

# Oracle® Solaris 11.1 の管理: SAN 構成およびマルチパス化

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

#### U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

# 目次

---

はじめに .....	9
<b>1 Solaris I/O マルチパス化の概要 .....</b>	<b>13</b>
Solaris I/O マルチパス化の新機能 .....	13
Solaris I/O マルチパス化の概要 .....	15
ファイバチャネルソフトウェアの機能 .....	16
iSCSI ソフトウェア機能 .....	17
SAS ソフトウェア機能 .....	17
Solaris I/O マルチパス化機能 .....	18
サポートされるデバイス標準規格 .....	20
<b>2 ファイバチャネルマルチパス化構成の概要 .....</b>	<b>21</b>
FC デバイス構成タスクの概要 .....	21
Solaris I/O マルチパス化構成の考慮事項 .....	22
<b>3 Solaris I/O マルチパス化機能の構成 .....</b>	<b>25</b>
マルチパス化 I/O 機能の構成 .....	25
マルチパス化の考慮事項 .....	26
マルチパス化の有効化と無効化 .....	26
▼ マルチパス化を有効にする方法 .....	27
▼ マルチパス化を無効にする方法 .....	28
▼ マルチパス化が有効か無効かを確認する方法 .....	29
ポート単位ベースでのマルチパス化の有効化または無効化 .....	30
ポート構成の考慮事項 .....	30
サードパーティー製ストレージデバイスの構成 .....	32
サードパーティー製デバイスの構成の考慮事項 .....	32
サードパーティー製ストレージデバイスの構成: 新規デバイス .....	33

サードパーティー製ストレージデバイスの構成: デバイスの無効化 .....	34
デバイス名の変更の表示 .....	35
自動フェイルバックの構成 .....	36
▼ 自動フェイルバックの構成方法 .....	36
<b>4</b> マルチパス化デバイスの管理 .....	39
マルチパス化デバイスの管理 .....	39
マルチパス化サポート情報の表示 .....	39
マルチパス化サポートのための自動フェイルバックの構成 .....	46
<b>5</b> ファブリック接続されたデバイスの構成 .....	55
FC デバイスの考慮事項 .....	55
FC デバイスの追加 .....	56
▼ FC デバイスを追加する方法 .....	56
SPARC でのファブリックブートデバイスの構成 .....	57
ファブリックブートデバイスの考慮事項 .....	57
<b>6</b> <b>Solaris iSCSI</b> イニシエータの構成 .....	59
Oracle Solaris iSCSI の技術 (概要) .....	59
Oracle Solaris iSCSI のソフトウェア要件およびハードウェア要件の識別 .....	60
iSCSI イニシエータタスクの構成 .....	61
iSCSI 関連の用語 .....	61
推奨される iSCSI 構成方法 .....	62
動的または静的ターゲット発見の構成 .....	63
▼ iSCSI イニシエータを構成する方法 .....	64
▼ iSCSI ディスクにアクセスする方法 .....	66
▼ 発見されたターゲットデバイスを削除する方法 .....	66
iSCSI ベースのストレージネットワークにおける認証の構成 .....	68
▼ iSCSI イニシエータの CHAP 認証を構成する方法 .....	69
▼ iSCSI ターゲットの CHAP 認証を構成する方法 .....	71
他社製の RADIUS サーバーを使用して iSCSI 構成内の CHAP 管理を単純化する .....	71
Oracle Solaris での iSCSI マルチパスデバイスの設定 .....	74
▼ ターゲットの複数の iSCSI セッションを有効にする方法 .....	75
iSCSI 構成の監視 .....	78

▼ iSCSI 構成情報を表示する方法 .....	78
iSCSI イニシエータと iSCSI ターゲットのパラメータの変更 .....	80
iSCSI パラメータのチューニング .....	80
▼ iSCSI イニシエータと iSCSI ターゲットのパラメータを変更する方法 .....	83
iSCSI 構成に関する問題のトラブルシューティング .....	86
ローカルシステムから iSCSI ターゲットに接続できない .....	86
ローカルシステム上で iSCSI デバイスまたは iSCSI ディスクが利用できない .....	87
iSNS 発見方式を使用する場合の LUN マスクの使用 .....	88
iSCSI の一般的なエラーメッセージ .....	88
<b>7 仮想ファイバチャネルポートの構成 .....</b>	<b>95</b>
NPIV の概要 .....	95
NPIV の制限事項 .....	96
NPIV ポートの操作 .....	96
▼ NPIV ポートの作成方法 .....	96
▼ NPIV ポートの削除方法 .....	97
▼ NPIV ポートステータスの表示方法 .....	98
<b>8 FCoE ポートの構成 .....</b>	<b>99</b>
FCoE の概要 .....	99
FCoE の制限事項 .....	99
FCoE ポートの構成 .....	100
▼ FCoE ポートの作成方法 .....	100
▼ FCoE ポートの削除方法 .....	101
▼ FCoE ポートステータスの表示方法 .....	101
▼ FCoE ポートの再初期化を強制実行する方法 .....	102
FCoE ハードウェアオフロードの構成 .....	103
<b>9 SAS ドメインの構成 .....</b>	<b>105</b>
SAS マルチパス化の考慮事項 .....	105
SAS デバイスの動的発見 .....	105
SAS ブートデバイスの構成 .....	106

<b>10</b>	<b>IPFC SAN デバイスの構成</b> .....	107
	IPFC の考慮事項 .....	107
	ファイバチャネルアダプタのポートインスタンスの判別 .....	108
	IPFC の呼び出しと構成 .....	110
	▼ ネットワークインタフェースを手動で開始する方法 .....	111
	▼ 自動ネットワーク構成用にホストを構成する方法 .....	111
<b>11</b>	<b>x86 ベースシステムのファイバチャネルデバイスからの Oracle Solaris OS のブート</b> .....	113
	Oracle Solaris OS のセットアップ要件 .....	113
	Oracle Solaris OS のインストールの概要 .....	114
	Oracle Solaris OS のインストール手続き .....	115
	▼ Oracle Solaris OS のインストール方法 .....	115
	▼ DVD またはネットワークベースの OS インストールを実行する方法 .....	115
<b>12</b>	<b>テープデバイス用の持続的なバインド</b> .....	121
	持続的なバインドの概要 .....	121
	テープリンクの作成 .....	122
	▼ テープデバイスリンクの作成方法 .....	123
<b>A</b>	<b>ファブリック接続されたデバイスの手動構成</b> .....	125
	FC デバイスの手動構成 .....	125
	▼ FC デバイスを手動構成する方法 .....	126
	ファブリックデバイスノードの構成 .....	126
	LUN レベル情報が表示できることの確認 .....	127
	マルチパス化が有効にされていないデバイスノードの構成 .....	128
	▼ マルチパス化されていない FC デバイスを手動構成する方法 .....	129
	▼ マルチパス化されていない複数の FC デバイスを構成する方法 .....	130
	Solaris マルチパス化が有効にされているデバイスノードの構成 .....	131
	▼ マルチパス化された個々の FC デバイスを構成する方法 .....	132
	▼ マルチパス化された複数の FC デバイスを構成する方法 .....	133
	ファブリックデバイスの構成の解除 .....	135
	ファブリックデバイスの構成の解除 .....	135

---

<b>B</b>	サポートされる <b>FC-HBA API</b> .....	141
	サポートされるファイバチャネル HBA API .....	141
<b>C</b>	マルチパス化されたデバイスの問題のトラブルシューティング .....	143
	stmsboot の実行中にシステムがブートに失敗する .....	143
	▼ シングユーザーモードでブートの失敗から回復する方法 .....	143
	stmsboot の実行中にシステムがクラッシュする .....	144
	▼ システムクラッシュから回復する方法 .....	144
	索引 .....	147



# はじめに

---

Oracle Solaris 11.1 の管理: SAN 構成およびマルチパス化では、以前は SunStorageTek Traffic Manager ソフトウェアとして知られていた、Oracle Solaris オペレーティングシステムの統合された部分としての Solaris I/O マルチパス化機能の概要について説明します。このガイドには、ソフトウェアおよびデバイスをインストールおよび構成するための段階的な手順も含まれています。

このガイドは、ファイバチャネル (FC) SAN (Storage Area Network) および SAS (Serial Attached SCSI) ドメインを作成して保守するシステム管理者、ストレージ管理者、およびネットワーク管理者を対象としています。SAN および SAS ドメインの管理および保守における高度な専門知識があることを前提としています。

## お読みになる前に

本書を読む前に、最新の Oracle Solaris 11.1 オペレーティングシステム (OS) のリリースノートを確認してください。

## このドキュメントの構成

章	説明
第 1 章 「Solaris I/O マルチパス化の概要」	Oracle Solaris I/O マルチパス化機能の概要について説明します。
第 2 章 「ファイバチャネルマルチパス化構成の概要」	FC デバイス用の Solaris I/O マルチパス化機能の構成の概要について説明します。
第 3 章 「Solaris I/O マルチパス化機能の構成」	FC デバイスのマルチパス化機能を構成する方法について説明します。
第 4 章 「マルチパス化デバイスの管理」	<code>mpathadm</code> コマンドを使用して、マルチパス化デバイス情報を表示し、マルチパスフェイルバックサポートを有効にする方法について説明します。
第 5 章 「ファブリック接続されたデバイスの構成」	FC デバイスを構成する手順を示します。

章	説明
第6章「Solaris iSCSI イニシエータの構成」	iSCSI イニシエータを構成する手順を示します。
第7章「仮想ファイバチャネルポートの構成」	仮想ファイバチャネルポートとしても知られている、Nポート ID 仮想化 (NPIV) ポートを構成するための手順について説明します。
第8章「FCoE ポートの構成」	通常の Ethernet インタフェースにホストされている FCoE ポートを構成するステップについて説明します。
第9章「SAS ドメインの構成」	SAS ドメインを構成する手順を示します。
第10章「IPFC SAN デバイスの構成」	IPFC SAN デバイスを構成するときの考慮事項について説明します。
第11章「x86 ベースシステムのファイバチャネルデバイスからの Oracle Solaris OS のブート」	Oracle Solaris OS を x86 ベースのシステムに手動でインストールする方法について説明します。
第12章「テープデバイス用の持続的なバインド」	ディスクベースのデバイスについての自動発見のメリットを保持しながら Oracle Solaris OS でテープバインドを指定する方法について説明します。
付録 A 「ファブリック接続されたデバイスの手動構成」	ファブリック接続されたデバイスの手動構成について説明します。
付録 B 「サポートされる FC-HBA API」	サポートされる FC-HBA API を一覧表示します。
付録 C 「マルチパス化されたデバイスの問題のトラブルシューティング」	Solaris I/O マルチパス化機能を使用する際に発生する可能性がある問題のトラブルシューティング情報を提供します。

## Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照してください。聴覚に障害をお持ちの場合は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

## 表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表 P-1 表記上の規則

字体	説明	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上的コンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
<b>AaBbCc123</b>	ユーザーが入力する文字を、画面上的コンピュータ出力と区別して示します。	machine_name% <b>su</b> Password:
<i>aabbcc123</i>	プレースホルダ: 実際に使用する特定の名称または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
<i>AaBbCc123</i>	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第6章を参照してください。 キャッシュは、ローカルに格納されるコピーです。 ファイルを保存しないでください。 注: いくつかの強調された項目は、オンラインでは太字で表示されます。

## コマンド例のシェルプロンプト

次の表に、Oracle Solaris OS に含まれるシェルの UNIX システムプロンプトおよびスーパーユーザーのプロンプトを示します。コマンド例のシェルプロンプトは、そのコマンドを標準ユーザーで実行すべきか特権ユーザーで実行すべきかを示します。

表 P-2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	#

表 P-2 シェルプロンプト (続き)

シェル	プロンプト
C シェル	machine_name%
C シェルのスーパーユーザー	machine_name#

# Solaris I/O マルチパス化の概要

---

この章では、以前は StorageTek Traffic Manager ソフトウェアとして知られていた Solaris I/O マルチパス化機能の概要について説明します。

この章の内容は次のとおりです。

- 13 ページの「Solaris I/O マルチパス化の新機能」
- 15 ページの「Solaris I/O マルチパス化の概要」
- 20 ページの「サポートされるデバイス標準規格」

