

Veritas™ Dynamic Multi-Pathing 管理者ガイド

Solaris

6.0.1

Veritas™ Dynamic Multi-Pathing 管理者ガイド

このマニュアルで説明するソフトウェアは、使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができます。

製品バージョン: 6.0.1

マニュアルバージョン: 6.0.1 Rev 0

著作権について

Copyright © 2012 Symantec Corporation. All rights reserved.

Symantec、Symantec ロゴ、Veritas、Veritas Storage Foundation、CommandCentral、NetBackup、Enterprise Vault、LiveUpdate は、Symantec Corporation または同社の米国およびその他の国における関連会社の商標または登録商標です。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。

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、Symantec Corporation 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限り、免責されるものとします。Symantec Corporation は、本書の供給、性能、使用に関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。

ライセンス対象ソフトウェアと関連書類は、FAR 12.212 の規定によって商用コンピュータソフトウェアとみなされ、場合に応じて、FAR 52.227-19「Commercial Computer Software - Restricted Rights」、DFARS 227.7202「Rights in Commercial Computer Software or Commercial Computer Software Documentation」、そ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。米国政府によるライセンス対象ソフトウェアと関連書類の使用、修正、複製のリリース、実演、表示または開示は、本使用許諾契約の条項に従ってのみ行われるものとします。

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

<http://www.symantec.com>

第 1 章	DMP について	7
	Veritas Dynamic Multi-Pathing (DMP) について	7
	DMP の動作方法	8
	パスでの I/O を DMP で監視する方法	12
	負荷分散	13
	動的再設定	14
	DMP デバイスからの起動について	14
	クラスタ環境における DMP	14
	ディスクアレイへの複数パス	15
	デバイス検出	16
	ディスクデバイス	16
	DMP でのディスクデバイスの命名	16
	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	17
	エンクロージャに基づく名前の付け方について	18
第 2 章	ネイティブデバイス管理のための DMP の設定	23
	ネイティブデバイス管理のための DMP の設定について	23
	DMP への ZFS プールの移行	25
	EMC PowerPath から DMP への移行	25
	Hitachi Data Link Manager (HDLM) から DMP への移行	26
	Solaris Multiplexed I/O (MPxIO) から DMP への移行	27
	Oracle Automatic Storage Management (ASM) での Dynamic Multi-Pathing (DMP) デバイスの使用	28
	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	28
	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	29
	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	30
	既存の ZFS プールへの DMP デバイスの追加または ZFS プールの新規 作成	33
	ネイティブマルチパス設定の表示	35
	ネイティブデバイスの DMP サポートの削除	35

第 3 章	DMP の管理	37
	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	37
	DMP データベース情報の表示について	38
	ディスクへのバスの表示	38
	DMP ノードのカスタム名の設定	41
	DMP とネイティブマルチバスの共存	42
	vxddmpadm を使った DMP の管理	43
	DMP ノードに関する情報の取得	44
	DMP ノードについての統合された情報の表示	45
	LUN グループのメンバーの表示	47
	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バスの表示	47
	コントローラに関する情報の表示	50
	エンクロージャに関する情報の表示	51
	アレイポートに関する情報の表示	51
	TPD 制御デバイスに関する情報の表示	52
	拡張デバイス属性の表示	53
	VxVM の制御下におけるデバイスの無効化と有効化	55
	I/O 統計情報の収集と表示	56
	エンクロージャへのバスに関する属性の設定	62
	デバイスまたはエンクロージャの冗長レベルの表示	63
	アクティブバスの最小数の指定	64
	I/O ポリシーの表示	65
	I/O ポリシーの指定	65
	バス、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	72
	バス、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	73
	エンクロージャ名の変更	74
	I/O エラーに対する応答の設定	75
	I/O 調整機構の設定	76
	サブバスフェールオーバーグループ (SFG) の設定	77
	LIPP (Low-Impact Path Probing) の設定	78
	リカバリオプション値の表示	78
	DMP バスリストアポリシーの設定	79
	DMP バスリストアスレッドの停止	81
	DMP バスリストアスレッドの状態の表示	81
	アレイポリシーモジュール (Array Policy Modules) の設定	81

第 4 章	ディスクの管理	83
	ディスク管理について	83
	新しく追加されたディスクデバイスの検出と設定	83
	部分的なデバイス検出	84
	ディスクの検出とディスクアレイの動的な追加	85
	サードパーティ製ドライバの共存	87
	デバイス検出層の管理方法	89
	VxVM の ZFS との共存	101
	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	105
	永続的なデバイス名の再生成	105
	TPD 制御のエンクロージャに対するデバイスの命名の変更	106
	simple ディスクまたは nopriv ディスクのエンクロージャに基づく名前 の付け方	107
	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	109
第 5 章	デバイスの動的再設定	111
	オンラインの Dynamic Reconfiguration について	111
	DMP の制御下にある LUN のオンラインでの再設定	111
	既存のターゲット ID からの LUN の動的削除	112
	新しいターゲット ID への新しい LUN の動的追加	114
	既存のターゲット ID からの LUN の置換	115
	アレイ側からの LUN の特性の変更	116
	ホストバスアダプタのオンラインでの交換	116
	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	117
第 6 章	イベント監視	119
	Dynamic Multi-Pathing (DMP) のイベントソースデーモン(vxesd)につい て	119
	ファブリック監視と予防的なエラー検出	120
	Dynamic Multi-Pathing (DMP) の自動デバイス検出	121
	Dynamic Multi-Pathing (DMP) の iSCSI および SAN ファイバーチャネ ルトポロジの検出	121
	DMP イベントログ	122
	Dynamic Multi-Pathing (DMP) のイベントソースデーモンの起動と停 止	122

第 7 章	処理効率の監視とチューニング	125
	テンプレートを使った Veritas Dynamic Multi-Pathing (DMP) のチュー ニングについて	125
	DMP チューニングテンプレート	126
	DMP チューニングテンプレートの例	127
	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	130
	DMP 設定ファイルの管理	131
	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	132
	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	132
	DMP チューニングパラメータ	133
	用語集	141
	索引	149

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。

- [Veritas Dynamic Multi-Pathing \(DMP\) について](#)
- [DMP の動作方法](#)
- [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](#)

Veritas Dynamic Multi-Pathing (DMP) について

Veritas 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 (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 はスタンドアロン製品としても利用できます。DMP メタデバイスで ZFS をサポートするように拡張されています。DMP メタデバイス上に ZFS プールを作成できます。DMP は非ルート ZFS ファイルシステムのみをサポートします。

Veritas Dynamic Multi-Pathing は、Storage Foundation 製品とは別途にライセンスを取得できます。Veritas Volume Manager と Veritas File System の機能は、DMP のライセンスを取得しても提供されません。

Storage Foundation (SF) Enterprise ライセンス、SF HA Enterprise ライセンス、Storage Foundation Standard ライセンスを取得した場合は、DMP の機能を使うことができます。

Veritas Volume Manager (VxVM) ボリュームとディスクグループは ZFS プールと共存できますが、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 **ZFS** で利用できません。同様に、ディスクが **ZFS** 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

Veritas Dynamic Multi-Pathing (DMP) は、パスフェールオーバーと負荷分散を行って、可用性、信頼性、パフォーマンスを向上します。この機能は、様々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。ディスクへの様々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各種エンクロージャを識別できます。

p.83 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。

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。

アクティブ/パッシブ (A/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。

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。

このポリシーは、1 つのコントローラに複数のプライマリパスを持つことにより、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、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。

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/P-F)

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。

このポリシーは、1 つのコントローラに複数のプライマリパスを持つことにより、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、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。

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/P-G)

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/P-G 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ま
す。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および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。

このポリシーは、1 つのコントローラに複数のプライマリパスを持つことにより、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、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。

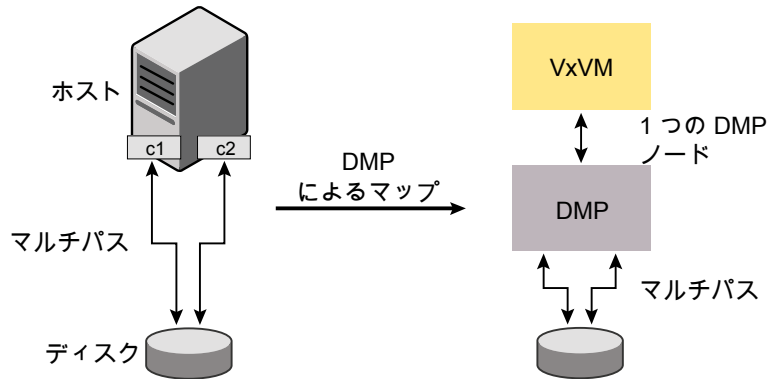
アレイポリシーモジュール (APM) では、DMP がサポートする標準タイプ以外のアレイタイプを DMP に定義できます。

Veritas 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、`/dev/vx/rdmp` と `/dev/vx/dmp` のディレクトリ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。

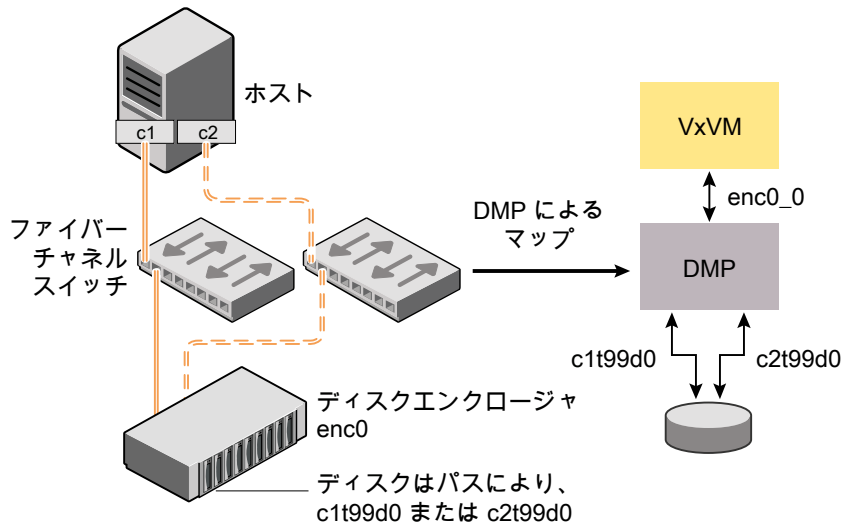
図 1-1 DMP で、ディスクに対する複数の物理パスを 1 つのノードで表す方法



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(c1t99d0 と c2t99d0)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.18 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.83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、パスリスト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、パスリスト、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。`vxddmpadm gettune`コマンドを使うと、スレッドに関する情報が提供されます。`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をスケジュール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(通常は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.79の「[DMPパスリストポリシーの設定](#)」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.12の「[パスフェールオーバー機構](#)」を参照してください。

p.13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.75 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.77 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、パス上に I/O アクティビティが存在しない場合でも、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実行さ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.78 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.76 の「[I/O 調整機構の設定](#)」を参照してください。

負荷分散

Dynamic Multi-Pathing (DMP) では、デフォルトで最少キュー I/O ポリシーを使って、アクティブ/アクティブ (A/A)、アクティブ/パッシブ (A/P)、明示的フェールオーバーモードのアクティブ/パッシブ (A/P-F)、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/P-G) のディスクアレイのパス間で負荷分散が行われます。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。

A/P ディスクアレイの場合は、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限り行われません。

A/P、A/PF、A/PG アレイの場合、A/A アレイの場合と同様に、現在アクティブなすべてのパス間で負荷分散が行われます。

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。

p.65 の「[I/O ポリシーの指定](#)」を参照してください。

動的再設定

動的再設定 (DR) は、一部のハイエンドエンタープライズシステムで使用可能な機能です。この機能を使うと、システムがまだ動作している間に、コンポーネント (CPU、メモリ、その他のコントローラや I/O ボードなど) を再設定できます。VxVM によって制御されているディスクが、再設定されるコンポーネントによって処理される可能性があります。

p.37 の「[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](#)」を参照してください。

DMP デバイスからの起動について

ルートディスクを VxVM の制御下に置くと、単一ディスクの場合はパスを 1 つ持つ DMP デバイスとして、マルチポートディスクアレイ内のディスクの場合は複数のパスを持つ DMP デバイスとして、自動的にアクセスされます。ルートディスクのカプセル化とミラー化を行うと、ディスクへの既存の物理パスが 1 つ以上失われた場合に対するシステムの信頼性が向上します。

メモ: SAN のブート可能 LUN は DMP によって制御する必要があります。SAN のブート可能 LUN の PowerPath と MPxIO の制御はサポートされていません。

クラスタ環境における DMP

メモ: VxVM (Veritas Volume Manager) のクラスタ機能を使うには、追加ライセンスが必要です。クラスタ化は VxVM に対してのみサポートされます。

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 `vxdctl 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す。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.85 の「[ディスクの検出とディスクアレイの動的な追加](#)」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.89 の「[デバイス検出層の管理方法](#)」を参照してください。

p.81 の「[アレイポリシーモジュール\(Array Policy Modules\)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。

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.8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。

すなわち、新しいディスクアレイをホストに動的に追加し、接続されているすべてのディスクデバイスについて、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するコマンドを実行し、新しいデバイスデータベースでDMPを再設定できるということです。

p.89の「[デバイス検出層の管理方法](#)」を参照してください。

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/[r]disk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。

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.8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.16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従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.86 の「[ディスクカテゴリ](#)」を参照してください。

デバイス名には次の名前の付け方の 1 つを使用できます。

- オペレーティングシステムに基づく名前の付け方
p.17 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.18 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に、`c#t#d#s#` 形式で名前が設定されます。

デバイス名の構文は `c#t#d#s#` です。この構文中の `c#` はホストバスアダプタ上のコントローラを、`t#` はターゲットコントローラ ID を、`d#` はターゲットコントローラ上のディスクを、`s#` はディスク上のパーティション(またはスライス)を表します。

メモ: 非 EFI ディスクの場合、スライス `s2` がディスク全体を表します。EFI ディスクと非 EFI ディスクのどちらの場合も、デバイス名のスライス部分を省略した場合はディスク全体を表すと見なされます。

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 はコントローラ別に名前をソートし、最も小さいコントローラ番号を選択します。たとえば、`c2` ではなく `c1` を選択します。同じコントローラから複数のパスが認識される場合、DMP はターゲット名が最も小さいパスを使います。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っています。

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。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。

