
ディスクエンクロージャ	高機能なディスクアレイの 1 つ。通常、ファイバーチャネルループが組み込まれたバックプレーンを備えており、ディスクのホットスワップを実現できます。
ディスクグループ	共通の設定情報を共有するディスクの集合。ディスクグループ設定情報は、既存の VxVM オブジェクト(ディスクやボリュームの属性など)やその関係に関する詳細情報を含むレコードのセットです。各ディスクグループには管理者が割り当てた名前と内部で定義された一意の ID があります。ディスクグループ名の bootdg(ブートディスクグループのエイリアス)、defaultdg(デフォルトのディスクグループのエイリアス)、nodg(ディスクグループなしを示す)は予約済みです。
ディスクグループ ID	ディスクグループの識別に使う一意の識別子。
ディスクコントローラ	VxVM のマルチパスサブシステムにおいて、ホストに接続されているコントローラ(ホストバスアダプタ、すなわち HBA)またはディスクアレイを指します。オペレーティングシステムではディスクの親ノードとして表されます。
ディスクメディアレコード	ディスク ID で特定のディスクを識別する設定レコード。このレコードによりディスクに論理(管理)名が付けられます。
ディスクメディア名	デバイス名の別称。
ディスク名	disk03 など、VxVM の制御下にあるディスクに付けられた論理名または管理名。ディスク名は、ディスクメディア名ともいいます。

デバイス名	<p>sda や sda3 など、物理ディスクへのアクセスに使うデバイス名またはアドレス。ここで、sda はデバイス全体を示し、sda3 は sda 上の 3 番目のパーティションを示します。</p> <p><b>SAN</b> 環境では、エンクロージャに基づく名前の付け方を使うほうが便利です。これは、エンクロージャ名 (enc0 など) とエンクロージャ内のディスク番号を下線文字で連結したものをデバイス名とする方法です (enc0_2 など)。デバイス名は、ディスクアクセス名ともいいます。</p>
データストライプ	ストライプの中の使用可能なデータ部分。ストライプからパリティ領域を引いた値に等しくなります。
トランザクション	個別ではなく原子的に実行される一連の設定変更処理を指します。正常に処理されたものも問題が発生したものも含まれます。トランザクションは設定の一貫性を保証するために内部的に使われます。
ノード	クラスタ内のホストの 1 つ。
ノードのクリーンシャットダウン	共有ボリュームへのすべてのアクセスが停止したときに、クリーンな状態でクラスタを離れるノードの機能。
ノードの中止	緊急の場合に、進行中の操作を停止することなくノードがクラスタから切り離される状況。
ノード参加	ノードがクラスタに参加し、共有ディスクにアクセスできるようにするプロセス。
パス	ディスクをホストに接続する場合、そのディスクへのパスは、ホスト上の HBA (ホストバスアダプタ)、SCSI またはファイバケーブルコネクタ、ディスクまたはディスクアレイ上のコントローラで構成されます。これらのコンポーネントによって、ディスクへのパスが形成されます。いずれかのコンポーネントに障害が発生すると、DMP はそのディスクのすべての I/O を残りの (代替) パスにシフトします。
パスグループ	ディスクが vxddmp によってマルチパス化されていない場合は、VxVM が各パスを個々のディスクとして見ます。このような場合、ディスクへのすべてのパスをグループ化できます。グループ化することにより、VxVM ではグループ内のパスの 1 つのみが認識されるようになります。
パブリックリージョン	VxVM が管理する物理ディスクの領域。使用可能領域を持ち、サブディスクの割り当てに使われます。
パリティ	障害の発生後、データを復元するのに使われる計算値。データが RAID 5 ボリュームに書き込まれている間、データに対し、排他的論理和 (XOR) 演算を用いてパリティが計算されます。計算されたパリティはボリュームに書き込まれます。RAID 5 ボリュームの一部に障害が発生した場合、障害が発生したボリュームの該当する部分に存在していたデータは残りのデータとパリティから再作成されます。
パリティストライプユニット	パリティ情報を含んでいる RAID 5 ボリュームストレージ領域。パリティストライプユニットに含まれるデータは、I/O エラーやディスクでの障害により失われた RAID 5 ボリュームの領域を復元するのに使えます。
パーティション	オペレーティングシステムとディスクドライブで直接認識される、物理ディスクデバイスの区画。