## Solaris I/O マルチパス化の新機能

- **iSCSI** イニシエータ構成 - iSCSI の一般的な構成情報および詳細な iSCSI イニシエータタスク情報がこのドキュメントで扱われています。詳細は、第 6 章「Solaris iSCSI イニシエータの構成」を参照してください。iSCSI ターゲットの構成方法については、『Oracle Solaris 11.1 の管理: デバイスとファイナルシステム』の第 11 章「COMSTAR を使用したストレージデバイスの構成 (タスク)」を参照してください。
- マルチパス化パッケージの変更 - Oracle Solaris 11.1 パッケージ名は `system/storage/multipath-utilities` です。このパッケージのインストールについては、27 ページの「マルチパス化を有効にする方法」を参照してください。
- マルチパス化されたデバイスの名前 - システムが Oracle Solaris OS にインストールされ、Solaris I/O マルチパス化が有効にされたあと、マルチパス化されたデバイスの名前は `c0` で始まります。例:

```
# mpathadm list lu
/dev/rdisk/c0t600A0B800026D63A0000A4994E2342D4d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF54E234013d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
```

```

Total Path Count: 4
Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF44E233FCFd0s2
Total Path Count: 4
Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A0000A4974E23424Ed0s2
Total Path Count: 4
Operational Path Count: 4
/dev/rdisk/c0t600A0B800029065C00007CF34E233F89d0s2
Total Path Count: 4
Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A0000A4964E234212d0s2
Total Path Count: 4
Operational Path Count: 4
/dev/rdisk/c0t600A0B800026D63A000099B94DE2DB34d0s2

```

- **N\_Port ID 仮想化 (NPIV)** – 1つのファイバチャネルアダプタが多数のNポートIDを持つことを可能にするファイバチャネル機能です。各NポートはSAN上で固有の識別子(ポートWWNおよびノードWWN)を持ち、ゾーン作成およびLUNマスクに使用できます。ソフトゾーン作成は、ポートWWNによってポートをグループ化するために使用でき、ゾーン作成の優先される方法です。詳細は、第7章「仮想ファイバチャネルポートの構成」を参照してください。
- **FCoE (Fibre Channel over Ethernet)** – INCITS T11によって開発中の提案された規格です。FCoEプロトコルはEthernet上でファイバチャネルをネイティブにマップし、Ethernet転送スキームから独立しています。このプロトコルは、ファイバチャネルのすべての構造を保持し、FCと同じ待ち時間、セキュリティ、およびトラフィック管理の属性を維持し、FCツール、トレーニング、およびSANへの投資を保護することによって、入出力の統合を実現します。詳細は、第8章「FCoEポートの構成」を参照してください。
- **FCoE ポートの再初期化** – `fcadm force-lip` コマンドを使用して、ポートリンクの再初期化を強制できます。FCoEポートの再初期化によって、スイッチからすべてのゾーン化イニシエータへの登録状態変更通知(RSCN)が発生する可能性があります。詳細は、102ページの「FCoEポートの再初期化を強制実行する方法」を参照してください。
- **MPxIO パス情報の表示** – `prtconf` コマンドおよび `fmdump` コマンドの両方が、MPxIOパス情報を提供するために更新されました。

たとえば次の出力は、マルチパス化されたデバイスを持つシステムからのものです。

```

# prtconf -v | grep path | more
Paths from multipath bus adapters:
    name='path-class' type=string items=1
    name='path-class' type=string items=1
    name='path-class' type=string items=1
    name='path-class' type=string items=1
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:a
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:a,raw
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:b

```

```
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:b,raw
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:c
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:c,raw
dev_path=/scsi_vhci/disk@g600a0b800026d63a0000a4994e2342d4:d
```

ディスクまたはベイ受容体のディスク占有装置についてのシャーシ、受容体、および占有装置情報を表示する `diskinfo` コマンドも、マルチパス化されたデバイス名を表示するために使用できます。例:

```
# diskinfo -O occupant-compdev
c0t50015179594B6F11d0
c0t5000C500335F95E3d0
c0t5000C500335F907Fd0
c0t5000C500335BD117d0
c0t5000C500335DC60Fd0
c0t5000C500335E106Bd0
c0t5000C500335BA8C3d0
c0t5000C500335FC3E7d0
```

- デバイスドライバ構成 - ドライバのカスタマイズは、前のリリースでは `/kernel/drv` ディレクトリでしたが、`/etc/driver/drv` ディレクトリで実行します。この改善により、システムをアップグレードしてもドライバのカスタマイズが上書きされることがなくなります。`/etc/driver/drv` ディレクトリ内のファイルは、アップグレード時に保持されます。`fp.conf`、`mpt.conf`、または `scsi_vhci.conf` への任意の変更は、`/etc/driver/drv` ディレクトリ内で実行する必要があります。

## Solaris I/O マルチパス化の概要

Solaris I/O マルチパス化機能によって、Oracle Solaris OS を実行中のシステムについて複数のアクセスパスが使用可能になります。マルチパス化により、マルチパス化された接続の使用を通じてストレージデバイスの高い可用性が実現されます。このセクションでは、次の内容について説明します。

- [16 ページの「ファイバチャネルソフトウェアの機能」](#)
- [17 ページの「SAS ソフトウェア機能」](#)
- [18 ページの「Solaris I/O マルチパス化機能」](#)

Solaris I/O マルチパス化機能は、SAN または SAS ドメイン上のストレージデバイスを特定します。ソフトウェアによって、ファイバチャネルストレージデバイスを、ループ、ファブリック、またはポイントツーポイントモードのいずれかで接続できます。このソフトウェアは、ファイバチャネル、iSCSI、および SAS ストレージデバイスを管理するための共通インタフェースを提供します。

マルチパス化の準備において、ターゲットとイニシエータを構成する方法については、『Oracle Solaris 11.1 の管理: デバイスとファイルシステム』の第 11 章「COMSTAR を使用したストレージデバイスの構成(タスク)」を参照してください。

## ファイバチャネルソフトウェアの機能

Solaris I/O マルチパス化には、次の主要な機能があります。

- 動的ストレージ発見 - このソフトウェアは、デバイスとデバイス構成に対して実行された任意の変更を自動的に認識します。この機能により、リブートしたり構成ファイルの情報を手動で変更したりすることなく、デバイスがシステムから使用可能になります。
- 持続的なデバイスネーミング - ソフトウェア内部で構成されたデバイスは、リブートや再構成を経過してもデバイスネーミングを維持します。この方針の唯一の例外が /dev/rmt にあるテープデバイスで、これらは除去されるまで変更されず、後日再生成されます。
- ファイバチャネル調停ループ (FCAL) サポート - サーバー上で使用される OpenBoot PROM (OBP) コマンドは、FC ループをスキャンするために FCAL 接続されたストレージにアクセスできます。
- ファブリックブート - Solaris OS は、ファブリックデバイスとファブリック以外のファイバチャネルデバイスからのブートをサポートします。ファイバチャネルスイッチによるファブリックトポロジは、高い速度、多くの接続、およびポートの遮断を実現します。
- FC-HBA ライブラリ - Storage Networking Industry Association ファイバチャネルホストバスアダプタ (SNIA FC-HBA) ライブラリとして以前知られていたものは、現在では FC-HBA ライブラリと呼ばれています。FC-HBA ライブラリのアプリケーションプログラミングインタフェース (API) は、FC HBA の管理を可能にし、FC HBA の情報を収集するために使用できるほかのアプリケーション (Oracle の StorEdge Enterprise Storage Manager など) 向けの標準ベースのインタフェースを提供します。

一般的な FC-HBA API についての詳細は、[libhbaapi\(3LIB\)](#) を参照してください。FC 仕様の追加情報については、次の場所に移動してください:

<http://www.t11.org>

- ファイバチャネルの仮想化 - N ポート ID 仮想化 (NPIV) はファイバチャネル規格の拡張で、1つのファイバチャネルポートによって SAN 上の多数のポートのシミュレーションを行うことができます。これは Oracle VM Server for SPARC や Oracle VM Server 3.0 for x86 ベースのシステムなどの仮想化環境で役立ちます。
- FCoE (Fibre Channel over Ethernet) - カプセル化されたファイバチャネルフレームを拡張された Ethernet 上で転送する新しい T11 規格が使用できます。Solaris FCoE は、通常の Ethernet コントローラと一緒に動作するように設計されたソフトウェア実装です。

## iSCSI ソフトウェア機能

iSCSI は Internet SCSI (Small Computer System Interface) の略語であり、データストレージサブシステムを結合するための、インターネットプロトコル (IP) ベースのストレージネットワーク標準です。iSCSI プロトコルを使用すると、SCSI コマンドが IP ネットワーク経由で転送されるため、ユーザーはあたかもブロックデバイスがローカルシステムに接続されているかのようにネットワーク経由でブロックデバイスにアクセスできます。

この機能は、Solaris システムが iSCSI サーバー (ターゲット) またはクライアント (イニシエータ) のいずれかとして動作できることを意味します。Solaris iSCSI ターゲットを設定する利点は、追加の FC HBA を使用せずに既存のファイバチャネルデバイスをクライアントに接続できることです。また、専用のアレイを持つシステムが、複製されたストレージを ZFS または UFS ファイルシステムと共有することも可能になりました。

詳細は、[第 6 章「Solaris iSCSI イニシエータの構成」](#) を参照してください。

## SAS ソフトウェア機能

- 動的ストレージ発見 - Oracle Solaris OS マルチパス化ソフトウェアは、デバイスとデバイス構成に対して実行された任意の変更を自動的に認識します。これにより、リブートしたり構成ファイルの情報を手動で変更したりすることなく、デバイスがシステムから使用可能になります。
- 持続的なデバイスネーミング - Solaris OS マルチパス化ソフトウェア内部で構成された動的ストレージ発見デバイスは、リブートや再構成を経過してもデバイスネーミングを維持します。

## Solaris I/O マルチパス化機能

Solaris I/O マルチパス化は x86 ベースのシステムではデフォルトで有効にされており、Oracle Solaris OS を実行する SPARC ベースのシステムではオプションです。ソフトウェアには次の機能が含まれています。

- パス管理 - Solaris I/O マルチパス化機能は、OS がサポートする任意のストレージデバイスへのパスを動的に管理します。デバイスへのパスの追加または削除は、パスがオンラインにされたりサービスから削除されたりしたときに自動的に実行されます。マルチパス化が有効なままであっても、デバイス名を変更したりアプリケーションを変更したりせずに、帯域幅および RAS を増加させるためにコントローラを追加できます。Oracle ストレージ製品では、構成ファイルを管理したりデータベースを最新の状態に保ったりすることは不要です。Oracle 以外のベンダー製ストレージについては、サポートを有効する方法と、そのストレージが Solaris I/O マルチパス化機能について認定されていることを確認する方法を、ストレージベンダーにお問い合わせください。
- 単一のデバイスインスタンス - Solaris I/O マルチパス化機能は Oracle Solaris OS に完全に統合されています。マルチパス化されたデバイスは、パスあたり 1 つのデバイスまたはデバイスリンクとしてではなく、単一のデバイスインスタンスとして表示されます。この機能により、パスごとに別々のデバイスではなく 1 つのストレージデバイス表現として表示することで、複雑なストレージアーキテクチャを format コマンドなどのユーティリティーやボリューム管理製品で管理するコストを削減します。
- フェイルオーバーサポート - 高いレベルの RAS を実装するには、ストレージデバイスへの冗長ホスト接続が必要です。Solaris I/O マルチパス化機能では、ストレージパスの障害を管理し、使用可能なセカンダリパスを経由してホスト I/O 接続を維持します。

次のコマンドを使用して、デバイスのフェイルオーバーサポートを判別できます。

```
# mpathadm show mpath-support libmpscsi_vhci.so
mpath-support: libmpscsi_vhci.so
Vendor: Sun Microsystems
Driver Name: scsi_vhci
Default Load Balance: round-robin
Supported Load Balance Types:
    round-robin
    logical-block
Allows To Activate Target Port Group Access: yes
Allows Path Override: no
Supported Auto Failback Config: 1
Auto Failback: on
Failback Polling Rate (current/max): 0/0
Supported Auto Probing Config: 0
Auto Probing: NA
Probing Polling Rate (current/max): NA/NA
Supported Devices:
.
.
.
```