VxVM は、多くの場合、デバイス名 `c0t0d0` によって、ブートディスク(ルートファイルシステムを含み、システムの起動に使用)を認識します。

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。

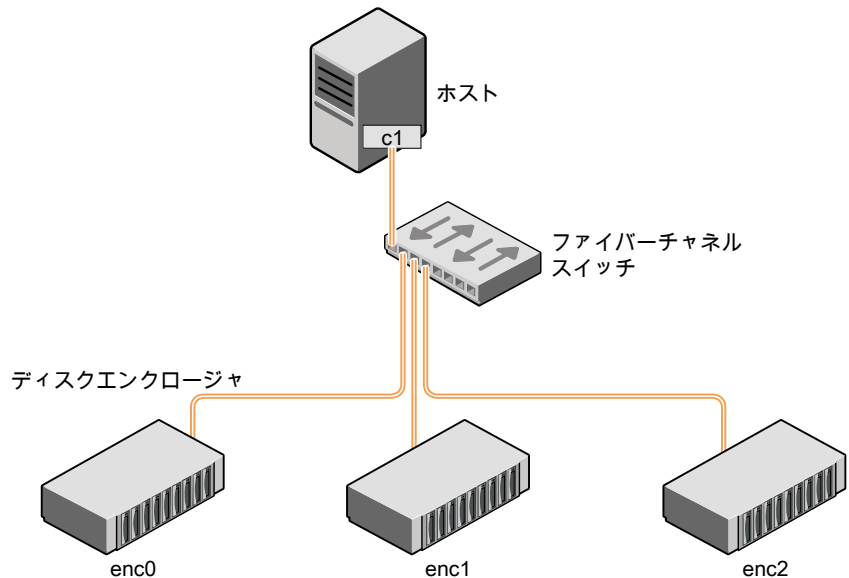
p.102の「ディスクデバイスの名前の付け方の変更」を参照してください。

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。この規則を使うと、アクセスに使うコントローラではなくエンクロージャの名前でディスクデバイスの名前を設定できます。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。たとえば、`c##t#d##s#`形式の名前の付け方では、コントローラベースでデバイス名を割り当てるため、同じホストコントローラに接続されている場合でも、エンクロージャが異なる場合があります。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。

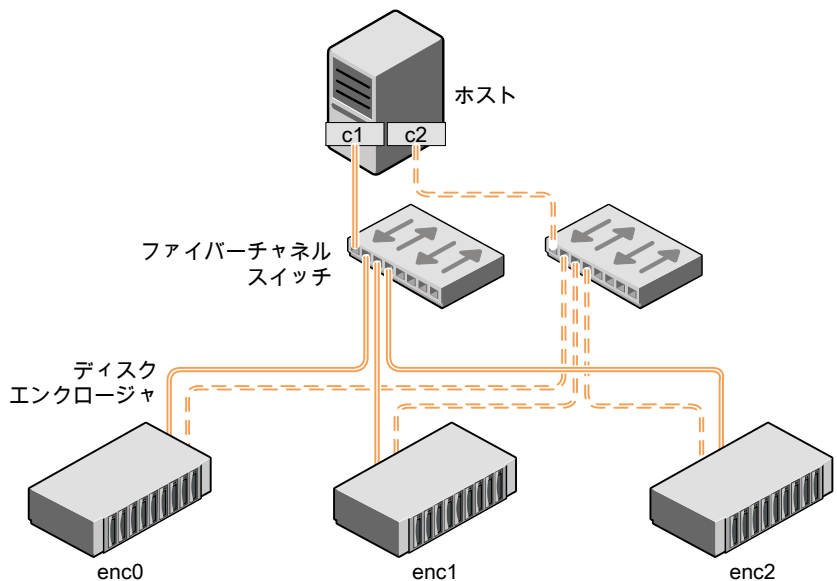
エンクロージャに基づく命名規則のもう1つの重要な利点は、VxVMがデータの冗長性のあるコピーを同じエンクロージャに配置するのを回避できることです。各エンクロージャは独立した障害のあるドメインと認識される可能性があるため、そのような配置を回避するのは好ましいことです。たとえば、ミラー化したボリュームがエンクロージャ enc1 のディスク上にもみ設定された場合は、スイッチとエンクロージャ間のケーブル障害により、ボリューム全体が利用できなくなることがあります。

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変換することができます。

p.74の「エンクロージャ名の変更」を参照してください。

「[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バ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、`c1t99d0` と `c2t99d0` が認識されています。

p.16 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

`vxdmpadm` コマンドを使ってエンクロージャ名を管理できます。

p.74 の「[エンクロージャ名の変更](#)」を参照してください。

`vxdmpadm(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せん。
- `OTHER_DISKS` カテゴリ内のディスク (DMP でマルチパス化されていないディスク) には、`c#t#d#s#` 形式で名前が設定されます。

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.74 の「[エンクロージャ名の変更](#)」を参照してください。

p.86 の「[ディスクカテゴリ](#)」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 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の内部では、DMP は DMP メタノードの名前の中でアレイボリューム ID (AVID) をインデックスとして使います。DMP メタノード名は `enclosureID_AVID` の形式です。

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、PowerPath の名前については 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 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 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](#)
- [DMP への ZFS プールの移行](#)
- [EMC PowerPath から DMP への移行](#)
- [Hitachi Data Link Manager \(HDLM\) から DMP への移行](#)
- [Solaris Multiplexed I/O \(MPxIO\) 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 ZFS プールへの DMP デバイスの追加または ZFS プールの新規作成](#)
- [ネイティブマルチパス設定の表示](#)
- [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、ZFS プールと、それらのプール上で動作する ZFS 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 をインストールした後に、ZFS で使うために DMP を設定します。ZFS 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 ZFS

のサポートを有効にします。また、`dmp_native_support` チューニングパラメータを有効にすると、使用中でないすべての ZFS プールが DMP デバイス上に移行されます。

`dmp_native_support` チューニングパラメータは、次のようにして ZFS で DMP のサポートを有効にします。

ZFS プール ZFS プールが使用中でない場合、ネイティブサポートを有効にするとプールが DMP デバイスに移行します。

ZFS プールが使用中の場合は、プールを無効にする手順を実行し、プールを DMP に移行します。

Veritas Volume Manager (VxVM) デバイス VxVM ラベルが付いたデバイスでは、ネイティブサポートが有効になりません。デバイスを ZFS で利用できるようにするには、VxVM ラベルを削除します。

VxVM デバイスは DMP 制御下のネイティブデバイスと共存できます。

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限り、DMP はデバイスを管理しません。TPD サポートを削除した後に、`dmp_native_support` チューニングパラメータを有効にするとデバイスが移行されます。

TPD デバイス上に ZFS プールを構築した場合は、ZFS プールを DMP デバイス上に移行するための手順を実行します。

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

```
# vxddmpadm settune dmp_native_support=on
```

この操作を最初に実行したとき、コマンドはプールが使用中かどうかを報告し、プールの移行はしません。プールを DMP 上に移行するには、プールを停止します。次に、`vxddmpadm settune` コマンドをもう一度実行して、プールを DMP 上に移行します。

`dmp_native_support` チューニングパラメータの値を確認するには、次のコマンドを使います。

```
# vxddmpadm gettune dmp_native_support
Tunable                               Current Value   Default Value
-----                               -
dmp_native_support                     on              off
```

DMP への ZFS プールの移行

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、ZFS プールと、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イ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
Solaris 10	EMC PowerPath	p.25 の「 EMC PowerPath から DMP への移行 」を参照してください。
Solaris 10	Hitachi Data Link Manager (HDLM)	p.26 の「 Hitachi Data Link Manager (HDLM) から DMP への移行 」を参照してください。
Solaris 10	Solaris Multiplexed I/O (MPxIO)	p.27 の「 Solaris Multiplexed I/O (MPxIO) から DMP への移行 」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。

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

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 ZFS プールの DMP サポートを有効にします。

```
# vxddmpadm settune dmp_native_support=on
```

- 2 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。

- 3 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。
- 4 PowerPath デバイスを使う ZFS プールをエクスポートします。

```
# zpool export poolname
```
- 5 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
```

ここで、*emcpowerXXXX* はデバイスの名前です。
- 6 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
```
- 7 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
```
- 8 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
```
- 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、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

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。
- 3 HDLM デバイスを使う ZFS プールをエクスポートします。

```
# zpool export poolname
```

- 4 HDLM パッケージをアンインストールします。
- 5 ZFS プールの DMP サポートを有効にします。

```
# vxddmpadm settune dmp_native_support=on
```

- 6 システムを再起動します。
- 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 ZFS プールが存在する場合は、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。

Solaris Multiplexed I/O (MPxIO) から DMP への移行

ここでは、デバイスを Solaris Multiplexed I/O (MPxIO) 制御から削除し、デバイス上で 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
- 手順で 1 つ以上のホストの再起動が必要になる

デバイスを MPxIO の制御から外し、デバイスで DMP を有効にするには

- 1 MPxIO デバイスを使うアプリケーションを停止します。
- 2 MPxIO 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 MPxIO デバイスで稼働する ZFS プールを非アクティブ化します。

- 4 ZFS プールの DMP サポートを有効にします。

```
# vxddmpadm settune dmp_native_support=on
```

- 5 次のコマンドを使って MPxIO を無効にします。

```
# stmsboot -d
```

- 6 システムを再ブートします。

再ブート後は、DMP が ZFS プールを制御します。すべての ZFS プールは DMP デバイス上に移行されます。

- 7 ファイルシステムをマウントします。

- 8 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

このリリースの 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.28 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.29 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.30 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/*) とブロックデバイス (/dev/vx/dmp) で利用可能です。

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。

値は /dev/vx/dmp/* または /dev/vx/rdmp/* に設定できます。

次に例を示します。

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

- 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/dmp/eva4k6k0_1  CANDIDATE
... ..
```

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き、これらのデバイスを Veritas 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。

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 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。
- 2 ASM で、ASM ディスクグループをマウント解除します。
- 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。

MPxIO または PowerPath から移行します。

p.23 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。

- 4 root ユーザーとして、手順 1 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。

- 5 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/dmp/* です。
- 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。
- 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。

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;
NAME                                PATH                                HEADER_STATUS
-----
ASM_DG1_0001 /dev/rds/c2t5006016130206782d9s6 MEMBER
ASM_DG1_0000 /dev/rds/c2t50001FE1500A8F08d1s6 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。

メモ: この手順には、システムの計画的な停止時間が必要となる場合があります。

p.23 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 1 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、vxmpadm 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。

```
# vxmpadm getdmpnode nodename=c2t5d9
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。

```
# vxmpraw 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。

```
# vxmpraw enable oracle dba 660
```

- 5 ASM から、ASM_DISKSTRING を設定します。

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

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

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

- 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/dmp/EVA4K6K0_0s6	MEMBER
	/dev/vx/dmp/EMC_CLARiion0_208s6	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 name,path,header_status from v$asm_disk where
header_status='MEMBER';
NAME                                PATH                                HEADER_STATUS
-----
ASM_DG1_0000 /dev/vx/dmp/EVA4K6K0_0s6          MEMBER
ASM_DG1_0001 /dev/vx/dmp/EMC_CLARiion0_208s6    MEMBER
```

既存の ZFS プールへの DMP デバイスの追加または ZFS プールの新規作成

dmp_native_support が ON の場合、利用可能な DMP デバイス上に新しい ZFS プールを作成できます。利用可能な DMP デバイスを既存の ZFS プールに追加することもできます。ZFS プールを DMP デバイス上に配置した後に、任意の ZFS コマンドを使ってプールを管理できます。

DMP デバイス上に ZFS プールを新規作成または DMP デバイスを既存の ZFS プールに追加するには

- 1 ZFS のために利用可能なディスクを選択します。vxdisk list コマンドは、VxVM によって使用中ではなく、タイプ (TYPE) が auto:none で、状態 (STATUS) が Online invalid であるディスクを表示します。

```
# vxdisk list

DEVICE                TYPE      DISK  GROUP  STATUS
. . .
tagmastore-usp0_0079  auto:none -    -      online invalid
tagmastore-usp0_0080  auto:none -    -      online invalid
```

- 2 DMP デバイス上に ZFS プールを新規作成します。

```
# zpool create newpool tagmastore-usp0_0079s2
# zpool status newpool
  pool: newpool
  state: ONLINE
  scrub: none requested
config:

          NAME                STATE      READ WRITE CKSUM
          newpool              ONLINE    0    0    0
          tagmastore-usp0_0079s2  ONLINE    0    0    0
```

- 3 既存の ZFS プールに DMP デバイスを追加します。

```
# zpool add newpool tagmastore-usp0_0080s2
# zpool status newpool
  pool: newpool
  state: ONLINE
  scrub: none requested
config:

          NAME                STATE      READ WRITE CKSUM
          newpool              ONLINE    0    0    0
          tagmastore-usp0_0079s2  ONLINE    0    0    0
          tagmastore-usp0_0080s2  ONLINE    0    0    0

errors: No known data errors
```

- 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 ZFS によって使用中であることが示されます。

```
# vxdisk list
```

```
. . .
```

```
tagmastore-usp0_0079 auto:ZFS - - ZFS
```

```
tagmastore-usp0_0080 auto:ZFS - - ZFS
```

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、`dmp_native_support` 属性が ON と表示されます。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 ラベルを削除すると、ネーティブマルチパスが有効になります。

■ 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限り、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。

```
# vxdmpadm gettune dmp_native_support
```

- 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ボリュームを表示します。ZFS で利用可能なボリュームはタイプ (TYPE) が `auto:none` と表示されます。ZFS によってすでに使われているボリュームはタイプ (TYPE) が `auto:ZFS` と表示されます。

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、再ブートやパッケージのアップグレードの前後で変化しません。

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、個別のデバイスを ZFS による制御から削除できます。

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
- DMP とネーティブマルチパスの共存
- `vxddmpadm` 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、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 コントローラの管理、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の管理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。

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 を有効または無効にする操作をサポートしません。

管理操作が終了したら、`vxddmpadm`を使って、HBA コントローラを介したパスを再び有効にしてください。

p.72 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.73 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.43 の「`vxddmpadm` 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 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

- ◆ `vxdisk 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。

```
# vxdisk path

SUBPATH      DANAME      DMNAME      GROUP      STATE
c1t0d0s2     c1t0d0s2    mydg01      mydg       ENABLED
c4t0d0s2     c1t0d0s2    mydg01      mydg       ENABLED
c1t1d0s2     c1t1d0s2    mydg02      mydg       ENABLED
c4t1d0s2     c1t1d0s2    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、c2t0d0s2 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 c2t0d0s2
```

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      c2t0d0
devicetag   c2t0d0
type        sliced
hostid      sys1
.
.
.
Multipathing information:
numpaths:   2
c2t0d0s2    state=enabled   type=primary
c1t0d0s2    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。