ファブリックモードディスク	ストレージエリアネットワーク (SAN) 上でファイバーチャネルスイッチを介してアクセスできるディスクデバイス。
ブロック	ディスクまたはアレイにデータを転送する最小の単位。
プライベートリージョン	構造化された VxVM 管理情報を保存するために使う物理ディスクの領域。プライベートリージョンには、ディスクヘッダー、目次、設定データベースが格納されます。目次にはディスクの内容がマップされます。ディスクヘッダーにはディスク ID が格納されます。信頼性を向上させるために、プライベートリージョンのデータはすべて冗長性が確保されています。
プライマリパス	アクティブ/パッシブディスクアレイでは、ディスクをディスクアレイ上の特定の 1 つのコントローラにバインドしたり、1 つのコントローラで所有することができます。そして、その特定のコントローラを介し、パスを使って、ディスクにアクセスできます。
ブレックス	ブレックスとは、物理ディスクサイズなどの制約を受けないディスク領域エリアを作成する、サブディスクの論理集合です。ミラーは、1 つのボリュームに複数のデータブレックスを作成することによって設定されます。ミラーボリューム内の各データブレックスには、ボリュームデータと同一のコピーが格納されます。連結ボリュームレイアウト、ストライプボリュームレイアウトおよび RAID 5 ボリュームレイアウトに対応するブレックスを作成したり、ボリュームログを保存するブレックスを作成することもできます。
ホットスワップ	システムの電源を切らなくても、システムから削除したりシステムに挿入できるデバイスのこと。
ホットリロケーション	ディスクに障害が発生したとき、冗長性のリストアとミラーボリュームや RAID 5 ボリュームへのアクセス権のリストアを自動的に実行する手法。これは、影響を受けたサブディスクを、スペアとして指定されたディスクや同じディスクグループ内の空き領域に再配置することによって実現されます。
ボリューム	仮想ディスク。ファイルシステムやデータベースなどのアプリケーションで使うディスクブロックの指定可能な範囲を表します。ボリュームは 1 - 32 個のブレックスの集合です。
ボリュームデバイスドライバ	アプリケーションと物理デバイスドライバレベルの間に仮想ディスクドライブを形成するドライバ。ボリュームデバイスドライバには、キャラクタデバイスノードが <code>/dev/vx/rdisk</code> に表示され、ブロックデバイスノードが <code>/dev/vx/dsk</code> に表示される仮想ディスクデバイスノードを介してアクセスします。
ボリューム設定デバイス	ボリューム設定デバイス ( <code>/dev/vx/config</code> ) は、ボリュームデバイスドライバに対するすべての設定変更が実行されるインターフェースです。
マスタリングノード	ディスクが接続されているノード。ディスク所有者ともいいます。
マスターノード	クラスタ内の特定の VxVM 操作を統合するためにソフトウェアによって指定されるノード。どのノードでも、マスターノードとして指定できます。
マルチパス	システムに接続されているディスクへの物理アクセスパスが複数ある場合、ディスクはマルチパス化されているといえます。この事実を意識することなく利用できるホスト上のソフトウェア (DMP ドライバなど) を、マルチパス機能を搭載したソフトウェアと呼びます。