- 対称/非対称型デバイスサポート - 次のディスクストレージデバイスがサポートされます。
  - すべての Oracle ディスクストレージ製品で、対称型と非対称型の両方
  - T10/T11 規格に準拠したサードパーティー製のすべての対称型ディスクデバイス
  - 多くのサードパーティー製の非対称型ディスクアレイ
  - T10 非対称論理ユニットアクセス (ALUA) をサポートする非対称デバイス用に、この T10 規格のサポートが追加されました。使用するデバイスがサポートされるかどうかを確認するには、ストレージベンダーにお問い合わせください。

ディスクストレージアレイで `f_asym_lsi` フェイルオーバーモジュールを使用する場合、次に示すように、`f_asym_lsi` を `scsi_vhci.conf` ファイルの末尾に移動することによって、`ddi-forceload` セクションの順序を手動で変更する必要があります。

```
ddi-forceload =
    "misc/scsi_vhci/scsi_vhci_f_asym_sun",
    "misc/scsi_vhci/scsi_vhci_f_asym_emc",
    "misc/scsi_vhci/scsi_vhci_f_sym_emc",
    "misc/scsi_vhci/scsi_vhci_f_sym_hds",
    "misc/scsi_vhci/scsi_vhci_f_sym",
#    "misc/scsi_vhci/scsi_vhci_f_tpgs_tape",
#    "misc/scsi_vhci/scsi_vhci_f_tape",
    "misc/scsi_vhci/scsi_vhci_f_tpgs",
    "misc/scsi_vhci/scsi_vhci_f_asym_lsi";
```

- I/O 負荷分散 - 単純なフェイルオーバーサポートの提供に加えて、Solaris マルチパス化機能では、ストレージデバイスへの任意のアクティブパスを使用して I/O を送信および受信できます。複数のホスト接続を経由して I/O が経路指定されるため、ホストコントローラの追加によって帯域幅を増加できます。ソフトウェアではラウンドロビン負荷分散アルゴリズムを使用することによって、個々の I/O 要求は、一連のアクティブなホストコントローラに向けて交互に経路指定されます。
- キューの深さ - SCSI ストレージアレイは、論理ユニット番号 (LUN) の形式でシステムにストレージを提供します。LUN では有限のリソースセットが使用可能で、たとえば格納できるデータの量や、1つのデバイスまたは LUN が一度に処理できるアクティブコマンドの数などがあります。デバイスが追加の I/O をブロックするまでに発行できるアクティブコマンドの数は、キューの深さとして知られています。Solaris I/O マルチパス化が有効になると、LUN がホストに対して持つ個々あるいは別個のパスの数に関係なく、LUN ごとに単一のキューが作成されます。この機能によって、ディスクドライブは LUN に対して 1つのキューを維持し均衡させることができ、実質的にキューの深さを管理できます。Oracle Solaris OS 用に利用できるほかのマルチパスソフトウェアには、この機能はありません。

- stmsboot コマンド – Oracle Solaris OS には stmsboot コマンドが含まれており、OS インストールが完了したあと、ブートデバイス上の Solaris マルチパス化機能を有効または無効にできます。このコマンドは SPARC ベースおよび x86 ベースの両方のシステムで使用でき、SAS マルチパス化のサポートを提供します。

SAS マルチパス化は Oracle Solaris OS では SPARC ベースまたは x86 ベースのいずれのシステムについてもデフォルトで有効にされていないため、マルチパス化を有効にするにはインストール後処理ステップとして stmsboot コマンドの使用が必要です。

SPARC ベースのシステムでは FC デバイスのマルチパス化をデフォルトで有効にしないため、インストール後処理ステップとして stmsboot コマンドが必要です。

X86 ベースのシステムでは FC デバイスのマルチパス化をデフォルトで有効にするため、インストール後処理ステップとしての stmsboot コマンドの使用は必須ではありません。

- 動的再構成 – Solaris I/O マルチパス化機能は動的再構成 (DR) 操作をサポートしません。

## サポートされるデバイス標準規格

Solaris I/O マルチパス化機能は、デバイスとの通信とデバイス管理についてのオープンスタンダードに基づいており、標準規格ベースのほかのデバイスおよびソフトウェアとの相互運用性が確保されています。Solaris I/O マルチパス化機能によって次の標準規格がサポートされています。

- ANSI 標準: Information Technology – SNIA Multipath Management API Specification (ANSI INCITS 412-2006)
- T10 標準、SCSI-3、SAM、FCP、SPC、および SBC を含む
- T11.3 FC 標準、FC-PH、FC-AL、FC-LS、および FC-GS を含む
- T11.5 ストレージ管理標準、FC-HBA を含む
- IETF 標準、RFC 2625 を含む
- Serial Attached SCSI-2 (SAS2)

# ファイバチャネルマルチパス化構成の概要

---

この章では、Solaris I/O マルチパス化機能を持つファイバチャネル (FC) デバイスの構成について説明します。

次の項目について説明します。

- 21 ページの「FC デバイス構成タスクの概要」
- 22 ページの「Solaris I/O マルチパス化構成の考慮事項」

## FC デバイス構成タスクの概要

このセクションでは、FC デバイス用の Solaris I/O マルチパス化機能を構成するためのタスクの概要について説明します。

次の表には、構成タスク、各タスクの説明、および関連する手順を見つけることができる参照情報が一覧表示されています。

表 2-1 マルチパス化された FC デバイスを構成するためのタスク

構成タスク	タスクの説明	参照情報
マルチパス化機能を有効にする	マルチパス化は次のようにして有効化されます。 <ul style="list-style-type: none"><li>■ x86/x64 プラットフォームの FC デバイスについてはデフォルトによる</li><li>■ SPARC プラットフォーム上では手動構成による</li><li>■ SAS デバイスについては手動構成による</li></ul>	第 3 章「Solaris I/O マルチパス化機能の構成」。

表 2-1 マルチパス化された FC デバイスを構成するためのタスク (続き)

構成タスク	タスクの説明	参照情報
FC デバイスを構成する	Oracle Solaris OS では、FCAL、ファブリック、およびポイントツーポイントで接続されたデバイスがホストから使用できません。	第 5 章「ファブリック接続されたデバイスの構成」
仮想 FC ポートを構成する	仮想化環境で特に役立つ N ポート ID 仮想化 (NPIV) ポートを構成できます。	第 7 章「仮想ファイバチャネルポートの構成」
FC ブートデバイスを設定する	Solaris I/O マルチパス化機能によって、Solaris サーバーがファブリックデバイスからブートできます。	55 ページの「FC デバイスの考慮事項」
IPFC SAN デバイスを構成する	IPFC デバイスのホスト認識と、SAN における FC 経由の IP の実装を構成できます。IPFC の構成は、FC アダプタポートの Fabric Port (FP) ドライバのインスタンスに依存します。	第 10 章「IPFC SAN デバイスの構成」
SAS デバイスを構成する	Oracle Solaris OS では、SAS デバイスのマルチパス化がサポートされています。	第 9 章「SAS ドメインの構成」

## Solaris I/O マルチパス化構成の考慮事項

Solaris I/O マルチパス化機能を構成する前に、次のことを検討してください。

- ストレージおよびスイッチについてのベンダー固有のドキュメントに従ってポートおよびゾーンを構成します。
- LUN マスクによって、特定の LUN が特定のホストから可視になります。マスクについて説明されているベンダー固有のストレージのドキュメントを参照してください。
- SAN のホストおよびデバイスについて、電源管理を無効にしておく必要があります。電源管理についての詳細は、[poweradm\(1M\)](#) を参照してください。
- SAN ブートプロセスを管理する STMS ブートユーティリティーが Solaris I/O マルチパス化機能に含まれています。stmsboot コマンドを発行すると、マルチパス化ソフトウェアを有効または無効にしたときにデバイス名の変更を反映するよう /etc/vfstab およびダンプ構成の自動更新がアクティブにされます。このソフトウェアは SPARC ベースシステム上のデバイスではデフォルトで無効化されており、x86 ベースシステム上のデバイスではデフォルトで有効化されていることに注意してください。
- ファブリック接続されたデバイスは、インストールおよびブート時に自動的に構成され、ホストから使用できるようになります。

---

注-アップグレードを実行していて、任意のFCデバイスをアップグレード後に使用できなくする場合、これらのデバイスは、`cfgadm -c unconfigure` コマンドを使用して手動で構成を解除する必要があります。ただし、これらのデバイスをシステムで永続的に使用できなくするには、スイッチゾーン機能またはLUNマスクの使用を検討した方がよい場合もあります。`cfgadm -c unconfigure` によって実行された変更は、FCデバイスの手動構成が有効にされないかぎり、リブート後に持続しません。ブートまたはインストール中にFCデバイスの発見を無効にする方法を見つけるには、[付録A「ファブリック接続されたデバイスの手動構成」](#)を参照してください。

---



## Solaris I/O マルチパス化機能の構成

---

この章では、Oracle Solaris OS 内の FC デバイスに、Solaris I/O マルチパス化機能を構成する方法について説明します。また、SPARC ベースシステム、x86 ベースシステム、ポート単位ベース、およびサードパーティー製ストレージデバイス上でマルチパス化機能を有効または無効にする際の考慮事項についても説明します。

次の項目について説明します。

- 25 ページの「マルチパス化 I/O 機能の構成」
- 26 ページの「マルチパス化の考慮事項」
- 26 ページの「マルチパス化の有効化と無効化」
- 30 ページの「ポート単位ベースでのマルチパス化の有効化または無効化」
- 32 ページの「サードパーティー製ストレージデバイスの構成」
- 36 ページの「自動フェイルバックの構成」

### マルチパス化 I/O 機能の構成

FC デバイス用のマルチパス化機能を構成して、サポートされるすべての FC HBA を制御することができます。マルチパス化は SPARC ベースシステム上の FC デバイスについてはデフォルトで無効ですが、x86 ベースシステム上ではデフォルトで有効にされています。マルチパス化機能の構成は、ユーザーの意図したシステムの使用方法に依存します。

---

注 - マルチパス化機能はパラレル SCSI デバイスでは使用できませんが、FC、SAS、および iSCSI ディスクデバイスでは使用できます。マルチパス化はテープドライブやテープライブラリでもサポートされます。

---

## マルチパス化の考慮事項

マルチパス化構成を変更する前に、次の考慮事項に注意してください。そのあとで、後続のセクションで説明されている、マシンアーキテクチャー (SPARC または x86) についての指示に従ってください。一部のデバイスは、マルチパスソフトウェアと一緒に動作するために正しく構成する必要があります。使用するデバイスについてのデバイス固有構成に関する詳細は、ストレージアレイのドキュメントを参照してください。

- デバイス固有およびデバイス名の変更の考慮事項

/dev および /devices のツリーでは、マルチパス化されたデバイスは、それらがマルチパス制御を受けることを示す新しい名前を受け取ります。したがって、デバイスはマルチパス制御を受けるとき、元の名前とは異なる名前を持つことになります。

マルチパスが無効なデバイス名:

```
/dev/dsk/c1t1d0s0
```

マルチパスが有効なデバイス名:

```
/dev/dsk/c0t60003BA27D5170003E5D2A7A0007F3D2d0s0
```

したがって、デバイス名を直接使用するアプリケーションは、マルチパス構成を無効から有効へ、あるいはその逆に変更したときは常に新しい名前を使用するように構成する必要があります。

- /etc/vfstab エントリおよびダンプ構成の更新

システムの /etc/vfstab ファイルおよびダンプ構成には、デバイス名の参照も含まれています。SPARC ベースおよび x86 ベースの両方のシステムでは、stmsboot コマンドによって、/etc/vfstab ファイルダンプ構成は新しいデバイス名で自動的に更新されます。/etc/vfstab ファイルに一覧表示されないアプリケーション依存のファイルシステムが存在する場合、stmsboot コマンドを使用して、古いデバイスパスと新しいデバイスパスのマッピングを判別できます。



注意 - devfsadm -C を実行したか、再構成ブートを実行した場合、古いデバイスパスは存在せず、stmsboot -L コマンドでこの情報を提供できません。