[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起きている (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、eva4k6k0_6 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=eva4k6k0_6
```

vxddmpadm getsubpaths コマンドの通常の出例は、次のとおりです。

NAME	STATE [A]	PATH-TYPE [M]	CTLR-NAME	ENCLR-TYPE	ENCLR-NAME	ATTRS
c0t50001FE1500A8F08d7s2	ENABLED (A)	PRIMARY	c0	EVA4K6K	eva4k6k0	-
c0t50001FE1500A8F09d7s2	ENABLED (A)	PRIMARY	c0	EVA4K6K	eva4k6k0	-
c0t50001FE1500A8F0Cd7s2	ENABLED	SECONDARY	c0	EVA4K6K	eva4k6k0	-
c0t50001FE1500A8F0Dd7s2	ENABLED	SECONDARY	c0	EVA4K6K	eva4k6k0	-

DMP ノードのカスタム名の設定

DMP 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。DMP ノード名は DMP の名前付け方によってデバイス名から生成されます。

p.16 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付け方によ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 コマンドの出例では、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もできます。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。

ファイルから DMP ノードを割り当てるには

- 1 スクリプト `vxgetdmpnames` を使って、設定内のデバイスからポピュレートされたサンプルファイルを取得します。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。
- 2 名前を割り当てるには、次のコマンドを使います。

```
# 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 OSN 名または EBN 名を使うには、次のコマンドを使います。

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

DMP とネーティブマルチパスの共存

DMP (Dynamic Multi-Pathing) では、RAW デバイスでのマルチパス化の使用がサポートされます。 `dmp_native_multipathing` チューニングパラメータがこの動作を制御します。 `dmp_native_multipathing` チューニングパラメータが `on` に設定されている場合は、DMP によって、RAW デバイスパス上で送信される `open`、`close`、`ioctl` などの操作の I/O 要求が傍受されます。

`dmp_native_multipathing` チューニングパラメータが `off` に設定されている場合、これらの要求は RAW デバイスに直接送信されます。A/PF アレイでは、Solaris プラットフォームの `format` コマンドによって、バスパスの追加の属性 (ベンダー ID、プロダクト ID、ジオメトリ情報など) が表示されません。この問題を回避するには、`dmp_native_multipathing` チューニングパラメータを有効にします。DMP によって要求が傍受され、プライマリパス上にルーティングされます。

A/P アレイでは、`dmp_native_multipathing` 機能を有効にすると、侵入なしでコマンドが成功するようになります。この機能は、A/A または A/A-A アレイには利点がありません。

次のツールのいずれかで、すでにマルチパスを管理している場合は、DMP のネーティブマルチパスを有効にしないでください。

- EMC PowerPath
- Sun StorEdge Traffic Manager (MPxIO と呼ばれる)

EMC PowerPath が最初にインストールされている場合、`dmp_native_multipathing` を `on` に設定するコマンドは失敗します。VxVM が最初にインストールされている場合は、EMC PowerPath をインストールする前に、`dmp_native_multipathing` が `off` に設定されていることを確認します。

vxdmpadm を使った DMP の管理

vxdmpadm ユーティリティは、DMP に対するコマンドライン管理インターフェースです。

vxdmpadm 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.44 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.45 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.47 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.47 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.50 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.51 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.51 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。
- 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.52 の「[TPD 制御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.53 の「[拡張デバイス属性の表示](#)」を参照してください。
- p.55 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.56 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.62 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.63 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.64 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.65 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.72 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.74 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.75 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.76 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.79 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 Modules) の設定
p.81 の「アレイポリシーモジュール (Array Policy Modules) 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.133 の「DMP チューニングパラメータ」を参照してください。

以降の項では、出力例を示してこれらのタスクを詳しく説明します。

vxddmpadm (1M) 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 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/rdsdsk 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=c0t5006016041E03B33d0s2
```

```
NAME                STATE    ENCLR-TYPE    PATHS ENBL DSBL ENCLR-NAME
=====
emc_clariion0_16  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。

```
# vxdmpadm -v getdmpnode nodename=c0t5006016041E03B33d0s2

NAME          STATE  ENCLR-TYPE  PATHS ENBL DSBL ENCLR-NAME  SERIAL-NO  ARRAY_VOL_ID
=====
emc_clariion0_16  ENABLED EMC_CLARiION 6    6    0  emc_clariion0 600601606 16
```

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。

```
# vxdmpadm getdmpnode enclosure=enc0

NAME          STATE  ENCLR-TYPE  PATHS  ENBL  DSBL  ENCLR-NAME
=====
c2t1d0s2  ENABLED  T300        2      2      0    enc0
c2t1d1s2  ENABLED  T300        2      2      0    enc0
c2t1d2s2  ENABLED  T300        2      2      0    enc0
c2t1d3s2  ENABLED  T300        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。

```
# vxdmpadm 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

`vxdmpadm list dmpnode` コマンドは DMP の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せん。

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。

```
# vxdmpadm 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。

```
# vxdmpadm list dmpnode enclosure=enclosure name
```

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。

```
# vxdmpadm 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 = 600601606D121B008FB6E0CA8EDBDB11
udid = DGC%5FDISK%5FCK200070400359%5F600601606D121B008FB6E0CA8EDBDB11
dev-attr = lun
###path = name state type transport ctrlr hwpath aprotID aprotWWN attr
path = c0t5006016141E03B33d1s2 enabled(a) primary FC c0
/pci@1e,60000/SUNW,emlxs@3/fp@0,0 A5 50:06:01:61:41:e0:3b:33 -
path = c0t5006016041E03B33d1s2 enabled(a) primary FC c0
/pci@1e,60000/SUNW,emlxs@3/fp@0,0 A4 50:06:01:60:41:e0:3b:33 -
path = c0t5006016841E03B33d1s2 enabled secondary FC c0
/pci@1e,60000/SUNW,emlxs@3/fp@0,0 B4 50:06:01:68:41:e0:3b:33 -
path = c1t5006016141E03B33d1s2 enabled(a) primary FC c1
/pci@1e,60000/SUNW,emlxs@3,1/fp@0,0 A5 50:06:01:61:41:e0:3b:33 -
path = c1t5006016841E03B33d1s2 enabled secondary FC c1
/pci@1e,60000/SUNW,emlxs@3,1/fp@0,0 B4 50:06:01:68:41:e0:3b:33 -
path = c1t5006016041E03B33d1s2 enabled(a) primary FC c1
/pci@1e,60000/SUNW,emlxs@3,1/fp@0,0 A4 50:06:01:60:41: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 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=c11t0d10s2
```

上記のコマンドを実行すると、次のような出力が表示されます。

NAME	STATE	ENCLR-TYPE	PATHS	ENBL	DSBL	ENCLR-NAME
c11t0d8s2	ENABLED	ACME	2	2	0	enc1
c11t0d9s2	ENABLED	ACME	2	2	0	enc1
c11t0d10s2	ENABLED	ACME	2	2	0	enc1
c11t0d11s2	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 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 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
c1t65d0s2	ENABLED (A)	-	Disk_1	Disk	c1	-
c1t66d0s2	ENABLED (A)	-	Disk_2	Disk	c1	-
c2t65d0s2	ENABLED (A)	-	Disk_1	Disk	c2	-
c2t66d0s2	ENABLED (A)	-	Disk_2	Disk	c2	-
c3t2d0s2	ENABLED (A)	-	EMC0_1	EMC0	c3	-
c3t2d1s2	ENABLED (A)	-	EMC0_2	EMC0	c3	-
c4t2d0s2	ENABLED (A)	-	EMC0_1	EMC0	c4	-
c4t2d1s2	ENABLED (A)	-	EMC0_2	EMC0	c4	-

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

```
# vxddmpadm getsubpaths dmpnodename=c2t66d0s2
```

NAME	STATE [A]	PATH-TYPE [M]	CTLR-NAME	ENCLR-TYPE	ENCLR-NAME	ATTRS
------	-----------	---------------	-----------	------------	------------	-------

```
=====
c2t66d0s2 ENABLED (A) PRIMARY          c2          ACME          enc0          -
c1t66d0s2 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.65 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 [-]  CTRLR-NAME  ENCLR-TYPE  ENCLR-NAME  ATTRS
=====
c2t1d0s2     ENABLED (A)  PRIMARY        c2t1d0s2    ACME         enc0         -
c2t2d0s2     ENABLED (A)  PRIMARY        c2t2d0s2    ACME         enc0         -
c2t3d0s2     DISABLED     SECONDARY      c2t3d0s2    ACME         enc0         -
c2t4d0s2     ENABLED     SECONDARY      c2t4d0s2    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、エンクロージャの名前とアレイポート 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。

```
# 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=HDS9500-ALUA0 portid=1A

NAME          STATE [A]    PATH-TYPE [M]  DMPNODENAME  ENCLR-NAME  CTRLR  ATTRS
=====
c1t65d0s2     ENABLED (A)  PRIMARY        c1t65d0s2    HDS9500-ALUA0  c1     -
c1t66d0s2     ENABLED (A)  PRIMARY        c1t66d0s2    HDS9500-ALUA0  c1     -
=====
```

たとえば、WWN を通じ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。

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

NAME	STATE [A]	PATH-TYPE [M]	DMPNODENAME	ENCLR-NAME	CTLR	ATTRS
c1t65d0s2	ENABLED (A)	PRIMARY	c1t65d0s2	HDS9500-ALUA0	c1	-
c1t66d0s2	ENABLED (A)	PRIMARY	c1t66d0s2	HDS9500-ALUA0	c1	-

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せん。

```
# vxdmpadm 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。

```
# vxdmpadm getsubpaths enclosure=Disk
```

NAME	STATE [A]	PATH-TYPE [M]	DMPNODENAME	ENCLR-NAME	CTLR	ATTRS
c1t65d0s2	ENABLED (A)	-	Disk_1	Disk	c1	-
c1t66d0s2	ENABLED (A)	-	Disk_2	Disk	c1	-
c2t65d0s2	ENABLED (A)	-	Disk_1	Disk	c2	-
c2t66d0s2	ENABLED (A)	-	Disk_2	Disk	c2	-

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。

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

NAME	STATE [A]	PATH-TYPE [M]	DMPNODENAME	ENCLR-NAME	CTLR	ATTRS
c1t65d0s2	ENABLED (A)	-	Disk_1	Disk	c1	-
c1t66d0s2	ENABLED (A)	-	Disk_2	Disk	c1	-

デフォルトでは、vxdmpadm 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。

```
# vxdmpadm -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]
```

コントローラに関する情報の表示

次の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。

```
# vxdmpadm listctlr all
```

CTLR-NAME	ENCLR-TYPE	STATE	ENCLR-NAME
c1	OTHER	ENABLED	other0
c2	X1	ENABLED	jbod0
c3	ACME	ENABLED	enc0
c4	ACME	ENABLED	enc0

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、vxdmpadm 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。

```
# vxdmpadm listctlr enclosure=enc0
```

または

```
# vxdmpadm listctlr type=ACME
```

CTLR-NAME	ENCLR-TYPE	STATE	ENCLR-NAME
c2	ACME	ENABLED	enc0
c3	ACME	ENABLED	enc0

vxdmpadm 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

特定のエンクロージャの属性(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 数など)を表示するには、次のコマンドを使います。

```
# vxddmpadm listenclosure enc0
```

```
ENCLR_NAME ENCLR_TYPE  ENCLR_SNO          STATUS  ARRAY_TYPE  LUN_COUNT
=====
enc0       T3          60020f20000001a90000  CONNECTED  A/P          30
```

次のコマンドを実行すると、システム内のすべてのエンクロージャの属性が一覧表示されます。

```
# vxddmpadm listenclosure all
```

```
ENCLR_NAME ENCLR_TYPE  ENCLR_SNO          STATUS  ARRAY_TYPE  LUN_COUNT
=====
Disk       Disk        DISKS              CONNECTED  Disk         6
SENA0     SENA        508002000001d660  CONNECTED  A/A          57
enc0      T3          60020f20000001a90000  CONNECTED  A/P          30
```

A/P または ALUA アレイが MPxIO の管理下であれば、DMP は A/A モードのデバイスを要求します。上のコマンドの出力は A/A. として ARRAY_TYPE を示したものです。MPxIO 制御の下のアレイのために、DMP は A/P 特定の属性か ALUA 特定の属性を格納しません。これらの属性には、プライマリパスまたはセカンダリパス、ポートのシリアル番号、アレイコントローラ ID が含まれます。

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。

メモ: DMP はネイティブマルチパスドライバによって制御される LUN のアレイポートに関する情報は報告しません。DMP は `dmp_monitor_fabric` チューニングパラメータが有効で、イベントソースデーモン (esd) が実行されているときのみ、pWWN 情報を報告します。

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 c2t66d0s2 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。

```
# vxdmpadm getportids dmpnodename=c2t66d0s2
NAME          ENCLR-NAME  ARRAY-PORT-ID  pWWN
=====
c2t66d0s2    HDS9500V0   1A              20:00:00:E0:8B:06:5F:19
```

TPD 制御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 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。

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

p.106 の「[TPD 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
emcpower10s2    auto:sliced   disk1         ppdg          online
emcpower11s2    auto:sliced   disk2         ppdg          online
emcpower12s2    auto:sliced   disk3         ppdg          online
emcpower13s2    auto:sliced   disk4         ppdg          online
emcpower14s2    auto:sliced   disk5         ppdg          online
emcpower15s2    auto:sliced   disk6         ppdg          online
emcpower16s2    auto:sliced   disk7         ppdg          online
```

```
emcpower17s2    auto:sliced    disk8          ppdg          online
emcpower18s2    auto:sliced    disk9          ppdg          online
emcpower19s2    auto:sliced    disk10         ppdg          online
```

次のコマンドを実行すると、DMP が PowerPath 制御のノード emcpower10s2 に対応するパスを検出し、表示します。

```
# vxdmpadm getsubpaths tpdnodename=emcpower10s2
NAME      TPDNODENAME  PATH-TYPE [-] DMP-NODENAME  ENCLR-TYPE  ENCLR-NAME
=====
c7t0d10s2emcpower10s2-          emcpower10s2  EMC          EMC0
c6t0d10s2emcpower10s2-          emcpower10s2  EMC          EMC0
```

逆に次のコマンドは、DMP が、パス c7t0d10s2 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。

```
# vxdmpadm gettpdnode nodename=c7t0d10s2
NAME      STATE      PATHS  ENCLR-TYPE  ENCLR-NAME
=====
emcpower10s2  ENABLED    2      EMC          EMC0
```

拡張デバイス属性の表示

デバイス検出層 (DDL) 拡張属性は、DDL によって検出される VxVM、DMP LUN、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。

これらのカテゴリの一覧には、次のものが含まれます。

ハードウェア RAID の種類	LUN が属するストレージ RAID グループの種類を表示します。
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種類	メディアの種類、つまり SSD (半導体ディスク) 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 の種類 (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
```

vxdisk -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。

```
# vxdisk -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。

```
# vxdisk -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。

<http://www.symantec.com/docs/TECH170013>

メモ: DMP では、ネイティブマルチパスドライバによって制御されている LUN の拡張属性はサポートされません。

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 制御から除外されます。デバイスは、無効化されると DMP 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。