ミラー	ボリュームとそのボリューム内のデータの複製(サブディスクの順序付けられた集合)。各ミラーは、ミラーが関連付けられているボリュームの 1 つのプレックスで構成されます。
ミラー化	ボリュームの内容を複数のプレックスにミラー化するレイアウト手法。各プレックスにはボリュームに保存されているデータが複製されますが、プレックス自体のレイアウトは異なる場合があります。
ルータビリティ	<code>root</code> ファイルシステムや <code>swap</code> デバイスを <b>VxVM</b> 制御下に置く機能。その結果として作成されたボリュームは、冗長性を確保するためにミラー化でき、ディスク障害発生時に修復できます。
ルートディスク	ルートファイルシステムを含むディスク。 <b>VxVM</b> の制御下に置くことができます。
ルートパーティション	ルートファイルシステムが存在するディスク領域。
ルートファイルシステム	<b>UNIX</b> カーネルスタートアップシーケンスの一部としてマウントされる初期ファイルシステム。
ルートボリューム	ルートファイルシステムを含む <b>VxVM</b> ボリューム(システム構成でそのようなボリュームが指定されている場合)。
ルート設定	ルートディスクグループの設定データベース。他のディスクグループのレコードが常に保存されているという点で特殊です。このレコードはバックアップの目的のみに使われます。システム上のすべてのディスクデバイスを定義するディスクレコードも格納されます。
ログサブディスク	<b>DRL</b> を保存するために使うサブディスク。
ログプレックス	<b>RAID 5</b> ログを保存するために使うプレックス。ログプレックスという用語は、 <b>DRL</b> プレックスを指す場合にも使われます。
共有 <b>VM</b> ディスク	クラスタ内の共有ディスクグループに属する <b>VM</b> ディスク。
分散ロックマネージャ	クラスタ内の複数のシステム上で実行されるロックマネージャで、分散リソースへの一貫したアクセスを保証します。
切断	<b>VxVM</b> オブジェクトが別のオブジェクトと関連付けられているが、使えない状態。
専用ディスクグループ	クラスタ内のある 1 つのホストによってのみアクセスされるディスクで構成されるディスクグループ。
接続	<b>VxVM</b> オブジェクトが別のオブジェクトと相互に関連付けられ、使用可能になっている状態。
有効にされているパス	<b>I/O</b> に使えるディスクへのパス。
永続 <b>FastResync</b>	データ変更マップをディスクの <b>DCO</b> ボリュームに保存することによって、システムの再ブート後もマップを保持できる <b>FastResync</b> の形式の 1 つ。
永続状態ログ	リカバリには有効なミラーのみを使い、障害が発生したミラーがリカバリ対象として選択されないようにするログタイプ。カーネルログともいいます。
無効にされているパス	<b>I/O</b> に使えないディスクへのパス。パスは、実際のハードウェア障害が発生した場合や、ユーザーがコントローラで <code>vxdmpadm disable</code> コマンドを使ったときに、無効になります。

物理ディスク	下位ストレージデバイス。必ずしも <b>VxVM</b> の制御下にある必要はありません。
空きサブディスク	どのプレックスとも関連付けられておらず、 <code>putil[0]</code> フィールドが空であるサブディスク。
設定コピー	設定データベースの単一コピー。
設定データベース	既存の <b>VxVM</b> オブジェクト(ディスクやボリュームの属性など)に関する詳細情報を含むレコードのセット。
読み取り - ライトバックモード	各読み取り操作によって読み取り対象領域のプレックスの整合性を修復するリカバリモード。プレックスの整合性は、1 つのプレックスのブロックからデータを読み取り、他のすべての書き込み可能プレックスにデータを書き込むことによって修復されます。
連結	サブディスクが順番に連続して配列されているという特徴を持つレイアウトスタイル。
開始ノード	システム管理者が、 <b>VxVM</b> オブジェクトに変更を要求するユーティリティを実行するノード。ボリュームの再設定はこのノードで開始されます。
関連付け	<b>VxVM</b> オブジェクト間の関係を構築するプロセス。たとえば、プレックス内に開始点を持つように作成され定義されたサブディスクは、そのプレックスに関連付けられているといいます。
関連付けの解除	2 つの <b>VxVM</b> オブジェクト間に存在するリンクがすべて削除されるプロセス。たとえば、プレックスからサブディスクの関連付けを解除すると、プレックスからサブディスクが削除され、このサブディスクは空き領域プールに追加されます。
関連付けられたサブディスク	プレックスに関連付けられたサブディスク。
関連付けられたプレックス	ボリュームに関連付けられたプレックス。
関連付けを解除したサブディスク	プレックスとの関連付けを解除したサブディスク。
関連付けを解除したプレックス	ボリュームとの関連付けを解除したプレックス。
非永続 <b>FastResync</b>	データ変更マップをメモリに保存しているためにシステムの再ブート時にマップを保持できない <b>FastResync</b> の形式の 1 つ。