## マルチパス化の有効化と無効化

stmsboot コマンドを使用して、ファイバチャネル (FC) および SAS デバイス用のマルチパス化を有効または無効にできます。stmsboot コマンドは、次回リブート中にデバイス名の変更を反映するよう /etc/vfstab ファイルおよびダンプ構成を更新します。fp.conf または mpt.conf ファイルを手動で編集する必要はありません。

stmsboot -e、-d、および -u オプションには次の考慮事項が適用されます。

- stmsboot コマンドを実行したあと、すぐにリブートする必要があります。
- stmsboot コマンドはマシンをリブートして操作を完了するため、eeprom を使用して、システムが現在のブートデバイスからブートされるようにしてください。
- stmsboot コマンドは、/kernel/drv/fp.conf、/kernel/drv/mpt.conf、および /etc/vfstab ファイルを変更する前に、これらの元のコピーを保存します。stmsboot コマンドの使用中に予期しない問題が発生した場合は、[付録 C 「マルチパス化されたデバイスの問題のトラブルシューティング」](#) を参照してください。

---

注 - 以前の Oracle Solaris リリースでは、stmsboot コマンドは SPARC ベースのホストについてのみ、ブートデバイス上のマルチパス化を有効または無効にするために使用されていました。現在の Oracle Solaris リリースでは、このコマンドは、シリアル接続された SCSI デバイスと FC デバイス上のマルチパス化を有効または無効にするために使用されます。

---

## ▼ マルチパス化を有効にする方法

SPARC ベースまたは x86 ベースのシステム上のすべてのマルチパス対応デバイスについてマルチパス化を有効にするには、次のステップを実行します。特定の FC または SAS HBA ポート上でのみマルチパスを有効にするには、[30 ページの「ポート単位ベースでのマルチパス化の有効化または無効化」](#) を参照してください。

マルチパスソフトウェアは Oracle のサポートされるデバイスを自動的に認識します。サードパーティー製デバイスでマルチパスを有効にするには、/kernel/drv/scsi\_vhci.conf ファイルを /etc/driver/drv にコピーし、[26 ページの「マルチパス化の考慮事項」](#) のデバイス名の変更の考慮事項に示すようにエントリを追加します。

- 1 管理者になります。
- 2 マルチパスソフトウェアパッケージがインストールされていることを確認します。

```
# pkg info system/storage/multipath-utilities
Name: system/storage/multipath-utilities
Summary: Solaris Multipathing CLI
Description: Path Administration Utility for a Solaris Multipathing device
Category: Applications/System Utilities
State: Installed
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.175.0.0.0.0
Packaging Date: Tue Sep 27 01:40:01 2011
Size: 77.29 kB
```

```
FMRI: pkg://solaris/system/storage/multipath-utilities@
0.5.11,5.11-0.175.0.0.0.0:20110927T014001Z
```

そうでない場合はインストールします。

```
# pkg install system/storage/multipath-utilities
```

### 3 デバイスのマルチパス化を有効にします。

```
# stmsboot -e
```

```
WARNING: stmsboot operates on each supported multipath-capable controller
detected in a host. In your system, these controllers are
```

```
/devices/pci@780/pci@0/pci@8/SUNW,qlc@0/fp@0,0
/devices/pci@780/pci@0/pci@8/SUNW,qlc@0,1/fp@0,0
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2
/devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0
/devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0
```

If you do NOT wish to operate on these controllers, please quit stmsboot and re-invoke with `-D { fp | mpt }` to specify which controllers you wish to modify your multipathing configuration for.

```
Do you wish to continue? [y/n] (default: y) y
Checking mpzio status for driver fp
Checking mpzio status for driver mpt
WARNING: This operation will require a reboot.
Do you want to continue ? [y/n] (default: y) y
The changes will come into effect after rebooting the system.
Reboot the system now ? [y/n] (default: y) y
```

---

注- リブート中に、デバイス名の変更を反映するように `/etc/vfstab` およびダンプ構成が更新されます。

---

- 4 (省略可能) リブート後に必要な場合、[26 ページ](#)の「マルチパス化の考慮事項」のデバイス名の変更の考慮事項で説明されているように、新しいデバイス名を使用するようにアプリケーションを構成します。

## ▼ マルチパス化を無効にする方法

SPARC ベースまたは x86 ベースのシステム上のすべてのマルチパス対応デバイスについてマルチパス化を無効にするには、次のステップを実行します。特定の FC または SAS HBA ポート上でのみマルチパスを無効にするには、[30 ページ](#)の「ポート単位ベースでのマルチパス化の有効化または無効化」を参照してください。

- 1 デバイスのマルチパス化を無効にします。

```
# stmsboot -d
```

```
WARNING: stmsboot operates on each supported multipath-capable controller
detected in a host. In your system, these controllers are
```

```

/devices/pci@780/pci@0/pci@8/SUNW,qlc@0/fp@0,0
/devices/pci@780/pci@0/pci@8/SUNW,qlc@0,1/fp@0,0
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1
/devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2
/devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0
/devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0

```

If you do NOT wish to operate on these controllers, please quit stmsboot and re-invoke with -D { fp | mpt } to specify which controllers you wish to modify your multipathing configuration for.

```

Do you wish to continue? [y/n] (default: y) y
Checking mpzio status for driver fp
Checking mpzio status for driver mpt
WARNING: This operation will require a reboot.
Do you want to continue ? [y/n] (default: y) y
The changes will come into effect after rebooting the system.
Reboot the system now ? [y/n] (default: y) y

```

---

注- リブート中に、デバイス名の変更を反映するように /etc/vfstab およびダンプ構成が更新されます。

---

- 2 (省略可能) リブート後に必要な場合、[26 ページ](#)の「マルチパス化の考慮事項」のデバイス名の変更の考慮事項で説明されているように、新しいデバイス名を使用するようにアプリケーションを構成します。

## ▼ マルチパス化が有効か無効かを確認する方法

- 1 管理者になります。
- 2 マルチパス化されたデバイス情報がシステムで使用可能かどうかを確認します。  
次のコマンドによって、マルチパス化されたデバイス情報が返されないため、マルチパス化はこのシステムで無効です。

```

# prtconf -vc /devices/scsi_vhci |grep dev_link.*s2
#

```

次のコマンドによって、マルチパス化されたデバイス情報が返されるため、マルチパス化はこのシステムで有効です。

```

# prtconf -vc /devices/scsi_vhci |grep dev_link.*s2
dev_link=/dev/dsk/c0t5000C500335DC60Fd0s2
dev_link=/dev/rdisk/c0t5000C500335DC60Fd0s2
dev_link=/dev/dsk/c0t5000C500335E106Bd0s2
dev_link=/dev/rdisk/c0t5000C500335E106Bd0s2
dev_link=/dev/dsk/c0t5000C500335BA8C3d0s2
dev_link=/dev/rdisk/c0t5000C500335BA8C3d0s2
dev_link=/dev/dsk/c0t5000C500335FC3E7d0s2

```

```
dev_link=/dev/rdisk/c0t5000C500335FC3E7d0s2
dev_link=/dev/dsk/c0t50015179594B6F11d0s2
dev_link=/dev/rdisk/c0t50015179594B6F11d0s2
dev_link=/dev/dsk/c0t5000C500335F95E3d0s2
dev_link=/dev/rdisk/c0t5000C500335F95E3d0s2
dev_link=/dev/dsk/c0t5000C500335F907Fd0s2
dev_link=/dev/rdisk/c0t5000C500335F907Fd0s2
dev_link=/dev/dsk/c0t5000C500335BD117d0s2
dev_link=/dev/rdisk/c0t5000C500335BD117d0s2
```

## ポート単位ベースでのマルチパス化の有効化または無効化

マルチパス化は、ファイバチャネルホストバスアダプタ (HBA) コントローラの特定のポートで有効または無効にできます。特定の HBA コントローラポートのマルチパス化を有効にした場合、そのコントローラポートに接続されているサポートされるすべてのデバイスは、マルチパス操作作用に有効化されます。

### ポート構成の考慮事項

ソフトウェアをポート別に構成することを開始する前に、次のことを検討してください。

- FC のグローバル設定とポート単位のマルチパス設定は、`/kernel/drv/fp.conf` ファイルで指定されます。  
ポート単位のマルチパス設定は、グローバル設定よりも優先されます。したがって、グローバルなマルチパスが有効でも、特定のポートのマルチパスが無効な場合、そのポートはマルチパス化構成で使用できないことを意味します。逆に、グローバルなマルチパスが無効であっても、特定のポートが適切な `driver.conf` ファイル内で一覧表示されている場合は、そのポートのマルチパス化が有効になることがあります。
- 負荷分散は、`/kernel/drv/scsi_vhci.conf` ファイル内のグローバル負荷分散プロパティによって制御され、ポート単位ベースで制御されません。
- デバイスがホストに対して複数のパスを持つ場合、デバイスへのすべてのパスで、マルチパス化を有効または無効に構成する必要があります。
- ポート別にマルチパス化を構成することによって、このマルチパスソフトウェアは、Symantec (VERITAS) Dynamic Multipathing (DMP) や EMC PowerPath などのほかのマルチパスソリューションと共存できます。ただし、このマルチパスソフトウェアと別のマルチパス化ソリューションの間でデバイスおよびパスを共有しないでください。

### ▼ ポート別のマルチパス化の構成方法

次の手順は SPARC ベースおよび x86 ベースの両方のシステムに適用されます。

マルチパスソフトウェアで制御するポートの数に応じて、マルチパス化を大域的に有効または無効にしたり、指定されたポートについて有効または無効にしたりできます。

- 1 管理者になります。
- 2 マルチパスソフトウェアで制御する HBA コントローラポートを決定します。  
使用可能なデバイスを一覧表示するには、`/dev/cfg` ディレクトリで `ls -l` コマンドを実行します。次の例は、`ls -l` コマンドの出力を示します。

```
# ls -l
lrwxrwxrwx 1 root root 50 Jan 29 21:33 c0 ->
  ../../devices/pci@7c0/pci@0/pci@1/pci@0/ide@8:scsi
lrwxrwxrwx 1 root root 61 Jan 29 21:33 c1 ->
  ../../devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@1:scsi
lrwxrwxrwx 1 root root 61 Jan 29 21:33 c2 ->
  ../../devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2:scsi
lrwxrwxrwx 1 root root 53 Jan 29 21:33 c3 ->
  ../../devices/pci@7c0/pci@0/pci@9/LSILogic,sas@0:scsi
lrwxrwxrwx 1 root root 54 Apr 16 20:28 c5 ->
  ../../devices/pci@780/pci@0/pci@8/SUNW,qlc@0/fp@0,0:fc
lrwxrwxrwx 1 root root 56 Apr 16 20:28 c6 ->
  ../../devices/pci@780/pci@0/pci@8/SUNW,qlc@0,1/fp@0,0:fc
```

注 - コントローラ c5 および c6 は、デュアルポート FC HBA のポート A と B です。コントローラ c1 および c3 は、単一ポートの SAS HBA ポートです。コントローラ c2 は、Oracle の Sun Fire T2000 サーバーの内部 SAS コントローラです。

マルチパス化を明示的に有効または無効にする 1 つまたは複数のポートを決定します。

- 3 `/kernel/drv/fp.conf` ファイルを `/etc/driver/drv/fp.conf` ファイルにコピーします。
- 4 次のいずれかを選択して、特定の FC HBA ポートを有効または無効にします。
  - `/etc/driver/drv/fp.conf` 内で有効にする各 FC HBA ポートについて、次の行を追加します。
 

```
name="fp" parent="parent-name" port=port-number mpxio-disable="no";
```

 ここで、`parent-name` はポートデバイス名、`port-number` は FC HBA ポート番号です。
 

たとえば、次のエントリでは、指定された 2 つのポートを除くすべての FC HBA コントローラポートのマルチパス化が無効になります。

```
mpxio-disable="yes";
name="fp" parent="/pci@6,2000/SUNW,qlc@2" port=0 mpxio-disable="no";
name="fp" parent="/pci@13,2000/pci@2/SUNW,qlc@5" port=0 mpxio-disable="no";
```
  - 無効にする各 FC HBA ポートについて、次の行を追加します。

```
name="fp" parent="parent-name" port=port-number mpzio-disable="yes";
```