```
# vxdmpadm exclude { all | product=VID:PID |  
ctrlr=[¥!]ctrlrname | dmpnodename=diskname [ path=[¥!]pathname] }  
  
# vxdmpadm include { all | product=VID:PID |  
ctrlr=[¥!]ctrlrname | dmpnodename=diskname [ path=[¥!]pathname] }
```

各オプションの説明

all - すべてのデバイス

product=VID:PID - 指定した VID:PID を持つすべてのデバイス

ctrlr=ctrlrname - 指定したコントローラ上のすべてのデバイス

dmpnodename=diskname - DMP ノード下のすべてのデバイス

dmpnodename=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

vxdmpadm 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、コントローラについての I/O 統計情報を収集および表示することができます。

へ統計の収集を有効にするためには、このコマンドを入力してください:

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

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。

```
# vxdmpadm iostat reset
```

memory 属性を使うと、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できます。デフォルトでは、各 CPU に対して 32k(32 KB)に制限されます。

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。

```
# vxdmpadm iostat show {all | ctrlr=ctrlr-name ¥  
| dmpnodename=dmp-node ¥  
| enclosure=enclr-name [portid=array-portid] ¥  
| pathname=path-name | pwwn=array-port-wwn } ¥  
[interval=seconds [count=N]]
```

このコマンドを実行すると、すべてのパス(all)についての I/O 統計情報か、指定したコントローラ、DMP ノード、エンクロージャ、パス、ポート ID についての I/O 統計情報が表示されます。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

利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。

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ことがあります。

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。

```
# vxdmpadm iostat stop
```

vxdmpadm iostat コマンドの使用例

次に、vxdmpadm iostat コマンドを使ったセッション例を示します。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。

```
# vxdmpadm 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、すべてのバスに関する現在の統計情報が表示されます。

```
# vxdmpadm -u k iostat show all
                                cpu usage = 7952us      per cpu memory = 8192b
OPERATIONS                      BYTES                      AVG TIME(ms)
PATHNAME  READS  WRITES  READS  WRITES  READS  WRITES
c0t0d0    1088    0    557056k    0    0.00    0.00
c2t118d0   87     0    44544k    0    0.00    0.00
c3t118d0   0     0     0        0    0.00    0.00
c2t122d0   87     0    44544k    0    0.00    0.00
c3t122d0   0     0     0        0    0.00    0.00
c2t115d0   87     0    44544k    0    0.00    0.00
c3t115d0   0     0     0        0    0.00    0.00
c2t103d0   87     0    44544k    0    0.00    0.00
c3t103d0   0     0     0        0    0.00    0.00
c2t102d0   87     0    44544k    0    0.00    0.00
c3t102d0   0     0     0        0    0.00    0.00
c2t121d0   87     0    44544k    0    0.00    0.00
c3t121d0   0     0     0        0    0.00    0.00
c2t112d0   87     0    44544k    0    0.00    0.00
c3t112d0   0     0     0        0    0.00    0.00
c2t96d0    87     0    44544k    0    0.00    0.00
c3t96d0    0     0     0        0    0.00    0.00
c2t106d0   87     0    44544k    0    0.00    0.00
c3t106d0   0     0     0        0    0.00    0.00
```

```

c2t113d0      87      0      44544k      0      0.00      0.00
c3t113d0      0      0      0      0      0.00      0.00
c2t119d0      87      0      44544k      0      0.00      0.00
c3t119d0      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ジ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=c3t115d0s2
                cpu usage = 8132us      per cpu memory = 4096b
                OPERATIONS                BYTES                AVG TIME (ms)
PATHNAME  READS  WRITES  READS  WRITES  READS  WRITES
c3t115d0s2  0      0      0      0      0.00  0.00
```

```
# vxddmpadm -u k iostat show dmpnodename=c0t0d0s2
                cpu usage = 8501us      per cpu memory = 4096b
                OPERATIONS                BYTES                AVG TIME (ms)
PATHNAME  READS  WRITES  READS  WRITES  READS  WRITES
c0t0d0s2  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
c0t0d0s2  1088      0  57056k      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 dmpnodename=emc_clariion0_342 interval=1 count=2
                cpu usage = 164687us    per cpu memory = 409600b
                OPERATIONS                BLOCKS                AVG TIME (ms)
PATHNAME  READS  WRITES  READS  WRITES  READS  WRITES
c0t5006016041E03B33d6s2  3      0      33      0      0.02  0.00
c0t5006016141E03B33d6s2  3      0      3      0      0.16  0.00
c0t5006016841E03B33d6s2  0      0      0      0      0.00  0.00
c1t5006016041E03B33d6s2  1      0      16      0      0.02  0.00
c1t5006016141E03B33d6s2  2      0      2      0      0.18  0.00
```

```

c1t5006016841E03B33d6s2 0      0      0      0      0.00      0.00

c0t5006016041E03B33d6s2 0      0      0      0      0.00      0.00
c0t5006016141E03B33d6s2 0      0      0      0      0.00      0.00
c0t5006016841E03B33d6s2 0      0      0      0      0.00      0.00
c1t5006016041E03B33d6s2 0      0      0      0      0.00      0.00
c1t5006016141E03B33d6s2 0      0      0      0      0.00      0.00
c1t5006016841E03B33d6s2 0      0      0      0      0.00      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 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` コマンドを使用します。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=c5t2d1s2
```

```

                QUEUED I/Os      Pending I/Os
DMPNODENAME  READS  WRITES
c5t2d1s2      2     15             30

```

DMP ノード、パスまたはコントローラ上でエラーが返された I/O の件数を表示するには、次のコマンドを使います。

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

たとえば、パス上でエラーが返された I/O の件数を表示するには、次のようなコマンドを使います。

```
# vxddmpadm -e iostat show pathname=c1t5006016041E03B33d6s2 interval=1

cpu usage = 168144us      per cpu memory = 409600b
                        ERROR I/Os

```

PATHNAME	READS	WRITES
c1t5006016041E03B33d6s2	0	0
c1t5006016041E03B33d6s2	0	0

累積 I/O 統計情報の表示

累積 I/O 統計情報を DMP ノード、コントローラ、アレイポート ID、またはホストアレイコントローラのペアとエンクロージャごとに表示するには、`vxddmpadm iostat` コマンドの `groupby` 節を使います。 `groupby` 節を指定しない場合、統計はパスごとに表示されません。

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。

<code>h</code> または <code>H</code>	スループットを最も大きい単位で表示します。
<code>k</code>	スループットをキロバイト単位で表示します。
<code>m</code>	スループットをメガバイト単位で表示します。
<code>g</code>	スループットをギガバイト単位で表示します。
<code>bytes</code> <code>b</code>	スループットを正確なバイト数で表示します。
<code>us</code>	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]
```

次に例を示します。

```
# vxddmpadm iostat show groupby=dmpnode dmpnodename=c5t0d1s2
```

DMPNODENAME	OPERATIONS		BLOCKS		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
c5t0d1s2	0	0	0	0	0.00	0.00

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=ctrl ctrl=c5
```

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

アレイド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
```

ENCLRNAME	OPERATIONS		BLOCKS		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
EMC_CLARiION	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効です。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=c2t16d4s2
```

PATHNAME	OPERATIONS		BLOCKS		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
c3t16d4s2	10	110	2	25	12.00	27.96
c2t17d4s2	20	126	4	29	9.50	19.41

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=c2t17d4s2
```

PATHNAME	OPERATIONS		BLOCKS		AVG TIME (microsec)	
	READS	WRITES	READS	WRITES	READS	WRITES
c2t17d4s2	20	126	4	29	9500.00	19413.79

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。

パスに対して設定された属性は永続的で、/etc/vx/ddmpolicy.info ファイルに格納されます。

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 c2t10d0s2 pathtype=active
```

nomanual パスのもと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 c3t10d0s2 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 c1t20d0s2 ¥
pathtype=nopreferred
```

preferred [priority=N]	<p>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> <p>p.65 の「I/O ポリシーの指定」を参照してください。</p> <p>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。</p> <pre># vxddmpadm setattr enclosure enc0 ¥ iopolicy=priority # vxddmpadm setattr path clt20d0s2 ¥ pathtype=preferred priority=2</pre>
primary	<p>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。</p> <pre># vxddmpadm setattr path c3t10d0s2 ¥ pathtype=primary</pre>
secondary	<p>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。</p> <pre># vxddmpadm setattr path c4t10d0s2 ¥ pathtype=secondary</pre>
standby	<p>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。</p> <pre># vxddmpadm setattr path c2t10d0s2 ¥ pathtype=standby</pre>

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=encl_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 `vxdmpadm getattr` コマンドを使います。

```
# vxdmpadm 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。

```
# vxdmpadm 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 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した値は `dmpolicy.info` ファイルに格納され、永続的に保持されます。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、`vxdmpadm 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 `vxdmpadm setattr` コマンドを実行します。

```
# vxdmpadm 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。

```
# vxdmpadm 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、`vxdmpadm 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。

```
# vxdmpadm getattr enclosure Disk iopolicy
ENCLR_NAME          DEFAULT          CURRENT
-----
Disk                MinimumQ        Balanced
```

分散 I/O ポリシーでパーティションサイズ 2 MB に設定されているエンクロージャ `enc0` のパーティションサイズ (`partitionsizes`) を表示するには、次のコマンドを実行します。

```
# vxdmpadm getattr enclosure enc0 partitionsize
ENCLR_NAME          DEFAULT          CURRENT
-----
enc0                512              4096
```

I/O ポリシーの指定

`vxdmpadm 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 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。

警告: I/O ポリシーは `/etc/vx/dmppolicy.info` ファイルに記録され、システムを再ブートしても保持されます。

このファイルは編集しないでください。

設定できるポリシーは次のとおりです。

adaptive

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。

たとえば、エンクロージャ enc1 に対して I/O ポリシーを adaptive に設定するには、次のコマンドを実行します。

```
# vxddmpadm setattr enclosure enc1 ¥  
  iopolicy=adaptive
```

adaptiveminq

I/O が各パスの I/O キューの長さによってスケジューリングされること以外は adaptive 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。

```
balanced  
[partitionsizesize]
```

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 500KB 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。

`partitionsizesize` 属性に `size` 引数を使って、パーティションサイズを指定することができます。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。

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。

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。

デフォルト値を変更するには、`dmp_pathswitch_blks_shift` チューニングパラメータの値を調整します。

p.133 の「[DMP チューニングパラメータ](#)」を参照してください。

メモ: 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。

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。

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 `balanced` 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。

```
# vxddmpadm setattr enclosure enc0 ¥  
  iopolICY=balanced partitionsizesize=4096
```

minimumq

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 で自動的に判別されるため、設定を変更することはできません。

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。

```
# vxddmpadm setattr enclosure Disk ¥  
  iopolicy=minimumq
```

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。

priority

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.62 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。

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。

```
# vxddmpadm setattr arrayname SENA ¥  
  iopolicy=priority
```

ラウンドロビン

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。

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。

```
# vxddmpadm setattr arraytype A/A ¥  
  iopolicy=round-robin
```

singleactive

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。

たとえば、JBOD ディスクに対して I/O ポリシーを singleactive に設定するには、次のコマンドを実行します。

```
# vxdmpadm setattr arrayname Disk ¥
    iopolicy=singleactive
```

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限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。

```
# vxdmpadm 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 vxdmpadm gettattr コマンドに指定します。

```
# vxdmpadm 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 環境において負荷分散を設定する方法について説明します。

vxdisk 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 c3t2d15s2 には 8 つのプライマリパスがあります。

```
# vxdisk list c3t2d15s2

Device: c3t2d15s2
.
.
.
numpaths: 8
c2t0d15s2 state=enabled type=primary
c2t1d15s2 state=enabled type=primary
c3t1d15s2 state=enabled type=primary
c3t2d15s2 state=enabled type=primary
c4t2d15s2 state=enabled type=primary
c4t3d15s2 state=enabled type=primary
c5t3d15s2 state=enabled type=primary
c5t4d15s2 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。

```
# vxmpadm 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 vxmpadm iostat コマンドを実行すると、すべての I/O が 1 つのパス c5t4d15s2 に対して行われていることがわかります。

```
# vxmpadm iostat show dmpnodename=c3t2d15s2 interval=5 count=2
.
.
.
```

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

PATHNAME	OPERATIONS		KBYTES		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
c2t0d15s2	0	0	0	0	0.00	0.00
c2t1d15s2	0	0	0	0	0.00	0.00
c3t1d15s2	0	0	0	0	0.00	0.00
c3t2d15s2	0	0	0	0	0.00	0.00
c4t2d15s2	0	0	0	0	0.00	0.00
c4t3d15s2	0	0	0	0	0.00	0.00
c5t3d15s2	0	0	0	0	0.00	0.00
c5t4d15s2	10986	0	5493	0	0.41	0.00

次の vxdmpadm 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。

```
# vxdmpadm 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。

```
# vxdmpadm setattr enclosure ENC0 iopolicy=round-robin
# vxdmpadm 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。

```
# vxdmpadm 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。

```
# vxdmpadm iostat show dmpnodename=c3t2d15s2 interval=5 count=2
```

```
.
.
.
```

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

PATHNAME	OPERATIONS		KBYTES		AVG TIME (ms)	
	READS	WRITES	READS	WRITES	READS	WRITES
c2t0d15s2	2041	0	1021	0	0.39	0.00
c2t1d15s2	1894	0	947	0	0.39	0.00
c3t1d15s2	2008	0	1004	0	0.39	0.00
c3t2d15s2	2054	0	1027	0	0.40	0.00
c4t2d15s2	2171	0	1086	0	0.39	0.00
c4t3d15s2	2095	0	1048	0	0.39	0.00
c5t3d15s2	2073	0	1036	0	0.39	0.00
c5t4d15s2	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。

```
# vxdmpadm 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 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。それらのパスから発行された保留中の I/O 要求が完了するまで、コマンドはブロックされます。要求が終了するまで、ブロックされます。

システムボードを切断する前に、ボード上に置かれている HBA コントローラに対するすべての I/O を停止します。この操作を行うには、`vxdmpadm disable` コマンドを実行し、続いて Oracle 社が提供する動的再設定 (DR) 機能を実行します。

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。

```
# vxdmpadm [-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。

```
# vxdmpadm [-c|-f] disable ctrlr=ctrlr_name1[,ctrlr_name2,ctrlr_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。

```
# vxdmpadm [-c|-f] disable enclosure=enclr_name portid=array_port_ID
```

```
# vxdmpadm [-c|-f] disable pwnn=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=ctrlr_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す。この場合、`-f` オプションを使ってそのパスを強制的に無効にしない限りは、`disable` コマンドの実行はエラーになり、エラーメッセージが表示されます。