## 記号

/dev/vx/dmp ディレクトリ 11

/dev/vx/rdmp ディレクトリ 11

## A

A/A-A ディスクアレイ 9

A/A ディスクアレイ 9

A/P-C ディスクアレイ 10~11

A/PF ディスクアレイ 10

A/PG ディスクアレイ 11

A/P ディスクアレイ 10

active パス属性 65

adaptiveminq ポリシー 69

adaptive 負荷分散 69

APM

設定 83

APM (Array Policy Module)

設定 83

ASL

アレイサポートライブラリ 87~88

auto-trespass モード 10

## B

balanced パスポリシー 69

## C

check\_all ポリシー 81

check\_alternate ポリシー 82

check\_disabled ポリシー 82

check\_periodic ポリシー 82

## D

DDL 17

デバイス検出層 90

DISKS カテゴリ 88

ディスクの削除 101

ディスクの追加 98

認識されているディスクの一覧表示 97

DMP

check\_all リストアポリシー 81

check\_alternate リストアポリシー 82

check\_disabled リストアポリシー 82

check\_periodic リストアポリシー 82

DMP データベース情報の表示 41

DMP ノードによって制御されるパスの表示 49

DMP ノードの表示 47

DMP パスリストアスレッドの状態の表示 83

DMP パスリストアポリシーの設定 81

DMP リストアデーモンの停止 83

DMP リストアポーリング間隔の設定 81

dynamic multi-pathing 9

I/O エラーに対する応答の設定 76、80

I/O 調整の設定 78

I/O 統計情報の収集 59

path-switch チューニングパラメータ 138

recoveryoption 値の表示 80

TPD 情報の表示 55

vxdmpadm 44

アレイポートに関する情報の表示 53

アレイポートのパスの表示 50

アレイポートの無効化 74

アレイポートの有効化 75

エンクロージャに関する情報の表示 53

エンクロージャに対する DMP ノードの表示 46~47

エンクロージャに基づく名前の付け方 12

エンクロージャ名の変更 76

クラスタ環境上 15

コントローラに関する情報の表示 51

コントローラのパスの表示 50

コントローラの無効化 74

コントローラの有効化 75

セカンダリパスでの I/O のスケジューリング設定 71

ディスクデバイスの設定 85

ディスクの検出 85

テンプレートを使ったチューニング 126

ノード 11

ノードの LUN グループの表示 48

パスエージング 135

パス情報の表示 41

パスに対する DMP ノードの表示 46

パスの無効化 74

パスの有効化 75

パスフェールオーバー機構 13

負荷分散 14

メタノード 11

リストアポリシー 81

ログ記録レベル 136

dmp\_cache\_open チューニングパラメータ 134

dmp\_daemon\_count tunable 134

dmp\_delayq\_interval tunable 134

dmp\_fast\_recovery tunable 135

dmp\_health\_time チューニングパラメータ 135

dmp\_log\_level チューニングパラメータ 136

dmp\_low\_impact\_probe 136

dmp\_lun\_retry\_timeout チューニングパラメータ 137

dmp\_monitor\_fabric チューニングパラメータ 137

dmp\_monitor\_ownership tunable 137

dmp\_native\_support チューニングパラメータ 138

dmp\_path\_age チューニングパラメータ 138

dmp\_pathswitch\_blks\_shift チューニングパラメータ 138