例:

```
name="fp" parent="/pci@6,2000/SUNW,qlc@2" port=0 mpzio-disable="yes";
```

- 5 リブートおよび構成プロセスを開始します。

```
# stmsboot -u
```

リブートを確認するメッセージが表示されます。リブート中に、デバイス名の変更を反映するように `/etc/vfstab` ファイルおよびダンプデバイス構成が更新されます。

- 6 (省略可能) リブート後に必要な場合、[26 ページの「マルチパス化の考慮事項」](#)で説明されているように、新しいデバイス名を使用するようにアプリケーションを構成します。

## サードパーティー製ストレージデバイスの構成

---

注- サードパーティー製のデバイスを構成する前に、それらがサポートされていることを確認してください。サードパーティーのユーザードキュメントを参照するか、サードパーティーベンダーに問い合わせ、デバイスがマルチパスソフトウェアと一緒に動作するために必要な正しいベンダー ID、製品 ID、モードおよび各種設定の情報を調べてください。

---

### サードパーティー製デバイスの構成の考慮事項

サードパーティー製デバイスのマルチパス化を構成する前に、次のことに注意します:

- デバイスは `REPORT_LUNS` SCSI コマンド、および `SCSI-3 INQUIRY` コマンド `VPD Device Identification Page (0x83)` をサポートする必要があります。
- デバイスのベンダー ID (VID) および製品 ID (PID) が必要です。これらは `format` コマンドの後にシステムの `INQUIRY` オプションを付けて使用することで取得できます。詳細は、[format\(1M\)](#) のマニュアルページを参照してください。

マルチパス化が有効な場合、マルチパスアクセスは、デバイスを受け入れるデバイス固有の `scsi_vhci` フェイルオーバー実装に依存します。デフォルトは、`scsi_vhci` コードが各フェイルオーバー実装のプロープ関数を自動的に呼び出し、デバイスがサポートされていることを示す最初のプロープ結果を見つけることです。

プロープ実装は、`scsi_inquiry(9S)` データの組み合わせに基づいてサポートを判定します。T10 TPGS (Target-Port-Group-Support) 準拠を示す `INQUIRY` データを持つデバイスは、標準ベースの TPGS フェイルオーバー実装を使用します。非準拠デバイスの

場合、フェイルオーバー実装のプロープは通常、VID/PID を専用のコンパイル済みテーブルと照合して、サポートを判定します。

*probe* プロセスをオーバーライドするために、`scsi_vhci.conf` ファイルは `scsi-vhci-failover-override` プロパティをサポートします。 `scsi-vhci-failover-override` の値は、プロープによって現在受け入れられていないデバイスのサポートを確立したり、プロープサポートをオーバーライドしたり、またはデバイスのマルチパスサポートを無効にしたりするために使用できます。

## サードパーティー製ストレージデバイスの構成: 新規デバイス

マルチパス化はサードパーティー製の対称型ストレージデバイスで構成できません。対称型ストレージデバイスは、ストレージデバイスへのすべてのパスがアクティブで、任意のパスを経由して I/O コマンドを発行できるものです。

### ▼ サードパーティー製デバイスを構成する方法

システムのマルチパス化がすでに構成されている場合、次のステップを実行してサードパーティー製デバイスを構成してください。システムのマルチパス化が無効になっている場合、この章の前の方で説明されているようにマルチパス化を有効にするときにサードパーティー製デバイスを構成できます。

- 1 管理者になります。
- 2 `/kernel/drv/scsi_vhci.conf` ファイルを `/etc/driver/drv/scsi_vhci.conf` ファイルにコピーします。
- 3 `/etc/driver/drv/scsi_vhci.conf` ファイルを編集して、ベンダー ID および製品 ID のエントリを追加します。

ベンダー ID および製品 ID は、デバイスが SCSI INQUIRY データで返すベンダーおよび製品の識別文字列です。ベンダー ID は長さが 8 文字である必要があります。末尾の文字がスペースであっても 8 つすべての文字を指定する必要があります。

製品 ID は 16 文字までの長さが可能です。

```
scsi-vhci-failover-override =
"VendorID1ProductID1", "f_sym",
"VendorID2ProductID2", "f_sym",
...
"VendorIDnProductIDn", "f_sym";
```

エントリは「;」文字 (コンマ) で区切られ、最後のベンダー/製品エントリは「;」文字 (セミコロン) で終了します。

たとえば、「ACME」というベンダーから製品IDが「MSU」のデバイスを追加し、「XYZ」というベンダーから製品IDが「ABC」のデバイスを追加する場合、`/etc/driver/drv/scsi_vhci.conf` ファイルに次の行を追加します。

```
scsi-vhci-failover-override =
"ACME  MSU", "f_sym",
"XYZ   ABC", "f_sym";
```

- 4 `/etc/driver/drv/scsi_vhci.conf` ファイルを保存して終了します。
- 5 リブートおよび構成プロセスを開始します。
 

```
# stmsboot -u
```

 リブートを確認するメッセージが表示されます。リブート中に、デバイス名の変更を反映するように `/etc/vfstab` ファイルおよびダンプ構成が更新されます。
- 6 必要な場合、[26 ページの「マルチパス化の有効化と無効化」](#)で説明されているように、デバイス名の更新を実行します。

## サードパーティー製ストレージデバイスの構成: デバイスの無効化

マルチパス化は、特定のベンダーID/製品IDの組み合わせを持つすべてのデバイスについて無効にできます。この除外は `scsi_vhci.conf` ファイルで指定されます。

### ▼ サードパーティー製デバイスを無効にする方法

- 1 管理者になります。
- 2 `/kernel/drv/scsi_vhci.conf` ファイルを `/etc/driver/drv/scsi_vhci.conf` ファイルにコピーします。
- 3 ベンダーIDおよび製品IDエントリを `/etc/driver/drv/scsi_vhci.conf` ファイルに追加します。

ベンダーIDおよび製品IDは、デバイスがSCSI INQUIRY データで返すベンダーおよび製品の識別文字列です。ベンダーIDは長さが8文字である必要があります。末尾の文字がスペースであっても8つすべての文字を指定する必要があります。製品IDは16文字までの長さが可能です。

```
scsi-vhci-failover-override =
"VendorID1ProductID1", "NONE",
"VendorID2ProductID2", "NONE",
...
"VendorIDnProductIDn", "NONE";
```

前の例にあるエントリは「,」文字(コンマ)で区切られ、最後のベンダー/製品エントリは「;」文字(セミコロン)で終了します。たとえば、「ACME」というベンダーから製品IDが「MSU」のデバイスを追加し、「XYZ」というベンダーから製品IDが「ABC」のデバイスを追加する場合、`/etc/driver/drv/scsi_vhci.conf` ファイルに次の行を追加します。

```
scsi_vhci-failover-override =
"ACME    MSU", "NONE",
"XYZ     ABC", "NONE";
```

- 4 `scsi_vhci.conf` ファイルを保存して終了します。

- 5 リブートおよび構成プロセスを開始します。

```
# stmsboot -u
```

リブートを確認するメッセージが表示されます。リブート中に、デバイス名の変更を反映するように `/etc/vfstab` ファイルおよびダンプ構成が更新されます。

- 6 必要な場合、[26 ページの「マルチパス化の有効化と無効化」](#)で説明されているように、デバイス名の更新を実行します。

## デバイス名の変更の表示

`stmsboot` コマンドに `-e`、`-d`、または `-u` オプションを付けて呼び出すことでマルチパス構成に変更が加えられたあと、マルチパス化されていないデバイス名とマルチパス化されたデバイス名とのマッピングを表示できます。マッピングを表示するには、マルチパス化されていないデバイス名とマルチパス化されたデバイス名の両方が存在する必要があります。

すべてのコントローラ上のデバイスのマッピングを表示します。例:

```
# stmsboot -L
non-STMS device name                STMS device name
-----
/dev/rdisk/c2t8d0                   /dev/rdisk/c10t500000E01046DEE0d0
/dev/rdisk/c2t0d0                   /dev/rdisk/c10t500000E01046B070d0
/dev/rdisk/c2t3d0                   /dev/rdisk/c10t20000020372A40AFd0
/dev/rdisk/c2t12d0                  /dev/rdisk/c10t500000E01046DEF0d0
/dev/rdisk/c2t11d0                  /dev/rdisk/c10t500000E01046E390d0
/dev/rdisk/c3t8d0                   /dev/rdisk/c10t500000E01046DEE0d0
/dev/rdisk/c3t0d0                   /dev/rdisk/c10t500000E01046B070d0
/dev/rdisk/c3t3d0                   /dev/rdisk/c10t20000020372A40AFd0
/dev/rdisk/c3t12d0                  /dev/rdisk/c10t500000E01046DEF0d0
/dev/rdisk/c3t11d0                  /dev/rdisk/c10t500000E01046E390d0
```

`-l` オプションは、指定されたコントローラについてのデバイスのマッピングのみを表示します。次の例は、コントローラ 3 のマッピングを表示します。

```
# stmsboot -l3
non-STMS device name          STMS device name
-----
/dev/rdsk/c3t8d0             /dev/rdsk/c10t500000E01046DEE0d0
/dev/rdsk/c3t0d0             /dev/rdsk/c10t500000E01046B070d0
/dev/rdsk/c3t3d0             /dev/rdsk/c10t20000020372A40AFd0
/dev/rdsk/c3t12d0            /dev/rdsk/c10t500000E01046DEF0d0
/dev/rdsk/c3t11d0            /dev/rdsk/c10t500000E01046E390d0
```

## 自動フェイルバックの構成

一部のストレージデバイスでは、アレイ構成の一部として、コントローラがプライマリおよびセカンダリとして構成されています。セカンダリパスはプライマリパスよりも低いパフォーマンスレベルで動作することがあります。マルチパスソフトウェアはプライマリパスを使用してストレージデバイスと接続し、セカンダリパスをスタンバイに維持します。

プライマリパスが故障した場合、マルチパスソフトウェアはすべてのI/Oトラフィックをセカンダリパスに自動的に送信し、プライマリパスをオフラインにします。このプロセスは「フェイルオーバー」処理と呼ばれます。プライマリパスに関連した故障が修復すると、マルチパスソフトウェアはすべてのI/Oトラフィックをプライマリパスに自動的に送信し、セカンダリパスを以前のようにスタンバイに維持します。このプロセスはフェイルバック処理と呼ばれます。

自動フェイルバック処理を無効にして、マルチパスソフトウェアがプライマリパスに自動的にフェイルバックしないようにできます。プライマリパスに関連した故障が修復したあと、`luxadm` コマンドを使用して手動フェイルバック処理を実行できます。詳細は、[luxadm\(1M\)](#) を参照してください。

### ▼ 自動フェイルバックの構成方法

- 1 管理者になります。
- 2 `/kernel/drv/scsi_vhci.conf` ファイルを `/etc/driver/drv/scsi_vhci.conf` ファイルにコピーします。
- 3 `/etc/driver/drv/scsi_vhci.conf` ファイル内で、`auto-failback` エントリを変更して自動フェイルバック機能を有効または無効にします。  

```
auto-failback="enable";

auto-failback="disable";
```
- 4 ファイルを保存して終了します。

5 システムをリブートします。

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



## マルチパス化デバイスの管理

---

この章では、マルチパス化デバイス情報を表示する方法と、特定のマルチパス化サポートのための使用可能な最適パスへの自動フェイルバックを構成する方法について説明します。

次の項目について説明します。

- 42 ページの「特定のターゲットポートに関連付けられたすべての LU を表示する方法」
- 46 ページの「マルチパス化サポートのための自動フェイルバックの構成」

## マルチパス化デバイスの管理

Solaris OS マルチパス化サポートは、`mpathadm` コマンドを使用して判別および構成します。`mpathadm` コマンドにより、ANSI 標準のマルチパス管理 API を介したマルチパス化管理が可能になります。バス、イニシエータポート、ターゲットポート、および LU を表すためにこの章で使用される用語は、T10 仕様と一貫性があります。