`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。

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 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。また、`enable` 操作を行うと、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して I/O を行うことが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 ctlr=ctlr_name1[,ctlr_name2,ctlr_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 ctlr=ctlr_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 enc0 name=GRP1
```

この例では、エンクロージャ名を enc0 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 listenenclosure all
ENCLR_NAME      ENCLR_TYPE      ENCLR_SNO      STATUS
=====
other0          OTHER           OTHER_DISKS     CONNECTED
```

jbod0	X1	X1_DISKS	CONNECTED
GRP1	ACME	60020f20000001a90000	CONNECTED

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 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 要求を単一のパスに対し最大 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.78 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する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。

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

recoveryoption=default を指定すると、DMP を recoveryoption=fixedretry retrycount=5 に対応するデフォルト設定にリセットします。次に例を示します。

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

上記のコマンドを実行すると、I/O 調整がデフォルト設定されるという影響もあります。

p.76 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 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、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、vxdmpadm getattr コマンドを使います。

p.78 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、vxdmpadm setattr コマンドを次の形式で使います。

```
# vxdmpadm 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。

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

vxdmpadm 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。

```
# vxdmpadm 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 ドライバ内でより多くの 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。

```
# vxdmpadm 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。

```
# vxdmpadm 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.75 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。

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。

```
# vxdmpadm 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。デフォルトは 1 です。

```
# vxdmpadm settune dmp_sfg_threshold=N
```

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。

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。

```
# vxdmpadm getportids {ctrl=ctrl_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 に設定されています。

```
# 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]   Timebound[60]
HDS9500-ALUA0  Error-Retry       Fixed-Retry[5]  Timebound[20]
```

これは、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 3-1 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 3-1 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 3-2 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 3-2 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 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スレッドを保守します。パスに対して実行される分析のタイプは、設定されたチェックポリシーに応じて異なります。

メモ: DMP パスリストスレッドでは、`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.81 の「[DMP パスリストスレッドの停止](#)」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になります。

- `check_all`

パスリストスレッドは、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ためのコマンドは、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ためのコマンドは、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ためのコマンドは、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ためのコマンドは、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` を使います。

p.133 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

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 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 Modules) の設定

アレイポリシーモジュール (APM) は、アレイと組み合わせて使う、動的にロード可能なカーネルモジュール (DMP のプラグイン) です。APM では、次の操作を行うためのアレイ固有の手順とコマンドが定義されます。

- 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 または APM ベンダー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 を表示することができます。

```
# vxdatapadm 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。

```
# vxdatapadm listapm module_name
```

APM を追加および設定するには、次のコマンドを使います。

```
# vxdatapadm -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 を削除することができます。

```
# vxdatapadm -r cfgapm module_name
```

vxdatapadm (1M) 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](#)
- [VxVM の ZFS との共存](#)
- [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

Veritas Dynamic Multi-Pathing (DMP) 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.8 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` ユーティリティを使います。

p.89 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

`vxdiskconfig` ユーティリティを使うと、ホストに接続されている新しいディスクデバイス、オンラインにするディスクデバイスまたはこのホストに接続されているホストバスアダプタにゾーンされるファイバーチャネルデバイスがスキャンされ、設定されます。このコマンドは、プラットフォームに依存するインターフェースを呼び出し、新しいディスクデバイスの設定を行い、オペレーティングシステムの制御下に新しいディスクデバイスを追加します。こ

のコマンドは、DMPの設定デーモンを最後に起動した後で、追加されたディスクをスキャンします。この後、DMPが、これらの新しいディスクを動的に設定して認識します。

`vxdiskconfig`は、ディスクがホストに物理的に接続されている場合またはファイバーチャネルデバイスがホストにゾーンされている場合は、常に使われます。

`vxdiskconfig`は、`vxctl enable`を呼び出し、ボリュームのデバイスノードディレクトリの再構築と、DMPの内部データベースの更新が行われ、システムの新しい状態が反映されます。

`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。

```
# vxctl -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
```

上記のコマンドを実行すると、特有の DDI_NT_FABRIC プロパティセットを持つデバイスが検出されます。

次のコマンドでは、デバイス c1t1d0s2 と c2t2d0s2 をスキャンします。

```
# vxdisk scandisks device=c1t1d0s2,c2t2d0s2
```

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=c1t1d0s2,c2t2d0s2
```

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 ¥!ctrlr=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を検出できます。

```
# vxdisk scandisks pctrlr=/pci@1f,4000/scsi@3/
```

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。

vxdmpadm getctrlr 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

DMP 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。

場合によっては、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 はアレイ固有のアレイサポートライブラリ (ASL) 使用を要求します。ASL はアレイポリシーモジュール (APM) と共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。

<http://www.symantec.com/docs/TECH170013>

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みます。

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.97 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、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

Veritas Dynamic Multi-Pathing で認識されるディスクアレイは ASL で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.97 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。シマンテック社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。シマンテック社は VRTSaslapm パッケージ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 パッケージ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 パッケージをインストールするための、最新のダウンロード用パッケージ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 パッケージ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。

<http://www.symantec.com/docs/TECH170013>

各 VRTSaslapm パッケージは、**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 をインストールする必要があります。

VRTSaslapm パッケージ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。新しいディスクアレイ内の任意のディスクを順次接続していくと、vxconfigd を実行中であれば、vxconfigd によって、すぐに **DDL** デバイス検出機能が呼び出され、**VxVM** のデバイスリストに新しいディスクが追加されます。

最新の VRTSaslapm パッケージ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。手順について詳しくは、『**Veritas Storage Foundation and High Availability Solutions** 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および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。

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。

- ◆ 次のコマンドを入力します。

```
# vxctl enable
```

サードパーティ製ドライバの共存

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** をバイパスさせることができます。適切な **ASL** 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.106 の「TPD 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.52 の「TPD 制御デバイスに関する情報の表示」を参照してください。

EMC Symmetrix アレイの自動検出

VxVM 4.0 では、EMC Symmetrix アレイの設定方法は 2 種類ありました。

- EMC PowerPath がインストールされている環境では、EMC Symmetrix アレイは外部デバイスとして設定できました。
p.100 の「外部デバイス」を参照してください。
- EMC PowerPath がインストールされていない環境では、マルチパス化に DMP を使えました。

システムを VxVM 4.1 以降のリリースにアップグレードすると、既存の EMC PowerPath デバイスは DDL によって検出され、マルチパス化に PowerPath を使っている場合でも、DMP ノードを持つ自動設定ディスクとして DMP に設定されます。これらのアレイを外部デバイスとして設定する必要はありません。

表 4-1 に、PowerPath とともに DMP を使うシナリオを示します。

ASL はすべて、Storage Foundation 製品のインストール時にインストールされる ASL-APM パッケージに含まれています。

表 4-1 PowerPath とともに DMP を使うシナリオ

PowerPath	DMP	アレイ設定モード
インストール済み	libvxpp ASL が EMC Symmetrix アレイと DGC CLARiiON 要求を内部的に処理します。PowerPath がフェールオーバーを処理します	EMC Symmetrix - 任意 DGC CLARiiON - アクティブ/パッシブ (A/P)、明示的フェールオーバーモードのアクティブ/パッシブ (A/P-F)、および ALUA 明示的フェールオーバー
未インストールで、アレイが EMC Symmetrix	DMP がマルチパス化を処理します ASL 名は libvxemc です。	アクティブ/アクティブ

PowerPath	DMP	アレイ設定モード
未インストールで、アレイが DGC CLARiioN (CXn00)	DMP がマルチパス化を処理します ASL 名は libvxCLARiioN です。	アクティブ/パッシブ (A/P)、明示的フェールオーバーモードのアクティブ/パッシブ (A/P-F)、および ALUA

EMCpower ディスクが外部ディスクとして設定されている場合は、次の例に示すように、`vxddladm rmforeign` コマンドを使って外部定義を削除します。

```
# vxddladm rmforeign blockpath=/dev/dsk/emcpower10 ¥
charpath=/dev/rdisk/emcpower10
```

DMP が正確な照会データを受信できるようにするには、Symmetrix Director パラメータの共通シリアル番号 (C-bit) を有効に設定する必要があります。

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、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) カテゴリ内のサポート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

- ◆ 次のコマンドを入力します。

```
# vxddladm list
```

出力例を次に示します。

```
HBA c2 (20:00:00:E0:8B:19:77:BE)
  Port c2_p0 (50:0A:09:80:85:84:9D:84)
    Target c2_p0_t0 (50:0A:09:81:85:84:9D:84)
      LUN c2t0d0s2
  . . .
HBA c3 (iqn.1986-03.com.sun:01:0003ba8ed1b5.45220f80)
  Port c3_p0 (10.216.130.10:3260)
    Target c3_p0_t0 (iqn.1992-08.com.netapp:sn.84188548)
      LUN c3t0d0s2
      LUN c3t0d1s2
    Target c3_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についての情報を入手できます。取得される情報は次のとおりです。

ドライバ(Driver)	HBA を制御するドライバ。
ファームウェア(Firmware)	ファームウェアのバージョン。
検出(Discovery)	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

- ◆ 次のコマンドを使い、iSCSI デバイスを含めて、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
```

PortID	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。取得される情報は次のとおりです。

エイリアス(Alias)	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(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。

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。

```
TgtID      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

- ◆ 次のコマンドを使えば、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
```

```
TgtID      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。取得される情報は次のとおりです。

Device	デバイス名。
Target-ID	親ターゲット。
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。
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)
-----
c2t0d2s2   c2_p0_t0   Online    CLAIMED (libvxemc.so)
c3t1d2s2   c3_p0_t1   Online    SKIPPED (libvxemc.so)
c4t1d2s2   c4_p0_t1   Offline   ERROR
c4t1d2s2   c4_p0_t2   Online    EXCLUDED
c4t5d2s2   c4_p0_t5   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 デバイスパ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 4-2 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
FirstBurstLength	65535	512	16777215
InitialR2T	yes	no	yes
ImmediateData	yes	no	yes

パラメータ	デフォルト値	最小値	最大値
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。

- ◆ 次のようにコマンドを入力します。

```
# 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=libvxemc.so
```

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
```

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、次の例に示すように、`includearray` キーワードを使ってエクスクルードリストからエンタリを削除できます。

```
# vxddladm includearray libname=libvxemc.so
```

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。vxconfigd が実行されている場合、vxdisk scandisks コマンドを使ってアレイを検出し、詳細をデータベースに追加できます。

無効にされたディスクアレイの一覧表示

現在 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、次の作業を実行します。

- ◆ 次のようにコマンドを入力します。

```
# vxddladm listexclude
```

DISKS カテゴリ内のサポートされているディスクの一覧表示

DISKS (JBOD) カテゴリ内のサポートされているディスクを一覧表示するには、次の作業を実行します。

- ◆ 次のようにコマンドを入力します。

```
# vxddladm listjbod
```

サポートされるアレイライブラリに関する詳細の表示

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

ASL は、ディレクトリ /etc/vx/lib/discovery.d にあります。

サポートされるアレイライブラリに関する詳細を表示するには、次の作業を実行します

- ◆ 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)、アレイのプロダクト 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、指定されていない限り、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.8 の「**DMP の動作方法**」を参照してください。

警告:ここで説明する手順は、**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手順です。この操作を行わないと、**Veritas Volume Manager** 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 引数に対する引数と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、vxctl enable コマンドを使います。

```
# vxctl enable
```

p.87 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 `vxddladm 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。

```
# vxddladm listenclosure
ENCLR_NAME ENCLR_TYPE ENCLR_SNO STATUS ARRAY_TYPE LUN_COUNT
=====
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、`vxddladm list` コマンドを使います。

```
# vxddladm list
DEVICE     TYPE        DISK        GROUP        STATUS
Disk_0     auto:none   -           -           online invalid
Disk_1     auto:none   -           -           online invalid
...
```

- 8 DMP のパスが認識されたことを確認するには、`vxddladm getdmpnode` 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。

```
# vxddladm getdmpnode enclosure=Disk
NAME      STATE    ENCLR-TYPE PATHS ENBL DSBL ENCLR-NAME
=====
Disk_0    ENABLED  Disk        2     2     0     Disk
Disk_1    ENABLED  Disk        2     2     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。

`vxddladm(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。

ディスクを DISKS カテゴリから削除するには

- ◆ `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 では、マルチパス機能や RAM ディスク機能を提供するデバイスなど、サードパーティ製のドライバで制御される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。これらのデバイスに対しては、DMP ではなく、サードパーティ製の一部のアレイドライバに備えられているマルチパス機能を使うことをお勧めします。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 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
```

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。

外部デバイス機能は、RAM ディスクなどの標準的ではないデバイス、一部の半導体ディスク、EMC PowerPath などの仮想デバイスをサポートするため、VxVM 4.0 で導入されました。

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 の制御下でないため、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。

アレイに適切な ASL が使え、インストールされている場合は、上記の制限なしに使えます。

p.87 の「サードパーティ製ドライバの共存」を参照してください。

VxVM の ZFS との共存

ZFS は、Oracle for Solaris によって提供されるプールされたストレージモデルに基づくファイルシステムの一つです。ファイルシステムは、共通のストレージプール (zpool) から直接引き出すことができます。Veritas Volume Manager (VxVM) は ZFS のディスクと同じシステムで使用できます。

VxVM は、ZFS が使っているデバイスを、ディスクを上書きする可能性のある任意の VxVM 操作から保護します。これらの操作には、VxVM が使用するためのディスクの初期化や、ディスクのカプセル化が含まれます。ZFS が使っているデバイスに対して、これらのうちのいずれかの VxVM 操作を実行しようとする、VxVM によってエラーメッセージが表示されます。

VxVM で ZFS ディスクを管理するには、それらのディスクを ZFS の制御下から削除する必要があります。同様に、ZFS で VxVM ディスクの管理を開始する前には、VxVM の制御からディスクを除外する必要があります。

ディスクが ZFS で使われているかどうかを確認するには

- ◆ 次のように、`vxdisk list` コマンドを使います。

```
# vxdisk list
DEVICE                                TYPE          DISK  GROUP  STATUS
c1t0d0s2                              auto:none     -    -      online invalid
c1t1d0s2                              auto:none     -    -      online invalid
c2t5006016130603AE5d2s2              auto:ZFS      -    -      ZFS
c2t5006016130603AE5d3s2              auto:SVM      -    -      SVM
c2t5006016130603AE5d4s2              auto:cdsdisk -    -      online
c2t5006016130603AE5d5s2              auto:cdsdisk -    -      online
```

VxVM ディスクを ZFS ディスクとして再利用するには

- 1 ディスクがディスクグループに属する場合は、そのディスクをディスクグループから削除するか、またはディスクグループを破棄します。

ディスクをディスクグループから削除するには、次のコマンドを使います。

```
# vxdg [-g diskgroup] rmdisk diskname
```