dmp\_probe\_idle\_lun チューニングパラメータ 139

dmp\_probe\_threshold チューニングパラメータ 139

dmp\_restore\_cycles チューニングパラメータ 139

dmp\_restore\_interval チューニングパラメータ 139

dmp\_restore\_state チューニングパラメータ 140

dmp\_scsi\_timeout tunable 140

dmp\_sfg\_threshold チューニングパラメータ 140

dmp\_stat\_interval tunable 140

DMP サポート

JBOD デバイス 87

DMP の設定

テンプレートの使用 126

DMP のチューニング

テンプレートの使用 126

DMP ノード

設定名 43

統合情報の表示 47

DMP のパスフェールオーバー 13

DMP リストアのポーリング間隔 81

## E

errorcd デーモン 13

## F

FAILFAST フラグ 13

## H

HBA

一覧表示するターゲット 92

一覧表示のサポート対象 92

ポートの一覧表示 92

HBA 情報

表示 52

hdx 17

hdx に基づく名前の付け方 18

## I

I/O

DMP の統計情報の収集 59

セカンダリパスでのスケジュール設定 71

調整 13

I/O 調整 78

I/O 調整オプション

設定 81

I/O ポリシー

指定 68

表示 68

例 71

iSCSI パラメータ

DDL での管理 94

vxddladm での設定 94

## J

JBOD

DISKS カテゴリからのディスクの削除 101

DISKS カテゴリへのディスクの追加 98

DMP サポート 87

認識されているディスクの一覧表示 97

## L

LUN 10

アイドル状態 139

LUN グループ

詳細の表示 48

LUN グループフェールオーバー 11

## M

mrl

キーワード 67

## N

nomanual パス属性 65

nopreferred パス属性 65

## O

OTHER\_DISKS カテゴリ 88

**P**

preferred priority パス属性 66  
 primary パス属性 66  
 priority 負荷分散 70

**R**

recovery option 値  
   設定 80  
 restored デーモン 13  
 round-robin  
   負荷分散 70

**S**

scandisks  
   vxdisk サブコマンド 86  
 sdx に基づく名前の付け方 18  
 secondary パス属性 66  
 single active パスポリシー 70  
 standby パス属性 66

**T**

TPD  
   共存のサポート 89  
   パス情報の表示 55  
 tpdmode 属性 105

**U**

use\_all\_paths attribute 71  
 use\_avid  
   vxddladm オプション 103

**V**

vxdctl enable  
   新しいディスクの設定 85  
   デバイス検出機能の呼び出し 89  
 vxddladm  
   DISKS カテゴリからのディスクの削除 101  
   DISKS カテゴリ内の認識されているディスクの一覧  
   表示 97  
   DISKS カテゴリへのディスクの追加 99  
   iSCSI パラメータの設定 94  
   外部デバイスの追加 101  
   サポートされているディスクアレイの一覧表示 96  
   サポートされる HBA の一覧表示 92  
   すべてのデバイスの一覧表示 91  
   設定されたターゲットの一覧表示 93  
   設定されたデバイスの一覧表示 94

ディスクアレイのサポートを無効化するために使用 96  
 ディスクアレイのサポートを有効化するために使用 96  
 ディスクの名前の付け方の表示 104  
 名前の付け方の変更 103  
 ホストバスアダプタ上のポートの一覧表示 92  
 無効にされたディスクアレイの一覧表示 97、99