マルチパス化デバイスを管理するには、次のタスクを実行します。

## マルチパス化サポート情報の表示

`mpathadm` コマンドを使用して、マルチパス化サポート情報を表示し、マルチパス化の発見を管理することもできます。マルチパス化サポートおよびプロパティ情報は、マルチパス管理 API プラグインライブラリ名で識別され、これは `mpathadm` コマンドを使用して表示されます。

## ▼ マルチパス化サポート情報を表示する方法

- 1 管理者になります。
- 2 システムのマルチパス化サポートを識別します。

```
# mpathadm list mpath-support
mpath-support: libmpscsi_vhci.so
```

- 3 指定された **mpath** サポート名についてのマルチパス化サポートプロパティを表示します。

```
# mpathadm show mpath-support libmpscsi_vhci.so
mpath-support: libmpscsi_vhci.so
Vendor: Sun Microsystems
Driver Name: scsi_vhci
Default Load Balance: round-robin
Supported Load Balance Types:
    round-robin
    logical-block
Allows To Activate Target Port Group Access: yes
Allows Path Override: no
Supported Auto Failback Config: yes
Auto Failback: on
Failback Polling Rate (current/max): 0/0
Supported Auto Probing Config: 0
Auto Probing: NA
Probing Polling Rate (current/max): NA/NA
Supported Devices:
Vendor: SUN
Product: T300
Revision:
Supported Load Balance Types:
    round-robin

Vendor: SUN
Product: T4
Revision:
Supported Load Balance Types:
    round-robin

.
.
.
```

コマンド出力では、マルチパス化サポートソフトウェアによってサポートされるデバイス製品のリストも表示されます。マルチパス化サポート `libmpscsi_vhci.so` ライブラリファイルは、T10 ターゲットポートグループに準拠したデバイスをデフォルトでサポートします。

## ▼ 特定のイニシエータポートのプロパティを表示する方法

次のステップでは、イニシエータポートのプロパティを表示する方法について説明します。

- 1 イニシエータポートを一覧表示します。

```
# mpathadm list initiator-port
Initiator Port: iqn.1986-03.com.sun:01:ffffffffffff.4e94f9bd,4000002a00ff
Initiator Port: 210100e08ba41feb
Initiator Port: 210000e08b841feb
```

- 2 特定のイニシエータポートのプロパティを表示します。

```
# mpathadm show initiator-port 2000000173018713
Initiator Port: 210100e08ba41feb
Transport Type: Fibre Channel
OS Device File: /devices/pci@1,0/pci1022,7450@1/pci1077,141@2,1/fp@0,0
```

## ▼ 特定の LU 情報を表示する方法

mpathadm コマンドを使用して、LU のリストと、各 LUN のプロパティを表示します。表示される LU のリストには、特定の LU のプロパティを表示するために使用できる名前が含まれています。

- 1 マルチパス化された LU のリストを表示します。

```
# mpathadm list lu
/dev/rdsk/c0t600144F08069703400004E828EE10004d0s2
Total Path Count: 8
Operational Path Count: 8
/dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
Total Path Count: 8
Operational Path Count: 8
/dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
Total Path Count: 4
Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
Total Path Count: 4
Operational Path Count: 4
/dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
Total Path Count: 4
Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF44E233FCF0s2
Total Path Count: 4
Operational Path Count: 4
/dev/rdsk/c0t600A0B800026D63A0000A4974E23424Ed0s2
Total Path Count: 4
Operational Path Count: 4
.
.
.
```

- 2 リストからの LU 名を使用して、特定の LU についての構成情報を表示します。

```
# mpathadm show lu /dev/rdsk/c4t60003BA27D2120004204AC2B000DAB00d0s2
mpath-support: libmpscsi_vhci.so
Vendor: SUN
Product: Sun Storage NAS
Revision: 1.0
Name Type: unknown type
Name: 600144f08069703400004e828ee10004
```

```

Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:
  Initiator Port Name: 210100e08ba41feb
  Target Port Name: 2100001b329b6c3f
  Override Path: NA
  Path State: OK
  Disabled: no

  Initiator Port Name: 210100e08ba41feb
  Target Port Name: 2101001b32bb6c3f
  Override Path: NA
  Path State: OK
  Disabled: no

  Initiator Port Name: 210100e08ba41feb
  Target Port Name: 2100001b329b793c
  Override Path: NA
  Path State: OK
  Disabled: no

```

.  
.  
.

```

Target Port Groups:
  ID: 0
  Explicit Failover: no
  Access State: active optimized
  Target Ports:
    Name: 2100001b329b6c3f
    Relative ID: 1

    Name: 2101001b32bb6c3f
    Relative ID: 2

  ID: 1
  Explicit Failover: no
  Access State: standby
  Target Ports:
    Name: 2100001b329b793c
    Relative ID: 257

    Name: 2101001b32bb793c
    Relative ID: 256

```

## ▼ 特定のターゲットポートに関連付けられたすべてのLUを表示する方法

次のステップを使用して、ターゲットポートのパスを表示します。

### 1 LUのリストを表示します。

```

# mpathadm list lu
  /dev/rdisk/c0t600144F08069703400004E828EE10004d0s2

```

```

        Total Path Count: 8
        Operational Path Count: 8
/dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
        Total Path Count: 8
        Operational Path Count: 8
/dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
        Total Path Count: 4
        Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
        Total Path Count: 4
        Operational Path Count: 4
.
.
.

```

## 2 特定のLU情報を表示して、ターゲットポートを判別します。

```

# mpathadm show lu /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
Logical Unit: /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
mpath-support: libmpscsi_vhci.so
Vendor: SUN
Product: CSM200_R
Revision: 0660
Name Type: unknown type
Name: 600a0b800029065c00007cf54e234013
Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:
    Initiator Port Name: 210000e08b841feb
    Target Port Name: 200800a0b826d63b
    Override Path: NA
    Path State: OK
    Disabled: no

    Initiator Port Name: 210000e08b841feb
    Target Port Name: 200900a0b826d63b
    Override Path: NA
    Path State: OK
    Disabled: no
.
.
.

Target Port Groups:
    ID: 4
    Explicit Failover: yes
    Access State: standby
    Target Ports:
        Name: 200800a0b826d63b
        Relative ID: 0

        Name: 200800a0b826d63c
        Relative ID: 0

    ID: 14

```

```

Explicit Failover: yes
Access State: active
Target Ports:
  Name: 200900a0b826d63b
  Relative ID: 0

  Name: 200900a0b826d63c
  Relative ID: 0

```

### 3 特定のターゲットポート情報を表示します。

```

# mpathadm list lu -t 20030003ba27d212
mpath-support: libmpscsi_vhci.so
  /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
  Total Path Count: 4
  Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
  /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
  Total Path Count: 4
  Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
  /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
  Total Path Count: 4
  Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
  /dev/rdsk/c0t600A0B800029065C00007CF44E233FCFd0s2
  Total Path Count: 4
  Operational Path Count: 4
.
.
.

```

## ▼ 特定の名前を持つ LU を表示する方法

マルチパス化された LU について、パスおよびターゲットポートグループ情報を含む詳細情報を表示します。この情報の名前プロパティは、ハードウェアから導出し、このシステムによって使用されるこの LU の識別子を表します。名前が SCSI 照会の重要な製品データ (VPD) ページ 83h から導出される場合、名前タイプのプロパティは、SCSI 標準によって定義される関連した識別子タイプを表します。

### 1 マルチパス化された LU のリストを表示します。

```

# mpathadm list lu
  /dev/rdsk/c0t600144F08069703400004E828EE10004d0s2
  Total Path Count: 8
  Operational Path Count: 8
  /dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
  Total Path Count: 8
  Operational Path Count: 8
  /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
  Total Path Count: 4
  Operational Path Count: 4
  /dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
  Total Path Count: 4
  Operational Path Count: 4
  /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2

```

```

Total Path Count: 4
Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF44E233FCFd0s2
Total Path Count: 4
Operational Path Count: 4

```

```

.
.
.

```

## 2 選択された LU の構成情報を表示します。

```

# mpathadm show lu /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
Logical Unit: /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
mpath-support: libmpscsi_vhci.so
Vendor: SUN
Product: CSM200_R
Revision: 0660
Name Type: unknown type
Name: 600a0b800026d63a0000a4994e2342d4
Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:
Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

Target Port Groups:
ID: 3
Explicit Failover: yes
Access State: active
Target Ports:
Name: 200800a0b826d63b
Relative ID: 0

Name: 200800a0b826d63c

```

```
Relative ID: 0
```

```
ID: 13
Explicit Failover: yes
Access State: standby
Target Ports:
  Name: 200900a0b826d63b
  Relative ID: 0
  Name: 200900a0b826d63c
  Relative ID: 0
```

### 3 選択された LU 情報を表示します。

```
# mpathadm list lu -n 600a0b800026d63a0000a4994e2342d4
mpath-support: libmpscsi_vhci.so
                /dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2
                Total Path Count: 4
                Operational Path Count: 4
```

## マルチパス化サポートのための自動フェイルバックの構成

対称型デバイスでは、使用可能な最適パスへの自動フェイルバックが可能です。最初のパスでフェイルオーバーが発生した場合、待機パスは新しいオンラインパスになります。通常、待機パスは準最適なパスです。自動フェイルバックが有効な場合、最初のパスがオンラインに戻り、最初のパスへのフェイルオーバーが自動的に発生します。

### ▼ 特定のマルチパス化サポートのための自動フェイルバックを構成する方法

#### 1 管理者になります。

#### 2 サポートされるマルチバスドライバ情報を表示します。

```
# mpathadm list mpath-support
mpath-support: libmpscsi_vhci.so
```

#### 3 サポートされるマルチバスドライバの自動フェイルバックサポートを有効にします。

```
# mpathadm modify mpath-support -a on libmpscsi_vhci.so
```

#### 4 構成の変更を確認します。

```
# mpathadm show mpath-support libmpscsi_vhci.so
mpath-support: libmpscsi_vhci.so
                Vendor: Sun Microsystems
```

```

Driver Name:  scsi_vhci
Default Load Balance:  round-robin
Supported Load Balance Types:
    round-robin
    logical-block
Allows To Activate Target Port Group Access:  yes
Allows Path Override:  no
Supported Auto Failback Config:  1
Auto Failback:  on
Failback Polling Rate (current/max):  0/0
Supported Auto Probing Config:  0
Auto Probing:  NA
Probing Polling Rate (current/max):  NA/NA
Supported Devices:
    Vendor:  SUN
    Product:  T300
    Revision:
    Supported Load Balance Types:
        round-robin

    Vendor:  SUN
    Product:  T4
    Revision:
    Supported Load Balance Types:
        round-robin
.
.
.

```

---

注 - `mpathadm modify` コマンド設定によって開始された自動表示は、システムが実行している間有効です。ただし、変更された設定を持続させるには、`/etc/driver/drv/scsi_vhci.conf` ファイルを更新する必要があります。[36 ページの「自動ファイルバックの構成」](#)を参照してください。

---

## ▼ LU をフェイルオーバーする方法

この操作は、次の2つのカテゴリのデバイスにのみ該当します。

- マルチパス化サポートによって認識およびサポートされる、専有のフェイルオーバーメカニズムを持つ非対称型デバイス
- T10 標準のターゲットポートグループサポート `libmpscsi_vhci.so` に準拠し、明示的な非対称型 LU アクセスのモードを提供するデバイス

### 1 マルチパス化された LU のリストを表示します。

```

# mpathadm list lu
/dev/rdsk/c0t600144F08069703400004E828EE10004d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdsk/c0t600144F08069703400004E8183DF0002d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdsk/c0t600A0B800026D63A0000A4994E2342D4d0s2

```

```

Total Path Count: 4
Operational Path Count: 4
/dev/rdsk/c0t600A0B800029065C00007CF54E234013d0s2
Total Path Count: 4
Operational Path Count: 4
/dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
Total Path Count: 4
Operational Path Count: 4
.
.
.