ディスクグループを破棄するには、次のコマンドを使います。

```
# vxdg destroy diskgroup
```

- 2 ディスクを VxVM の制御下から削除します。

```
# /usr/lib/vxvm/bin/vxdiskunsetup diskname
```

- 3 これで、ZFS ツールを使ってディスクを ZFS デバイスとして初期化できるようになりました。

詳しくは、Oracle 社のマニュアルを参照してください。

VxVM でディスクを ZFS デバイスとして認識できるようにするには、手順 1 と手順 2 を実行する必要があります。

ZFS ディスクを VxVM ディスクとして再利用するには

- 1 ディスクを zpool から削除するか、zpool を破棄します。

詳しくは、Oracle 社のマニュアルを参照してください。

- 2 dd コマンドを使って、シグネチャブロックを消去します。

```
# dd if=/dev/zero of=/dev/rdisk/c#t#d#s# oseek=16 bs=512 count=1
```

c#t#d#s# は、ZFS デバイスが設定されているディスクスライスです。ディスク全体が ZFS デバイスとして使われている場合は、スライス 0 のシグネチャブロックを消去します。

- 3 これで、vxdiskadm コマンドまたは vxdisksetup コマンドを使ってディスクを VxVM デバイスとして初期化できるようになりました。

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 のコマンドでは、現在の名前の付け方に従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** 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 **on**、永続性の属性を **yes** にする必要があります。

メモ: デバイス名が非常に長いデバイス(32文字以上)は、名前の付け方に関係なく、エンクロージャに基づく名前で表されます。**OS** に基づく名前に **WWN** 識別子が含まれる場合、デバイス名は **WWN** の識別子を使って表示されます。ただし、デバイス名が **31** 文字未満の場合に限ります。デバイス名が **31** 文字より長い場合、そのデバイス名はエンクロージャ名を使って表示されます。

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くてもかどうかなを選択できます。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.105 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 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付け方またはエンクロージャに基づく名前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.16 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。DDL は永続的なデバイス名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、OS 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。

ディスク名が、新しいパス名に対応するようになりました。

TPD 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、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 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.105 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 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。vxddmpadm 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。

```
# vxddmpadm 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、EMC0 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。

```
# vxddisk list
```

DEVICE	TYPE	DISK	GROUP	STATUS
emcpower10s2	auto:sliced	disk1	mydg	online
emcpower11s2	auto:sliced	disk2	mydg	online
emcpower12s2	auto:sliced	disk3	mydg	online
emcpower13s2	auto:sliced	disk4	mydg	online
emcpower14s2	auto:sliced	disk5	mydg	online
emcpower15s2	auto:sliced	disk6	mydg	online
emcpower16s2	auto:sliced	disk7	mydg	online
emcpower17s2	auto:sliced	disk8	mydg	online
emcpower18s2	auto:sliced	disk9	mydg	online
emcpower19s2	auto:sliced	disk10	mydg	online

```
# vxddmpadm setattr enclosure EMC0 tpdmode=native
```

```
# vxddisk list
```

DEVICE	TYPE	DISK	GROUP	STATUS
c6t0d10s2	auto:sliced	disk1	mydg	online
c6t0d11s2	auto:sliced	disk2	mydg	online

c6t0d12s2	auto:sliced	disk3	mydg	online
c6t0d13s2	auto:sliced	disk4	mydg	online
c6t0d14s2	auto:sliced	disk5	mydg	online
c6t0d15s2	auto:sliced	disk6	mydg	online
c6t0d16s2	auto:sliced	disk7	mydg	online
c6t0d17s2	auto:sliced	disk8	mydg	online
c6t0d18s2	auto:sliced	disk9	mydg	online
c6t0d19s2	auto:sliced	disk10	mydg	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。

simple ディスクまたは nopriv ディスクのエンクロージャに基づく名前の付け方

OSに基づく名前の付け方をエンクロージャに基づく名前の付け方に変更すると、simple ディスクまたは nopriv ディスクが error 状態になり、ディスク上の VxVM オブジェクトに障害が発生することがあります。

エンクロージャに基づく名前の付け方への変更で発生する simple ディスクと nopriv ディスクの障害を処理するには、vxdarestore コマンドを使います。システムに simple ディスクや nopriv ディスクがまったく存在しない、または simple ディスクや nopriv ディスクが存在するデバイスの自動設定を VxVM で行っていない場合 (たとえば、RAM ディスクなどの非標準ディスクデバイスの場合) は、このコマンドを使う必要はありません。

メモ: OS に基づく名前の付け方を使っている場合は、vxdarestore を実行できません。さらに、simple ディスクと nopriv ディスクの障害の原因が、エンクロージャ名の変更、デバイス名の変更を伴うハードウェアの再構成または永続的な sliced ディスクの記録を含んでいるシステム上の名前の付け方の変更である場合、vxdarestore は障害に対処しません。

p.107 の「ブートディスクグループに含まれる simple ディスクまたは nopriv ディスクの error 状態の解消」を参照してください。

p.108 の「非ブートディスクグループに含まれる simple ディスクまたは nopriv ディスクの error 状態の解消」を参照してください。

vxdarestore (1M) マニュアルページを参照してください。

ブートディスクグループに含まれる simple ディスクまたは nopriv ディスクの error 状態の解消

ブートディスクグループ (通常は bootdg というエイリアスが設定される) が、simple ディスクと nopriv ディスクのどちらかまたは両方のみで構成されている場合は、名前の付け方を変更すると vxconfigd デーモンが無効状態になります。

ブートディスクグループに含まれる **simple** ディスクまたは **nopriv** ディスクの **error** 状態を解消するには、次の手順を実行します。

- 1 `vxdiskadm` を使って、`c#t#d#s#` 名前の付け方に戻します。
- 2 システムを停止して再起動するか、次のコマンドを入力して **VxVM** 設定デーモンを再起動します。

```
# vxconfigd -kr reset
```

- 3 エンクロージャに基づく名前の付け方を使う場合は、`vxdiskadm` を使って、**sliced** ディスクをディスクグループ `bootdg` に追加し、エンクロージャに基づく名前の付け方に戻した後に、次のコマンドを実行します。

```
# vxdarestore
```

非ブートディスクグループに含まれる **simple** ディスクまたは **nopriv** ディスクの **error** 状態の解消

`bootdg` 以外のインポートされたディスクグループが、**simple** ディスクおよび **nopriv** ディスクでのみ構成される場合、エンクロージャに基づく名前の付け方に変更した後は、ディスクグループが「`online dgdisabled`」状態になります。

非ブートディスクグループに含まれる **simple** ディスクまたは **nopriv** ディスクの **error** 状態を解消するには、次の手順を実行します。

- 1 次のコマンドを使って、ディスクグループをデポートします。

```
# vxdg deport diskgroup
```

- 2 `vxdarestore` コマンドを使って、障害が発生したディスクを復元し、そのディスク上のオブジェクトのリカバリを実行します。

```
# vxdarestore
```

- 3 次のコマンドを使って、ディスクグループを再度インポートします。

```
# vxdg import diskgroup
```

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) を使っ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
- [DMP の制御下にある LUN のオンラインでの再設定](#)
- [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。

次の種類のオンライン動的再設定を実行できます。

- [DMP の制御下にある LUN のオンラインでの再設定](#)
p.111 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。
- [ホストバスアダプタ \(HBA\) のオンラインでの交換](#)
p.116 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.117 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。

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.112 の「[既存のターゲット ID からの LUN の動的削除](#)」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.114 の「[新しいターゲット ID への新しい LUN の動的追加](#)」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.115 の「[既存のターゲット ID からの LUN の置換](#)」を参照してください。
- LUN の特性の変更
p.116 の「[アレイ側からの LUN の特性の変更](#)」を参照してください。

既存のターゲット ID からの LUN の動的削除

DMP (Veritas 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。

既存のターゲット ID から LUN を動的に削除するには

- 1 すべての Volume Manager による使用からデバイスを削除します。
ZFS によって使用中の LUN の場合は、zpool をエクスポートまたは破棄します。
- 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。LUN は、ディスクグループ内に存在せず、**online**、**nolabel**、**online invalid**、**online thinrclm** 状態の場合に削除できます。

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。

- 8 次のプロンプトで、アレイから LUN を削除します。

```
Enclosure=emc0 AVID=017E
Device=emc0_017e Serial=830017E000
-----
PATH=c15t0d6 ctlr=c15 port=7e-a [50:01:43:80:12:08:3c:26]
PATH=c17t0d6 ctlr=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。

新しいターゲット ID への新しい LUN の動的追加

DR (Dynamic Reconfiguration) ツールは、既存のターゲット ID への新しい LUN の追加を簡素化します。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。

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 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。
ツールがデバイス検出を発行します。
- 4 プロンプトが表示されたら、アレイから LUN を追加します。

- [y]を選択して DMP への LUN の追加を続行します。
操作がデバイススキャンを発行します。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]
```

- ディスクに追加するラベルを選択します。
- [終了]を選択して Dynamic Reconfiguration ツールを終了します。

既存のターゲット ID からの LUN の置換

DMP (Veritas Dynamic Multi-Pathing) には、既存のターゲット ID からの新しい LUN の置換を簡素化する 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。

既存のターゲット ID から LUN を動的に置換するには

- すべての Volume Manager による使用からデバイスを削除します。
ZFS によって使用中の LUN の場合は、zpool をエクスポートまたは破棄します。
- vxdiskadm ユーティリティを開始します。

```
# vxdiskadm
```
- vxdiskadm メニューから [Dynamic Reconfiguration 機能] オプションを選択します。
- [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 はディスクグループ内に存在せず、online、nolabel、online invalid、または online thinrlm 状態の場合に置換できます。
- 置換する LUN を 1 つ以上選択します。
- プロンプトで LUN の選択を確認します。

- 7 アレイから 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、アレイから LUN を追加します。
- 10 [y]を選択して DMP への LUN の追加を続行します。
操作がデバイススキャンを発行します。新しく検出されたデバイスが認識可能になります。
DMP 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。たとえば、EMC Symmetrix アレイでは、LUN に書き込み保護（読み取り専用）と読み書き両用を有効にできます。LUN のプロパティを変更する前に、Veritas Volume Manager (VxVM) 制御下からデバイスを削除する必要があります。

LUN のプロパティを変更するには

- 1 すべての **Volume Manager** による使用からデバイスを削除します。
ZFS によって使用中の LUN の場合は、**zpool** をエクスポートまたは破棄します。
- 2 デバイスの特性を変更します。
- 3 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
```

- 4 デバイスをディスクグループに戻します。

```
# vxdbg -g dgname adddisk daname
```

ホストバスアダプタのオンラインでの交換

DMP (Veritas 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。

従来のストレージサブシステムには、冗長性のために 2 つ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。最初のコントローラでオンラインコントローラアップグレードが実行されている間、DMP はすべての I/O を 2 番目の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は、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し、リセットし、オンラインになります。2 番目のコントローラでは同じ処理が実行され、I/O は最初の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。

```
# vxdmpadm 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。

<http://www.symantec.com/docs/TECH170013>

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 \(DMP\) の自動デバイス検出](#)
- [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 Veritas 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.120 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.122 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 \(OS イベント\)](#)
p.121 の「[Dynamic Multi-Pathing \(DMP\) の自動デバイス検出](#)」を参照してください。
- [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.121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.122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 がなくても、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。

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
```

`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
```

Dynamic Multi-Pathing (DMP) の自動デバイス検出

VxVM 4.0 よりも前のリリースでは、VxVM でデバイス検出を行うには `vxdisk scandisks` や `vxddctl enable` などのコマンドを手動で呼び出す必要がありました。vxesd は、Reconfiguration Coordination Manager (RCM) フレームワークとの連携によって検出処理を自動化します。

vxesd デーモンは、デバイス到着イベントの通知を受けるために、スクリプト `es_devfs.pl` を Solaris の `syseventd` デーモンに登録します。新しいデバイスをシステムに接続するために `cfgadm` が呼び出されると、`syseventd` デーモンは、デバイス到着イベント用に登録された `es_devfs.pl` などのスクリプトを実行します。`es_devfs.pl` スクリプトは vxesd とのソケットを確立し、イベントパラメータ (デバイスの物理パス) をデーモンに転送します。次に、vxesd デーモンが `vxconfigd` デーモンに接続して、到着したデバイスの DDL デバイス検出を開始します。操作全体は非同期的に実行され、`cfgadm` コマンドはイベントが `syseventd` キューに追加された後に戻ります。

`cfgadm` によってデバイスが削除された場合、`es_rcm.pl` スクリプトを使って関連する DMP パスを無効化する同様の処理が存在します。削除操作は同期的であり、`cfgadm` コマンドは、登録されたすべての切断スクリプトが実行を完了するまで待機します。

メモ: EMC PowerPath を使うシステムでは、低速な PowerPath 検出処理が原因で、デバイスが自動的に DMP によって要求され、DMP の制御下に入る場合があります。そのような状況では、PowerPath がデバイスの制御を要求してから、ディスクの追加と再起動までの間に vxesd デーモンが停止される場合があります。

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 およびファイバーチャネルデバイスのトポロジーを構築します。Solaris では、vxesd デーモンは iSCSI 管理 API (IMA) を使ってトポロジーを構築します。

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

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 (/etc/vx/dmpevents.log) 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 dmpnode を有効としてマーク付け
- パスまたは dmpnode を無効としてマーク付け
- パスの調整
- I/O エラー解析
- Host Bus Adapter (HBA) と Storage Area Network (SAN) のイベント

ログファイルの場所は /var/adm/vx/dmpevents.log ですが、/etc/vx/dmpevents.log にシンボリックリンクされます。ファイルが 10000 行に達すると、ログがローテーションされます。つまり、dmpevents.log の名前が dmpevents.log.X に変更され、新しい dmpevents.log ファイルが作成されます。

チューニングパラメータ dmp_log_level を使って、イベントログファイルの詳細レベルを変更できます。有効な値は 1 から 4 までです。デフォルトレベルは 1 です。

```
# vxddmpadm 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。

```
# vxddmpadm gettune dmp_log_level
```

各種のログレベルについて詳しくは、vxddmpadm(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 はブート時に 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 Veritas Dynamic Multi-Pathing \(DMP\) 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 Veritas Dynamic Multi-Pathing (DMP) のチューニングについて

Veritas 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。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、次の種類の情報について、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 チューニングパラメータ	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、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まります。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=0
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=30
dmp_sfg_threshold=1
dmp_stat_interval=1
dmp_monitor_ownership=on
dmp_native_multipathing=off
dmp_monitor_fabric=on
dmp_monitor_osevent=off
dmp_native_support=off
```

Namingscheme

```
namingscheme=ebn
persistence=yes
lowercase=yes
use_avid=yes
```

Arraytype

```
arraytype=CLR-A/PF
iopolicy=minimumq
partitionsizes=512
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
```

Arraytype