**vxdisk**

ディスクアクセス名の検出 106  
 ディスクデバイスのスキャン 86  
 マルチパス情報の表示 42

**vxdisk scandisks**

デバイスの再スキャン 86  
 デバイスのスキャン 86

**vxdmpadm**

APM 情報の表示 84  
 APM の削除 84  
 APM の設定 84  
 DMP データベース情報の表示 41  
 DMP での I/O の無効化 74  
 DMP での I/O の有効化 75  
 DMP のコントローラの無効化 40  
 DMP ノードによって制御されるパスを表示する 49  
 DMP パスリストアポリシーの指定 81  
 DMP リストアスレッドの状態の表示 83  
 DMP リストアデーモンの停止 83  
 I/O エラーに対する応答の設定 76、80  
 I/O エラーリカバリ設定の表示 80  
 I/O 調整の設定 78  
 I/O 調整の設定の表示 80  
 I/O 統計情報の収集 59  
 I/O ポリシーの設定 70  
 I/O ポリシーの表示 68  
 TPD 情報の表示 55  
 TPD 名前の付け方の変更 105  
 アレイポートに関する情報の一覧表示 53  
 エンクロージャに関する情報の表示 53  
 エンクロージャに対する DMP ノードの表示 46～47  
 エンクロージャ名の変更 76  
 コントローラに関する情報の表示 51  
 ディスクアクセス名の検出 106  
 ディスクの名前の付け方の変更 102  
 パス属性の設定 66  
 パスに対する DMP ノードの表示 46、48  
 パーティションサイズの表示 68  
 リストアポーリング間隔の設定 81  
**vxdmpadm** リスト  
 DMP ノードの表示 47  
**vxprint**  
 エンクロージャに基づくディスク名 106

エンクロージャに基づくディスク名で使用 106

## VxVM

ディスクの検出 87

## W

WWN 識別子 18

## あ

アイドル状態の LUN 139

アクセスポート 10

アクティブ/アクティブディスクアレイ 9

アクティブ/パッシブディスクアレイ 10

アクティブパス

デバイス 66~67

アレイ

DMP サポート 87

アレイサポートライブラリ (ASL) 88

アレイのボリューム ID

デバイス命名 103

アレイポート

DMP の無効化 74

DMP の有効化 75

情報の表示 53

一覧表示

DMP ノード 47

サポートされているディスクアレイ 96

永続

デバイス命名オプション 103

永続的なデバイス名データベース 104

永続的なデバイス命名 104

エラー

一時的なエラーの処理 137

エンクロージャ 19

ディスクアクセス名の検出 106

に関する情報の表示 53

パスに関する属性の設定 65、67

パスの冗長性 66~67

エンクロージャに基づく名前の付け方 19、21、102

DMP 12

vxprint 106

オプション値の再試行

設定 80

## か

外部デバイス

追加 101

概要

DMP 8

カスタム名

DMP ノード 43

カテゴリ

ディスク 88

キューに入れられた I/O

統計の表示 63

クラスタ

DMP の使用 15

コントローラ

DMP による無効化 40

DMP の無効化 74

DMP の有効化 75

情報の表示 51

コントローラ ID

表示 52

## さ

最少キュー負荷分散ポリシー 70

サードパーティ製ドライバ (TPD) 89

指定

冗長性のレベル 67

冗長性の最小レベル

デバイスの指定 67

デバイスの表示 66

冗長性のレベル

デバイスの指定 67

デバイスの表示 66

冗長ループアクセス 20

ストレージプロセッサ 10

セカンダリパス 10

セカンダリパスの表示 42

設定

パスの冗長性レベル 67

属性

active 65

nomannual 65

nopreferred 65

preferred priority 66

primary 66

secondary 66

standby 66

パスに関する設定 65

パスの設定 67

## た

ターゲット

一覧表示 92

## チューニングパラメータ

dmp\_cache\_open 134  
 dmp\_daemon\_count 134  
 dmp\_delayq\_interval 134  
 dmp\_fast\_recovery 135  
 dmp\_health\_time 135  
 dmp\_log\_level 136  
 dmp\_low\_impact\_probe 136  
 dmp\_lun\_retry\_timeout 137  
 dmp\_monitor\_fabric 137  
 dmp\_monitor\_ownership 137  
 dmp\_native\_support 138  
 dmp\_path\_age 138  
 dmp\_pathswitch\_blks\_shift 138  
 dmp\_probe\_idle\_lun 139  
 dmp\_probe\_threshold 139  
 dmp\_restore\_cycles 139  
 dmp\_restore\_interval 139  
 dmp\_restore\_state 140  
 dmp\_scsi\_timeout 140  
 dmp\_sfg\_threshold 140  
 dmp\_stat\_interval 140