```

## 2 特定のLUの構成情報を表示します。

```

# mpathadm show lu /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
Logical Unit: /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
mpath-support: libmpscsi_vhci.so
Vendor: SUN
Product: CSM200_R
Revision: 0660
Name Type: unknown type
Name: 600a0b800026d63a0000a4984e234298
Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

Target Port Groups:
ID: 5
Explicit Failover: yes
Access State: active
Target Ports:
Name: 200800a0b826d63b

```

```

Relative ID: 0

Name: 200800a0b826d63c
Relative ID: 0

ID: 15
Explicit Failover: yes
Access State: standby
Target Ports:
  Name: 200900a0b826d63b
  Relative ID: 0

  Name: 200900a0b826d63c
  Relative ID: 0

```

### 3 LU フェイルオーバーを手動で強制実行します。

```
# mpathadm failover lu /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
```

この操作に成功した場合、デバイスのターゲットポートグループのアクセス状態は、論理ユニットのフェイルオーバーの結果、変更されます。

### 4 アクセス状態の変更を確認します。

```
# mpathadm show lu /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
```

```

Logical Unit: /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
  mpath-support: libmpscsi_vhci.so
  Vendor: SUN
  Product: CSM200_R
  Revision: 0660
  Name Type: unknown type
  Name: 600a0b800026d63a0000a4984e234298
  Asymmetric: yes
  Current Load Balance: round-robin
  Logical Unit Group ID: NA
  Auto Failback: on
  Auto Probing: NA

```

#### Paths:

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

```

```
Initiator Port Name: 210000e08b841feb
```

```

Target Port Name: 200900a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

```

Target Port Groups:

```

ID: 5
Explicit Failover: yes
Access State: standby
Target Ports:
  Name: 200800a0b826d63b
  Relative ID: 0

  Name: 200800a0b826d63c
  Relative ID: 0

```

```

ID: 15
Explicit Failover: yes
Access State: active
Target Ports:
  Name: 200900a0b826d63b
  Relative ID: 0

  Name: 200900a0b826d63c
  Relative ID: 0

```

## ▼ LUパスを有効にする方法

LUへのパスが無効な場合、`enable` コマンドによってパスは元の有効な状態に変更されます。イニシエータポート名、ターゲットポート名、およびLUを使用して、完全なパス名を指定する必要があります。変更を確認するには、論理ユニットに対して `show` コマンドを実行します。

### 1 マルチパス化されたLUのリストを表示します。

```

# mpathadm list lu
/dev/rds/c0t600144F08069703400004E828EE10004d0s2
  Total Path Count: 8
  Operational Path Count: 8
/dev/rds/c0t600144F08069703400004E8183DF0002d0s2
  Total Path Count: 8
  Operational Path Count: 8
/dev/rds/c0t600A0B800026D63A0000A4994E2342D4d0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rds/c0t600A0B800029065C00007CF54E234013d0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rds/c0t600A0B800026D63A0000A4984E234298d0s2
  Total Path Count: 4
  Operational Path Count: 4
/dev/rds/c0t600A0B800029065C00007CF44E233FCFd0s2
  Total Path Count: 4
  Operational Path Count: 4
.
.
.

```

## 2 選択された LU の構成情報を表示します。

```
# mpathadm show lu
Logical Unit: /dev/rdsk/c0t600A0B800026D63A0000A4984E234298d0s2
mpath-support: libmpscsi_vhci.so
Vendor: SUN
Product: CSM200_R
Revision: 0660
Name Type: unknown type
Name: 600a0b800026d63a0000a4984e234298
Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

Target Port Groups:
ID: 5
Explicit Failover: yes
Access State: standby
Target Ports:
  Name: 200800a0b826d63b
  Relative ID: 0

  Name: 200800a0b826d63c
  Relative ID: 0

ID: 15
Explicit Failover: yes
Access State: active
Target Ports:
  Name: 200900a0b826d63b
  Relative ID: 0

  Name: 200900a0b826d63c
  Relative ID: 0
```

### 3 LU パスを有効にします。

```
# mpathadm enable path -i 210000e08b841feb -t 200900a0b826d63b \  
-l /dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2
```

## ▼ LU パスを無効にする方法

この操作は LU のパスの動作状態にかかわらず、LU のパスを使用不能にします。

---

注 - 無効な状態は、リブートの前後で持続しません。次回のブート手順の前にパスが動作中であった場合、パスはデフォルトで有効にされます。指定されたパスが、残っている最後の動作中のパスであるとき、この操作は許可されません。

---

### 1 マルチパス化された LU のリストを表示します。

```
# mpathadm list lu  
/dev/rdisk/c0t600144F08069703400004E828EE10004d0s2  
Total Path Count: 8  
Operational Path Count: 8  
/dev/rdisk/c0t600144F08069703400004E8183DF0002d0s2  
Total Path Count: 8  
Operational Path Count: 8  
/dev/rdisk/c0t600A0B800026D63A0000A4994E2342D4d0s2  
Total Path Count: 4  
Operational Path Count: 4  
/dev/rdisk/c0t600A0B800029065C00007CF54E234013d0s2  
Total Path Count: 4  
Operational Path Count: 4  
/dev/rdisk/c0t600A0B800026D63A0000A4984E234298d0s2  
Total Path Count: 4  
Operational Path Count: 4  
/dev/rdisk/c0t600A0B800029065C00007CF44E233FCFd0s2  
Total Path Count: 4  
Operational Path Count: 4  
.  
.  
.
```

### 2 特定の LU の構成情報を表示します。

```
# mpathadm show lu /dev/rdisk/c0t600A0B800029065C00007CF34E233F89d0s2  
Logical Unit: /dev/rdisk/c0t600A0B800029065C00007CF34E233F89d0s2  
mpath-support: libmpscsi_vhci.so  
Vendor: SUN  
Product: CSM200_R  
Revision: 0660  
Name Type: unknown type  
Name: 600a0b800029065c00007cf34e233f89  
Asymmetric: yes  
Current Load Balance: round-robin  
Logical Unit Group ID: NA  
Auto Failback: on  
Auto Probing: NA  
  
Paths:  
Initiator Port Name: 210000e08b841feb
```

```

Target Port Name: 200800a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63b
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200800a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

```

```

Initiator Port Name: 210000e08b841feb
Target Port Name: 200900a0b826d63c
Override Path: NA
Path State: OK
Disabled: no

```

```

Target Port Groups:
ID: 8
Explicit Failover: yes
Access State: standby
Target Ports:
    Name: 200800a0b826d63b
    Relative ID: 0

    Name: 200800a0b826d63c
    Relative ID: 0

ID: 18
Explicit Failover: yes
Access State: active
Target Ports:
    Name: 200900a0b826d63b
    Relative ID: 0

    Name: 200900a0b826d63c
    Relative ID: 0

```

- 3 イニシエータポートおよびターゲットポートの名前を選択します。
- 4 選択されたLUパスを無効にします。