```
arraytype=ALUA
iopolicy=adaptive
partitionsizes=512
use_all_paths=no
```

```
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
Arraytype
arraytype=Disk
iopolicy=minimumq
partitionsizesize=512
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
Arrayname
arrayname=EMC_CLARiION
iopolicy=minimumq
partitionsizesize=512
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
Arrayname
arrayname=EVA4K6K
iopolicy=adaptive
partitionsizesize=512
use_all_paths=no
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
Arrayname
arrayname=Disk
iopolicy=minimumq
partitionsizesize=512
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
Enclosure
serial=CK200051900278
arrayname=EMC_CLARiION
arraytype=CLR-A/PF
iopolicy=minimumq
partitionsizesize=512
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
dmp_lun_retry_timeout=0
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=0
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=0
Enclosure
serial=DISKS
arrayname=Disk
arraytype=Disk
iopolicy=minimumq
partitionsizesize=512
recoveryoption=nothrottle
recoveryoption=timebound iotimeout=300
redundancy=0
dmp_lun_retry_timeout=0
```

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。

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。

```
# vxdmpadm 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まります。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。

```
# vxdmpadm config check file=filename
```

DMP は設定のチェック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 vxdmpadm 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。

```
# vxdmpadm 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

```
◆ # vxdmpadm 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。

# vxdmpadm 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.133 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)
- 冗長性
- dmp_lun_retry_timeout

名前の付け方の属性:

- 名前の付け方
- 永続
- 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	<p>このパラメータを on に設定した場合、ASL (Array Support Library) によって実行される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、ASL による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、キャッシュは行われません。</p> <p>デフォルト値は on です。</p> <p>ネイティブマルチパスで DMP をサポートするには、キャッシュを有効にする必要があります。したがって、dmp_native_multipathing パラメータが on に設定されている場合、dmp_cache_open パラメータは off に設定できません。</p>
dmp_daemon_count	<p>サービスバスのエラー処理、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。</p> <p>スレッドのデフォルト数は 10 です。</p>
dmp_delayq_interval	<p>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。</p> <p>デフォルト値は 15 秒です。</p>

パラメータ	説明
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。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>
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 - 5.0 より前のリリースに存在したすべての DMP ログメッセージを表示します。</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>デフォルト値は 1 です。</p>

パラメータ	説明
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 <code>check_disabled</code> である場合、または <code>check_periodic</code> ポリシーの <code>check_disabled</code> フェーズの間にのみ最適化されます。</p> <p>デフォルト値は on です。</p>
dmp_lun_retry_timeout	<p>HBA と SCSI ドライバ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>通常は、このような特殊な処理は不要です。したがって、<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 0 です。ディスクのすべてのパスがエラーになった場合、DMP はアプリケーションの I/O をエラーにします。パスの接続は、1 回しか調べられません。</p> <p>DMP 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、DMP がアプリケーション I/O 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 0 以外の値に設定します。LUN のすべてのパスがエラーになっても、I/O を提供する必要がある場合、DMP は指定した期間中、5 秒毎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、DMP はそのことを検出して I/O 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 1 つで I/O が正常に処理されない限り、DMP はエラーが発生したすべてのパスでディスクに I/O を提供する処理をエラーにしません。</p>
dmp_monitor_fabric	<p>イベントソースデーモン (vxesd) が Storage Networking Industry Association (SNIA) HBA API を使うかどうかを決定します。この API により、DDL は SAN トポロジーに関する情報を収集し、ファブリックイベントを監視することで、フェールオーバーのパフォーマンスを向上させます。</p> <p>このパラメータが on に設定されている場合、DDL は SNIA HBA API を使います (この機能を使うため、HBA ベンダー固有の HBA-API ライブラリが提供されているはずですが)。</p> <p>このパラメータが off に設定されている場合、SNIA HBA API は使いません。</p> <p>この DDL 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 5.0 以前の場合、デフォルト設定は off です。5.0 以降のリリースでは、デフォルト設定は on です。</p>

パラメータ	説明
dmp_monitor_osevent	<p>イベントソースデーモン (vxesd) が再設定操作などのオペレーティングシステムイベントを監視するかどうかを決定します。</p> <p>このパラメータが on に設定されている場合、vxesd はオペレーティングシステムのデバイスの接続などの操作を監視します。</p> <p>このパラメータが off に設定されている場合、vxesd はオペレーティングシステムの操作を監視しません。DMP が EMC PowerPath と共存している場合は、問題を避けるためにこのパラメータを off に設定することを推奨します。</p> <p>EMC PowerPath がインストールされていない場合、デフォルト設定は on です。すでに PowerPath がインストールされているシステムに DMP をインストールすると、DMP は dmp_monitor_osevent を off に設定します。</p>
dmp_monitor_ownership	<p>ALUA 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 on に設定すると、DMP はデバイスをポーリングして LUN 所有権の変更を確認します。ポーリング間隔は dmp_restore_interval チューニングパラメータで指定します。デフォルト値は on です。</p> <p>dmp_monitor_ownership チューニングパラメータが off のとき、DMP は LUN 所有権の変更を確認するポーリングを行いません。</p>
dmp_native_multipathing	<p>DMP が OS の RAW パスで直接 I/O を傍受するかどうかを決定します。</p> <p>DMP によって RAW パスで直接 I/O のマルチパス化を行う場合はチューニングパラメータを on に設定し、行わない場合は off に設定します。</p> <p>デフォルト値は off です。</p> <p>ネイティブマルチパスで DMP をサポートするには、キャッシュを有効にする必要があります。したがって、dmp_cache_open パラメータが off に設定されている場合、dmp_native_multipathing パラメータは on に設定できません。</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化を行うかどうかを決定します。</p> <p>DMPでネイティブデバイスのマルチパスを行うには、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>Veritas 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(256KB)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性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SANRIS2800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.65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 要求がスケジューリング設定されていない 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にのみ解釈されます。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.79 の「DMP パスリストアポリシーの設定」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、vxddmpadm start restore コマンドを使っても設定できます。</p> <p>p.79 の「DMP パスリストアポリシーの設定」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"> ■ check_all ■ check_alternate ■ check_disabled ■ check_periodic <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.79 の「DMP パスリストアポリシーの設定」を参照してください。</p>
dmp_restore_state	<p>このパラメータを enabled に設定した場合、パスリストアスレッドの開始が有効になります。</p> <p>p.79 の「DMP パスリストアポリシーの設定」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.81 の「DMP パスリストアスレッドの停止」を参照してください。</p>
dmp_retry_count	<p>パスビジーエラーのために特定のパスで I/O がエラーになったとき、DMP はそのパスをビジー状態に設定して、その後 15 秒間使用しません。特定のパスが</p> <p>dmp_retry_count に指定した回数のパスビジーエラーを連続して報告した場合、DMP はこのパスをエラーありに設定します。dmp_retry_count のデフォルト値は 5 です。</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 30 秒です。</p>

パラメータ	説明
dmp_sfg_threshold	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、サブパ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。 デフォルト値は 1 です。
dmp_stat_interval	DMP 統計情報を収集する間隔。 最小値は 1 秒で、これがデフォルトです。

用語集

bootdg	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。
DCO ボリューム	永続 FastResync 変更マップと DRL を保持するために使う特別なボリューム。「dirty region logging」を参照。
DCO (データ変更オブジェクト)	DCO ボリュームの FastResync マップについての情報を管理するために使う VxVM オブジェクト。ボリューム上で永続 FastResync を実行するには、DCO オブジェクトと DCO 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。
dirty region logging	VxVM 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 DCO ボリュームがボリュームに関連付けられている場合、DRL (dirty region logging) は DCO ボリュームに保存されます。それ以外の場合、DRL 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。
FastResync	高速再同期化機能。STALE 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。
Fibre Channel	ストレージエリアネットワーク (SAN) 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。
hostid	VxVM でホストを識別するための文字列。ホストのホスト ID は volboot 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。
JBOD (Just a Bunch Of Disks)	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。
RAID (Redundant Array of Independent Disks)	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。
SAN (storage area network: ストレージエリアネットワーク)	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。
VM ディスク	VxVM 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。VM ディスクは、VxVM ディスクと呼ぶこともあります。
volboot ファイル	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。volboot ファイルの保存場所はシステムによって異なります。

vxconfigd	VxVM 設定デーモン。VxVM の設定の変更を行います。VxVM の操作の前に、このデーモンを起動する必要があります。
空きサブディスク	どのブレックスとも関連付けられておらず、putil[0] フィールドが空であるサブディスク。
空き領域 (free space)	VxVM 制御下にあるディスク領域のうち、サブディスクに割り当てられておらず、かつ他の VxVM オブジェクト用に予約されていない領域。
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。
永続 FastResync	データ変更マップをディスクの DCO 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 FastResync 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。
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。
エンクロージャ	「ディスクエンクロージャ」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、ブレックス、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。
開始ノード	システム管理者が、VxVM 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。
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/vfstab 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	ストライプ化したブレックス内の 1 つまたは複数のサブディスクセット。ストライプ化は、ブ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。
関連付けの解除	2 つの VxVM 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、ブレックスからサブディスクの関連付けを解除すると、ブ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	ブレックスに関連付けられたサブディスク。
関連付けられたブレックス	ボリュームに関連付けられたブレックス。
関連付けを解除したサブディスク	ブ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ブ 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。
関連付け	VxVM 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えます。
共有 VM ディスク	クラスタ内の共有ディスクグループに属する VM ディスク。
共有ディスクグループ (shared disk group)	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 (クラスタ共有ディスクグループともいいます)。
共有ボリューム (shared volume)	共有ディスクグループに属するボリューム。クラスタの複数のノードで同時に起動されま 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)。
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	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)。
原子操作 (atomic operation)	完全に成功するか、または失敗してすべてが操作開始前の状態のままになる操作。成功 すると、すべての操作がただちに有効になります。その変更過程はユーザーには認識さ れません。操作が一部でも失敗した場合、操作は中止され部分的変更はすべて破棄さ れます。 クラスタでは、原子操作が行われる場合は常にすべてのノードが対象になります。
サブディスク	論理ディスクセグメントを形成する連続したディスクブロックのセット。サブディスクをプレッ クスと関連付けることによって、ボリュームを形成することができます。
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。
ストライプユニットサイズ	各ストライプユニットのサイズ。The default stripe unit size is 64KB.ストライプユニット サイズは、ストライプ幅と呼ぶこともあります。
ストライプユニット	各ストライプ化プレックスの (カラム内の) サブディスクに、交互に割り当てられる同じサイ ズの領域。アレイにおけるストライプユニットは、アレイの次のディスクから割り当てが実行 される前に各ディスク上に存在する、論理的に連続した一連のブロックです。ストライプユ ニットは、ストライプエレメントともいいます。
ストライプ	一連のカラムにまたがって同じ位置をとる一連のストライプユニット。
スナップショット	ボリューム (対話的スナップショット) またはファイルシステム (ファイルシステムスナップシ ョット) のポイントインタイムコピー (PITC: point-in-time copy)。

スパースブックス	ボリュームより小さいブックスまたは空白 (バックアップ先のサブディスクを持たないブックスの領域)が存在するブックス。
スライス	論理ディスクデバイスの標準区画。パーティションとスライスという用語は、同じ意味で使われます。
スレーブノード	クラスタのマスターノードとして指定されていないノード。
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。
接続	VxVM 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。
切断	VxVM 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。
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 VxVM 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。
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。
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。
ディスクアクセス名	デバイス名の別称。
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があります。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 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。 たとえば、デバイス名 /dev/sbus@1f,0/QLGC,isp@2,10000/sd@8,0:c で表されるディスクの場合、パスの構成要素 QLGC,isp@2,10000 は、ディスク sd@8,0:c のホストに接続されたディスクコントローラを表します。
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。
ディスクメディア名	ディスク名の別称。f
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	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の識別子が付けられます。
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。
デバイス名	c0t0d0s2 などの物理ディスクへのアクセスに使われるデバイス名またはアドレス。 c#t#d##s# 構文は、コントローラ、ターゲットアドレス、ディスク、スライス(またはパーティション)を識別します。 SAN 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。
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。

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	クラスタ内のホストの 1 つ。
パスグループ	ディスクが vxvmpc 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。
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。
非永続 FastResync	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 FastResync の形式の 1 つ。
ファイルシステム	1 つの構造に組織された、ファイルの集合。UNIX のファイルシステムは、ディレクトリやファイルで構成された階層構造になっています。
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。
物理ディスク	下位ストレージデバイス。必ずしも VxVM の制御下にある必要はありません。
ブートディスクグループ (boot disk group)	システムがブートされる場合があるディスクが含まれた専用ディスクグループ。
ブートディスク (boot disk)	システムをブートするために使われるディスク。

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。
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。
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。
分散	ボリューム（およびそのファイルシステムやデータベース）が大きすぎて 1 つのディスクに収まらない場合に、複数の物理ディスクにまたがって設定するレイアウト手法。
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。
ボリュームイベントログ	ボリュームドライバイベントをユーティリティにレポートするデバイスインターフェース (/dev/vx/event)。
ボリューム設定デバイス	ボリューム設定デバイス (/dev/vx/config) 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。
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バ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 /dev/vx/rdisk に表示され、ブロックデバイスノードが /dev/vx/dsk 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	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。
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。
マルチパス化	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。
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。
無効にされているパス	I/O 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。
有効にされているパス	I/O に使えるディスクへのパス。
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。
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 VxVM 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。
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。VxVM 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	UNIX 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 VxVM 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)。
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。
ログサブディスク	DRL を保存するために使うサブディスク。
ログプレックス	RAID 5 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、DRL プレックスを指す場合にも使われます。
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。

記号

/dev/vx/dmp ディレクトリ 10
 /dev/vx/rmdp ディレクトリ 10
 /etc/vx/dmppolicy.info ファイル 65

A

A/A-A ディスクアレイ 8
 A/A ディスクアレイ 8
 A/P-C ディスクアレイ 9~10
 A/P-F ディスクアレイ 9
 A/P-G ディスクアレイ 10
 A/P ディスクアレイ 9
 active パス属性 62
 adaptiveminq ポリシー 66
 adaptive 負荷分散 66
 APM
 設定 81
 ASL
 Array Support Library 85
 アレイサポートライブラリ 86
 auto-trespass モード 9

B

balanced パスポリシー 67

C

c# 17
 c#t#d#s# 17
 c#t#d#s# に基づく名前の付け方 17
 c0d0t0 17
 check_all ポリシー 79
 check_alternate ポリシー 80
 check_disabled ポリシー 80
 check_periodic ポリシー 80