## 調整 13

## ディスク 88

DISKS カテゴリからの削除 101  
 DISKS カテゴリへの追加 98  
 DMP による検出 85  
 JBOD で認識されているディスクの一覧表示 97  
 OTHER\_DISKS カテゴリ 88  
 VxVM による検出 87  
 新しく追加されたディスクの設定 85  
 アレイサポートライブラリ 88  
 永続的な名前の設定 104  
 エンクロージャ 19  
 カテゴリ 88  
 検出機能の呼び出し 89  
 スキャン 86  
 セカンダリパス 42  
 デバイス検出層 90  
 名前の付け方 18  
 名前の付け方の表示 104  
 名前の付け方の変更 102  
 プライマリパス 42  
 無効パス 42  
 メタデバイス 17  
 有効パス 42

## ディスクアレイ

A/A 9  
 A/A-A 9

## A/P 10

## A/PF 10

## A/P-G 11

DISKS カテゴリからのディスクの削除 101

DISKS カテゴリ内の認識されているディスクの一覧表示 97

DISKS カテゴリへのディスクの追加 98

DMP でのサポート 96

JBOD デバイス 87

アクティブ/アクティブ 9

アクティブ/パッシブ 10

サポートされているディスクアレイの一覧表示 96

サポートの無効化 96

サポートの有効化 96

非対称アクティブ/アクティブ 9

マルチパス化 16

無効にされているディスクアレイの一覧表示 97

## ディスク名

永続的な名前の設定 104

## デバイス

JBOD 87

外部デバイスの追加 101

すべての一覧表示 91

パスの冗長性 66~67

パス名 17

ファブリックデバイス 86

メタデバイス 17

## デバイス検出

partial 86

概要 17

デバイス検出層 90

デバイス検出層 (DDL) 17、90

デバイス名 17

永続的な名前設定 104

ユーザー指定 43

統計情報収集 13

統計の表示

キューに入れられた I/O 63

無効な I/O 63

## な

## 名前

デバイス 17

名前の付け方

TPD エンクロージャに対する変更 105

ディスク 18

ディスクに対する表示 104

ディスクのレイアウト属性の変更 102

## ノード

DMP 11

## は

## パス

DMP の無効化 74

DMP の有効化 75

属性の設定 65、67

パスエージング 135

パーティションサイズ

値の表示 68

指定 69

## パフォーマンス

DMP の負荷分散 14

非 `auto-trespass` モード 10

非対称アクティブ/アクティブディスクアレイ 9

非明示的フェールオーバーモード 10

## 表示

DMP ノード 47

HBA 情報 52

サポートされているディスクアレイ 96

冗長性のレベル 66

ピンポン効果 15

ファブリックデバイス 86

フェールオーバーモード 10

負荷分散 9

ポリシーの指定 68

ポリシーの表示 68

部分的なデバイス検出 86

プライマリパス 10、42

## ポート

一覧表示 92

## ま

## マルチパス化

情報の表示 41

## 無効な I/O

統計の表示 63

無効パス 42

明示的フェールオーバーモード 10

## 命名

DMP ノード 43

メタデバイス 17

## メタノード

DMP 11

## や

## 有効パス

表示 42

ユーザー指定デバイス名 43

## ら

## リストアポリシー

`check_all` 81`check_alternate` 82`check_disabled` 82`check_periodic` 82

論理ユニット番号 10

## わ

ワールドワイドネーム識別子 18