D

d# 17
 DDI_NT_FABRIC プロパティ 85
 DDL 16
 デバイス検出層 89

DISKS カテゴリ 86

サポートされているディスクの一覧表示 96
 ディスクの削除 99
 ディスクの追加 97

DMP

check_all リストアポリシー 79
 check_alternate リストアポリシー 80
 check_disabled リストアポリシー 80
 check_periodic リストアポリシー 80
 DMP データベース情報の表示 38
 DMP デバイスからのブート 14
 DMP ノードによって制御されるパスの表示 47
 DMP ノードの表示 45~46
 DMP パスリストアスレッドの状態の表示 81
 DMP パスリストアポリシーの設定 79
 DMP リストアデーモンの停止 81
 DMP リストアポーリング間隔の設定 79
 I/O エラーに対する応答の設定 75、78
 I/O 調整の設定 76
 I/O 統計情報の収集 56
 path-switch チューニングパラメータ 137
 recoveryoption 値の表示 78
 TPD 情報の表示 52
 vxdmpadm 43
 アレイポートに関する情報の表示 52
 アレイポートのパスの表示 48
 アレイポートの無効化 72
 アレイポートの有効化 73
 エンクロージャに関する情報の表示 51
 エンクロージャに対する DMP ノードの表示 45
 エンクロージャに基づく名前付け 11
 エンクロージャ名の変更 74
 クラスタ環境上 14
 コントローラに関する情報の表示 50
 コントローラのパスの表示 48
 コントローラの無効化 72
 コントローラの有効化 73
 セカンダリパスでの I/O のスケジュール設定 69
 ディスク検出 84
 ディスクデバイスの設定 84
 テンプレートをを使ったチューニング 125
 動的マルチパス 8

- ノード 10
- ノードの LUN グループの表示 47
- バスエージング 134
- バス情報の表示 38
- バスに対する DMP ノードの表示 44
- バスの無効化 72
- バスの有効化 73
- バスフェールオーバー機構 12
- 負荷分散 14
- メタノード 10
- リストアポリシー 79
- ログ記録レベル 134
- dmp_cache_open チューニングパラメータ 133
- dmp_daemon_count チューニングパラメータ 133
- dmp_delayq_interval チューニングパラメータ 133
- dmp_fast_recovery チューニングパラメータ 134
- dmp_health_time チューニングパラメータ 134
- dmp_log_level チューニングパラメータ 134
- dmp_low_impact_probe 135
- dmp_lun_retry_timeout チューニングパラメータ 135
- dmp_monitor_fabric チューニングパラメータ 135
- dmp_monitor_osevent チューニングパラメータ 136
- dmp_monitor_ownership tunable 136
- dmp_native_multipathing チューニングパラメータ 136
- dmp_native_support チューニングパラメータ 137
- dmp_path_age チューニングパラメータ 137
- dmp_pathswitch_blks_shift チューニングパラメータ 137
- dmp_probe_idle_lun チューニングパラメータ 138
- dmp_probe_threshold チューニングパラメータ 138
- dmp_restore_cycles チューニングパラメータ 138
- dmp_restore_interval チューニングパラメータ 138
- dmp_restore_state チューニングパラメータ 139
- dmp_scsi_timeout チューニングパラメータ 139
- dmp_sfg_threshold チューニングパラメータ 140
- dmp_stat_interval チューニングパラメータ 140
- DMP サポート
 - JBOD デバイス 86
- DMP の設定
 - テンプレートの使用 125
- DMP のチューニング
 - テンプレートの使用 125
- DMP ノード
 - DMP ノードについての統合された情報の表示 45
 - 名前の設定 41
- DMP のバスフェールオーバー 12
- DMP リストアのポーリング間隔 79
- DR
 - 動的再設定 14

E

- EMC PowerPath
 - DMP との共存 88
- EMC Symmetrix
 - 自動検出 88
- errord デーモン 12

F

- FAILFAST フラグ 12

H

- HBA
 - サポートの一覧表示 91
 - ターゲットの一覧表示 91
 - ポートの一覧表示 91
- HBA 情報
 - 表示 50

I

- I/O
 - DMP の統計情報の収集 56
 - セカンダリバスでのスケジュール設定 69
 - 調整 12
- I/O 調整 76
- I/O 調整オプション
 - 設定 79
- I/O ポリシー
 - 指定 65
 - 表示 65
 - 例 70
- iSCSI パラメータ
 - DDL での管理 93
 - vxddladm での設定 93

J

- JBOD
 - DISKS カテゴリからのディスクの削除 99
 - DISKS カテゴリへのディスクの追加 97
 - DMP サポート 86
 - サポートされているディスクの一覧表示 96

L

- LUN 9
 - アイドル状態 138
- LUN グループ
 - 詳細の表示 47
- LUN グループフェールオーバー 10

M

mrl
 キーワード 64

N

nomanual パス属性 62
 nopreferred パス属性 62
 nopriv ディスク
 エンクロージャの問題 107

O

OS に基づく名前の付け方 17
 OTHER_DISKS カテゴリ 86

P

PowerPath
 DMP との共存 88
 preferred priority パス属性 63
 primary パス属性 63
 priority 負荷分散 68

R

restored デーモン 12

S

s# 17
 s2 パーティション 17
 scandisks
 vxdisk サブコマンド 84
 secondary パス属性 63
 simple ディスク
 エンクロージャの問題 107
 single active パスポリシー 69
 standby パス属性 63

T

t# 17
 TPD
 共存のサポート 88
 パス情報の表示 52
 tpdmode 属性 106

U

use_all_paths 属性 69
 use_avid
 vxddladm オプション 104

V

vxdaresore
 simple/nopriv ディスクの障害の処理 107
 vxdctl enable
 デバイス検出機能の呼び出し 87
 vxddladm
 DISKS カテゴリからのディスクの削除 89、99～100
 DISKS カテゴリ内のサポートされているディスクの一
 覧表示 96
 DISKS カテゴリへのディスクの追加 98
 iSCSI パラメータの設定 93
 外部デバイスの追加 100
 サポートされているディスクアレイの一覧表示 95
 サポートされる HBA の一覧表示 91
 すべてのデバイスの一覧表示 90
 設定されたターゲットの一覧表示 92
 設定されたデバイスの一覧表示 93
 ディスクアレイのサポートを再有効化するための使
 用 95
 ディスクの名前の付け方の表示 105
 名前の付け方の変更 104
 ホストバスアダプタ上のポートの一覧表示 91
 無効にされたディスクアレイの一覧表示 96、98
 vxdisk
 ディスクアクセス名の検出 109
 ディスクデバイスのスキャン 84
 マルチパス情報の表示 40
 vxdiskadm
 ディスクの名前の付け方の変更 102
 vxdiskconfig
 目的 84
 vxdisk スキャンディスク
 デバイスの再スキャン 84
 デバイスのスキャン 84
 vxdmpadm
 /O エラーに対する応答の設定 75
 APM 情報の表示 82
 APM の削除 82
 APM の設定 82
 DMP データベース情報の表示 38
 DMP での I/O の無効化 72
 DMP での I/O の有効化 73
 DMP でのコントローラの無効化 37
 DMP ノードによって制御されるパスの表示 47
 DMP パスリストアポリシーの指定 79
 DMP リストアスレッドの状態の表示 81
 DMP リストアデーモンの停止 81
 I/O エラーに対する応答の設定 78
 I/O エラーリカバリ設定の表示 78

I/O 調整の設定 76
 I/O 調整の設定の表示 78
 I/O 統計情報の収集 56
 I/O ポリシーの設定 68～69
 I/O ポリシーの表示 65
 TPD 情報の表示 52
 TPD 名前の付け方の変更 106
 アレイポートに関する情報の一覧表示 52
 エンクロージャに関する情報の表示 51
 エンクロージャに対する DMP ノードの表示 45
 エンクロージャ名の変更 74
 コントローラに関する情報の表示 50
 ディスクアクセス名の検出 109
 パス属性の設定 63
 パスに対する DMP ノードの表示 44、47
 パーティションサイズの表示 65
 リストアポーリング間隔の設定 79
vxmdpdm list
 DMP ノードの表示 45
vxprint
 エンクロージャに基づくディスク名 109
 エンクロージャに基づくディスク名で使用 109
VxVM
 ディスク検出 85

W
 WWN 識別子 17、103

あ
 アイドル状態の LUN 138
 アクセスポート 9
 アクティブ/アクティブディスクアレイ 8
 アクティブ/パッシブディスクアレイ 9
 アクティブパス
 デバイス 63～64
 アレイ
 DMP サポート 85
 アレイサポートライブラリ (ASL) 86
 アレイのボリューム ID
 デバイスの名前の付け方 104
 アレイポート
 DMP に対する有効化 73
 DMP の無効化 72
 情報の表示 52
 アレイポリシーモジュール (APM)
 設定 81
 一覧表示
 サポートされているディスクアレイ 94

永続
 デバイス命名オプション 104
 永続的なデバイス名データベース 105
 永続的なデバイス命名 105
 エラー
 一時的なエラーの処理 135
 エンクロージャ 18
 nopriv ディスクの問題 107
 simple ディスクの問題 107
 情報の表示 51
 ディスクアクセス名の検出 109
 パスに関する属性の設定 64
 パスの冗長性 63～64
 パスの属性の設定 62
 エンクロージャに基づく名前付け
 DMP 11
 エンクロージャに基づく名前の付け方 18、20
 vxprint により表示 109
 エンクロージャベースの名前の付け方 102
 オプション値の再試行
 設定 78

か

外部デバイス
 追加 100
 カスタム名
 DMP ノード 41
 カテゴリ
 ディスク 86
 キューに入れられた I/O
 統計の表示 59
 クラスタ
 DMP の使用 14
 コントローラ
 DMP での無効化 37
 DMP に対する有効化 73
 DMP の無効化 72
 情報の表示 50
 コントローラ ID
 表示 50

さ

最少キュー負荷分散ポリシー 68
 サードパーティ製ドライバ (TPD) 88
 指定
 冗長性のレベル 64
 冗長性の最小レベル
 デバイスの指定 64

- デバイスの表示 63
- 冗長性のレベル
 - デバイスの指定 64
 - デバイスの表示 63
- 冗長ループアクセス 19
- ストレージプロセッサ 9
- スライス
 - s2 17
- セカンダリパス 9
- セカンダリパスの表示 40
- 設定
 - パスの冗長性レベル 64
- 属性
 - active 62
 - nomannual 62
 - nopreferred 62
 - preferred priority 63
 - primary 63
 - secondary 63
 - standby 63
 - パスに関する設定 64
 - パスに対する設定 62

た

- ターゲット
 - 一覧表示 91
- チューニングパラメータ
 - dmp_cache_open 133
 - dmp_daemon_count 133
 - dmp_delayq_interval 133
 - dmp_fast_recovery 134
 - dmp_health_time 134
 - dmp_log_level 134
 - dmp_low_impact_probe 135
 - dmp_lun_retry_timeout 135
 - dmp_monitor_fabric 135
 - dmp_monitor_osevent 136
 - dmp_monitor_ownership 136
 - dmp_native_multipathing 136
 - dmp_native_support 137
 - dmp_path_age 137
 - dmp_pathswitch_blks_shift 137
 - dmp_probe_idle_lun 138
 - dmp_probe_threshold 138
 - dmp_restore_cycles 138
 - dmp_restore_interval 138
 - dmp_restore_state 139
 - dmp_scsi_timeout 139
 - dmp_sfg_threshold 140

- dmp_stat_interval 140
- 調整 12
- ディスク 86
 - c0t0d0 17
 - DISKS カテゴリからのディスクの削除 99
 - DISKS カテゴリへの追加 97
 - DMP による検出 84
 - JBOD でサポートされているディスクの一覧表示 96
 - OTHER_DISKS カテゴリ 86
 - VxVM による検出 85
 - 新しく追加されたディスクの設定 84
 - アレイサポートライブラリ 86
 - 永続的な名前の設定 105
 - エンクロージャ 18
 - カテゴリ 86
 - 検出機能の呼び出し 87
 - スキャン 84
 - セカンダリパス 40
 - デバイス検出層 89
 - 名前の付け方 17
 - 名前の付け方の表示 105
 - 名前の付け方の変更 102
 - プライマリパス 40
 - 無効にされているパス 40
 - メタデバイス 16
 - 有効にされているパス 40

ディスクアレイ

- A/A 8
- A/A-A 8
- A/P 9
- A/P-F 9
- A/P-G 10
- DISKS カテゴリからのディスクの削除 99
- DISKS カテゴリ内のサポートされているディスクの一覧表示 96
- DISKS カテゴリへのディスクの追加 97
- DMP でのサポート 94
- JBOD デバイス 86
- アクティブ/アクティブ 8
- アクティブ/パッシブ 9
- サポートされているディスクアレイの一覧表示 95
- サポートの再有効化 95
- サポートの無効化 95
- 非対称アクティブ/アクティブ 8
- 複数パス 15
- 無効にされているディスクアレイの一覧表示 96
- ディスク名
 - 永続的な名前の設定 105

デバイス

- JBOD 86
- 外部デバイスの追加 100
- すべての一覧表示 90
- パスの冗長性 63~64
- パス名 16
- ファブリック 84
- メタデバイス 16

デバイス検出

- 導入 16
- 部分的 84

デバイス検出層 89

デバイス検出層 (DDL) 16、89

デバイス名 16

- 永続的な名前の設定 105
- ユーザー指定 41

統計情報の収集 12

統計の表示

- キューに入れられた I/O 59
- 無効な I/O 59

動的再設定 14

な

名前

- デバイス 16

名前の付け方

- DMP ノード 41
- TPD エンクロージャに対する変更 106
- ディスク 17
- ディスクの表示 105
- ディスクの変更 102

ノード

- DMP 10

は

パス

- DMP に対する有効化 73
- DMP の無効化 72
- 属性の設定 62、64

パスエージング 134

パーティション

- s2 17

パーティションサイズ

- 値の表示 65
- 指定 67

パフォーマンス

- DMP の負荷分散 14

非 auto-trespass モード 9

非対称アクティブ/アクティブディスクアレイ 8

非明示的フェールオーバーモード 9

表示

- DMP ノード 45
- HBA 情報 50
- サポートされているディスクアレイ 94
- 冗長性のレベル 63

ビンポン効果 15

ファブリックデバイス 84

フェールオーバーモード 9

負荷分散 8

- ポリシーの指定 65

- ポリシーの表示 65

部分的なデバイス検出 84

プライマリパス 9、40

ポート

- 一覧表示 91

ま

マルチパス化

- 情報の表示 38

無効な I/O

- 統計の表示 59

無効にされているパス 40

明示的フェールオーバーモード 9

メタデバイス 16

メタノード

- DMP 10

や

有効にされているパス

- 表示 40

ユーザー指定デバイス名 41

ら

ラウンドロビン

- 負荷分散 68

リカバリオプション値

- 設定 78

リスト

- DMP ノード 45

リストアポリシー

- check_all 79
- check_alternate 80
- check_disabled 80
- check_periodic 80

論理ユニット番号 9

わ

ワールドワイドネーム識別子 17、103