```

# mpathadm disable path -i 210000e08b841feb -t 200900a0b826d63b \
-l /dev/rdisk/c0t600A0B800029065C00007CF34E233F89d0s2

```



## ファブリック接続されたデバイスの構成

---

この章では、ファブリック接続された (FC) デバイスを構成するために使用するステップの概要について説明します。Oracle Solaris OS では、FCAL、ファブリック、およびポイントツーポイントで接続されたデバイスがシステムから自動的に使用できます。この機能は、以前の Solaris リリースで実行されていた以前のバージョンの SAN Foundation ソフトウェアとは異なります。これらのバージョンでは、ファブリック接続されたデバイスのホスト上での使用条件を得るには、手動での構成ステップが必要でした。ファブリック接続されたデバイスを手動で構成する方法を見つけるには、付録 A 「ファブリック接続されたデバイスの手動構成」を参照してください。

次の項目について説明します。

- 55 ページの「FC デバイスの考慮事項」
- 56 ページの「FC デバイスの追加」
- 57 ページの「SPARC でのファブリックブートデバイスの構成」

### FC デバイスの考慮事項

Solaris I/O マルチパス化機能を構成する前に、次の考慮事項を確認してください。

- ストレージおよびスイッチについてのベンダー固有のドキュメントに従ってポートおよびゾーンを構成します。
- デバイスを手動で構成する必要はありません。
- LUN マスクによって、特定の LUN が特定のシステムから見えるようになります。マスクについて説明されているベンダー固有のストレージのドキュメントを参照してください。
- マルチパス化機能を持つまたは持たないアレイおよびその他のストレージデバイスを SAN に接続します。Solaris マルチパス化は、製品名にバンドルされる関連のアプリケーションです。

- SAN ブートプロセスを管理する STMS ブートユーティリティーが Solaris I/O マルチパス化機能に含まれています。stmsboot コマンドを発行すると、マルチパス化ソフトウェアを有効または無効にしたときにデバイス名の変更を反映するよう /etc/vfstab およびダンプ構成の自動更新がアクティブにされます。このソフトウェアは SPARC ベースシステム上のデバイスではデフォルトで無効化されており、x86 ベースシステム上のデバイスではデフォルトで有効化されていることに注意してください。

## FCデバイスの追加

次の表は、FC デバイスを追加および削除するときに使用される一般的なコマンドを一覧表示したものです。

コマンド	説明	参照先
cfgadm および cfgadm_fp	デバイスおよび FC デバイスを動的に再構成します。これらのコマンドは、SAN のストレージデバイスを構成するためにもっとも頻繁に使用されます。	<a href="#">cfgadm(1M)</a> および <a href="#">cfgadm_fp(1M)</a>
format	システムに接続されているデバイスを識別し、ディスクにラベルを付ける機能を提供します。	<a href="#">format(1M)</a>
luxadm	ストレージデバイスおよび FC_AL デバイスを管理します。	<a href="#">luxadm(1M)</a>

注 - マルチパス化機能が有効なときに format コマンドを使用すると、各 LUN についてデバイス識別子のインスタンスが 1 つのみ表示されます。マルチパス化機能を有効にしない場合、各パスにつき 1 つの識別子が表示されます。

### ▼ FC デバイスを追加する方法

- 1 FC デバイスに目的の LUN を作成します。
- 2 必要な場合、FC デバイスに HBA 制御のための LUN マスクを適用します。
- 3 ストレージデバイスをシステムに接続します。
- 4 必要な場合、FC デバイスのスイッチにポートベースのゾーンまたは WWN ゾーンを作成します。

- 5 ストレージデバイスのLUNまたはディスクグループ上で使用可能な既存のファイルシステムをマウントします。

## SPARCでのファブリックブートデバイスの構成

Oracle Solaris OS を実行中の SPARC サーバーがファブリックディスクデバイスからブートするように設定できます。

### ファブリックブートデバイスの考慮事項

以前の Solaris OS リリースで内部ブートデバイスにアクセスできたように、ファブリックブートデバイスには Oracle Solaris インストール方法を通じてアクセスできません。詳細は、Oracle Solaris インストールガイドを参照してください。ファブリックブートデバイスを有効にする際は次の点を考慮してください。

- 次のアクションによって、ブートデバイスへの干渉を最小限に抑えてください。
  - ブートデバイスが、過剰に利用されるターゲットまたは LUN でないようにする
  - ターゲットまたは LUN にアプリケーションおよびソフトウェアをインストールすることを避ける
  - ホストとファブリックデバイス間の物理的な距離と、ホップ数を削減する
- ファブリックブート手順を開始する前に、ブートディスクをボリュームマネージャーコントロールから削除します。
- 最新の HBA fcode およびドライバがシステムの HBA 用にロードされていることを確認します。
- ブートデバイスについてマルチパス化を使用する場合、第3章「Solaris I/O マルチパス化機能の構成」に記載されているように `stmsboot` コマンドを使用します。



## Solaris iSCSI イニシエータの構成

---

この章では、Oracle Solaris リリースで Solaris iSCSI イニシエータを構成する方法について説明します。iSCSI イニシエータの構成に関連する手順については、61 ページの「iSCSI イニシエータタスクの構成」を参照してください。

次の項目について説明します。

- 59 ページの「Oracle Solaris iSCSI の技術 (概要)」
- 62 ページの「推奨される iSCSI 構成方法」
- 61 ページの「iSCSI イニシエータタスクの構成」
- 68 ページの「iSCSI ベースのストレージネットワークにおける認証の構成」
- 74 ページの「Oracle Solaris での iSCSI マルチパスデバイスの設定」
- 78 ページの「iSCSI 構成の監視」
- 80 ページの「iSCSI イニシエータと iSCSI ターゲットのパラメータの変更」
- 86 ページの「iSCSI 構成に関する問題のトラブルシューティング」

COMSTAR による Oracle Solaris iSCSI ターゲットの構成についての詳細は、『Oracle Solaris 11.1 の管理: デバイスとファイルシステム』の第 11 章「COMSTAR を使用したストレージデバイスの構成 (タスク)」を参照してください。

iSCSI ディスクのインストールおよびブートについての詳細は、『Oracle Solaris 11.1 システムのインストール』の「GUI インストールを実行する方法」の手順 7 を参照してください。

### Oracle Solaris iSCSI の技術 (概要)

iSCSI は Internet SCSI (Small Computer System Interface) の略語であり、データストレージサブシステムを結合するための、インターネットプロトコル (IP) ベースのストレージネットワーク標準です。

iSCSI プロトコルを使用すると、SCSI コマンドが IP ネットワーク経由で転送されるため、ユーザーはあたかもブロックデバイスがローカルシステムに接続されているかのようにネットワーク経由でブロックデバイスにアクセスできます。

既存の TCP/IP ネットワーク内のストレージデバイスを使用する場合は、次の解決方法を利用できます。

- iSCSI ブロックデバイスまたはテープ - SCSI コマンドとデータをブロックレベルから IP パケットへと変換します。あるシステムと、テープデバイスやデータベースなどのターゲットデバイスとの間で、ブロックレベルのアクセスが必要になる場合には、ネットワーク内で iSCSI を使用することをお勧めします。ブロックレベルデバイスへのアクセスにはロックがかからないため、iSCSI ターゲットデバイスなどのブロックレベルデバイスに複数のユーザーやシステムがアクセスできます。
- NFS - ファイルデータを IP 経由で転送します。ネットワーク内で NFS を使用する利点は、ファイルデータを複数のシステム間で共有できることにあります。NFS 環境で利用可能なデータに多数のユーザーがアクセスする場合、必要に応じてファイルデータへのアクセスにロックがかかります。

Oracle Solaris で iSCSI ターゲットおよびイニシエータを使用する利点を次に示します。

- iSCSI プロトコルは、既存の Ethernet ネットワーク上で動作します。
  - サポートされている任意のネットワークインタフェースカード (NIC)、Ethernet ハブ、または Ethernet スイッチを使用できます。
  - 1 つの IP ポートから複数の iSCSI ターゲットデバイスを処理できます。
  - IP ネットワークの既存のインフラストラクチャーや管理ツールを使用できます。
- クライアントに接続可能な既存のファイバチャネルデバイスを利用でき、ファイバチャネル HBA の費用がかかりません。また、専用のアレイを持つシステムが、複製されたストレージを Oracle Solaris ZFS または UFS ファイルシステムでエクスポートすることも可能になりました。
- このプロトコルは、適切なハードウェアを備えたファイバチャネル SAN (Storage Area Network) 環境または iSCSI SAN 環境への接続に使用できます。

Oracle Solaris での iSCSI イニシエータソフトウェア使用時の現時点における制限や制約を、次に示します。

- SLP を使用する iSCSI デバイスは、現時点ではサポートされていません。
- iSCSI ターゲットをダンプデバイスとして構成することはできません。
- 既存のネットワーク経由で大量のデータを転送すると、パフォーマンスに影響する可能性があります。

## Oracle Solaris iSCSI のソフトウェア要件およびハードウェア要件の識別

- Oracle Solaris ストレージソフトウェアおよびデバイス

- ストレージデバイスを提供するシステム用の group/feature/storage-server ソフトウェアパッケージ
- iSCSI 管理ユーティリティ用の system/storage/iscsi/iscsi-initiator ソフトウェアパッケージ
- サポートされている任意の NIC

## iSCSI イニシエータタスクの構成

ここでは、iSCSI イニシエータの構成に関連する一般的なタスクのリストを示します。一部のタスクはネットワーク構成の必要に応じたオプションです。次の一部のリンクは、ネットワーク構成およびターゲット構成を説明する別のドキュメントに移動します。

- 60 ページの「Oracle Solaris iSCSI のソフトウェア要件およびハードウェア要件の識別」
- 『Oracle Solaris 11.1 ネットワークの構成と管理』
- 『Oracle Solaris 11.1 の管理: デバイスとファイルシステム』の第 11 章「COMSTAR を使用したストレージデバイスの構成 (タスク)」
- 63 ページの「動的または静的ターゲット発見の構成」
- 64 ページの「iSCSI イニシエータを構成する方法」
- 66 ページの「iSCSI ディスクにアクセスする方法」
- 68 ページの「iSCSI ベースのストレージネットワークにおける認証の構成」
- 74 ページの「Oracle Solaris での iSCSI マルチパスデバイスの設定」
- 78 ページの「iSCSI 構成の監視」

## iSCSI 関連の用語

iSCSI ターゲットおよびイニシエータを構成する前に、次の用語を確認してください。

用語	説明
発見	利用可能なターゲットの一覧をイニシエータに提供するプロセス。

用語	説明
発見方式	<p>iSCSI ターゲットを発見するための方法。現時点では次の3つの方法を使用できます。</p> <ul style="list-style-type: none"> <li>■ Internet Storage Name Service (iSNS) - 1つ以上の iSNS サーバーと対話することで可能性のあるターゲットを発見します。</li> <li>■ SendTargets - <i>discovery-address</i> を使用することによって、可能性のあるターゲットが発見されます。</li> <li>■ 静的 - 静的なターゲットアドレスが構成されます。</li> </ul>
イニシエータ	iSCSI ターゲットに対する SCSI 要求を発行するドライバ。
イニシエータグループ	イニシエータのセット。イニシエータグループが LU と関連付けられている場合、そのグループのイニシエータだけが LU にアクセスできます。
iqn または eui アドレス形式	<p>iqn (iSCSI 修飾名) アドレスは、iSCSI ネットワーク内のデバイスの一意の識別子で、書式は <i>iqn.date.authority:uniqueid</i> です。iSCSI イニシエータまたはターゲットは、その初期化時に IQN 名が自動的に割り当てられます。</p> <p>eui (拡張一意識別子) アドレスは、16 桁の 16 進数で構成され、SCSI と InfiniBand の両方の標準で使用される GUID のクラスを識別します。SRP デバイスでは、eui アドレス形式を使用します。</p>
論理ユニット (LU)	ストレージシステム内の一意の番号が付けられたコンポーネント。LU が1つ以上の SCSI ターゲットと関連付けられているときは、1つ以上の SCSI イニシエータを使用してターゲットにアクセスできます。論理ユニットは論理ユニット番号 (LUN) で識別されます。
ターゲットデバイス	iSCSI ストレージコンポーネント。
ターゲットグループ	ターゲットのセット。1つのターゲットグループ内のすべてのターゲットに対して LU を使用可能にできます。
ターゲットポータルグループ	特定の iSCSI ターゲットが待機するインタフェースを決定する IP アドレスのリスト。TPG には、IP アドレスと TCP ポート番号が含まれます。

## 推奨される iSCSI 構成方法

ネットワーク内の iSCSI デバイスを構成する前に、次の iSCSI の推奨事項を確認してください。

- デバイス

- 可用性を高めるために、マルチパス化されたデバイスパスの使用を検討してください。
- セッションあたりの複数接続 (MCS) のサポートにより、同じ iSCSI セッションでイニシエータからターゲットへの複数の TCP/IP 接続が可能です。
- ネットワーク
  - iSCSI デバイスを高速 (gigE 以上) の専用ネットワークに配置します。
  - 可能であれば、各 Ethernet トランザクションでより多くのデータを転送してフレームの数を削減するために、ジャンボフレームを使用してください。
  - ギガビットネットワークインフラストラクチャでは、CAT6 規格のケーブルを使用します。
  - iSCSI ストレージネットワークをローカルエリアネットワークのトラフィックから分離します
  - TCP/IP スタック内で複数のスレッドを使用するために、複数のセッションまたは接続を構成します
  - Nagle アルゴリズムを無効にするなどの TCP チューニングを検討します
- セキュリティー
  - 物理的なセキュリティーに加えて、各ホストが独自のパスワードを持つよう CHAP 認証を使用します。
  - iSNS ターゲット発見ドメインの使用を考慮します。これは、独自のアクセス制御が有効でないターゲットにアクセス制御を提供し、一方で各イニシエータのログオンプロセスを、ネットワーク内で使用可能なターゲットの関連するサブセットに制限することにより、セキュリティーを強化します。

## 動的または静的ターゲット発見の構成

デバイス発見を実行するために、動的デバイス発見方式のいずれかを構成するか、または静的 iSCSI イニシエータターゲットを使用するかを決定します。

- 動的デバイス発見 - 次の 2 つの動的デバイス発見方式を使用できます。
  - SendTargets - ファイバチャネルブリッジへの iSCSI などのように、iSCSI ノードが多数のターゲットを公開している場合、その iSCSI ノードに IP アドレスとポートの組み合わせを提供でき、その iSCSI イニシエータは SendTargets 機能を使ってデバイス発見を実行できるようになります。
  - iSNS - iSNS (Internet Storage Name Service) を使用すると、できるかぎり少ない構成情報で、iSCSI イニシエータがアクセス権を持つターゲットを発見できます。また、ストレージノードの動作状態が変更されたときに iSCSI イニシエータに通知する状態変更通知機能もあります。iSNS 発見方式を使用するために、iSNS サーバーのアドレスとポートの組み合わせを指定して、デバイス発見を実行するために指定した iSNS サーバーを iSCSI イニシエータで照会でき

るようにすることができます。iSNS サーバーのデフォルトポートは 3205 です。iSNS の詳細については、RFC 4171 を参照してください。

<http://www.ietf.org/rfc/rfc4171.txt>

iSNS 発見サービスは、ネットワーク上のすべてのターゲットを発見するための管理モデルを提供します。

Oracle Solaris での iSNS サポートの設定についての詳細は、『Oracle Solaris 11.1 の管理: デバイスとファイルシステム』の第 12 章「Oracle Solaris Internet Storage Name Service (iSNS) の構成と管理」を参照してください。

- 静的デバイス発見 - iSCSI ノードのターゲットが少数である場合や、イニシエータのアクセス対象となるターゲットを制限する場合には、次の静的ターゲットアドレス命名規則を使用して、*target-name* を静的に構成できます。

*target,target-address[:port-number]*

また、アレイの管理ツールから静的ターゲットアドレスを決定できます。

---

注 - 単一の iSCSI ターゲットが静的デバイス発見方式と動的デバイス発見方式の両方によって発見されるように構成しないでください。発見方式を重複して使用すると、イニシエータが iSCSI ターゲットデバイスと通信するときのパフォーマンスが低下する可能性があります。

---

## ▼ iSCSI イニシエータを構成する方法

イニシエータの構成処理には、iSCSI ターゲット発見方式の特定が含まれます。これにより、使用可能なターゲットのリストがイニシエータに提示されます。iSCSI ターゲットは、静的、SendTargets、または iSNS 動的発見用に構成できます。SendTargets オプションを使用した動的発見は、iSCSI からファイバチャネルへのブリッジなどを經由して、大量のターゲットにアクセスする iSCSI イニシエータに最適な構成です。SendTargets 動的発見では、iSCSI イニシエータによるターゲット発見の実行のために、iSCSI ターゲットの IP アドレスとポートの組み合わせが必要です。もっとも一般的な発見方式は、SendTargets です。

ターゲット発見方式の構成時に、選択した方式に応じて次の情報を指定する必要があります。

- SendTargets - ターゲットの IP アドレス
- iSNS - iSNS サーバーのアドレス
- 静的 - ターゲットの IP アドレスとターゲット名

ターゲット発見方式の詳細は、63 ページの「動的または静的ターゲット発見の構成」を参照してください。

### 1 iSCSI イニシエータサービスを有効にします。

```
initiator# svcadm enable network/iscsi/initiator
```

- ターゲットを提供しているサーバーにログインする際、ターゲットの名前と IP アドレスを確認します。

```
target# ipadm show-addr
ADDRESS          TYPE      STATE      ADDR
lo0/v4           static   ok         127.0.0.1/8
e1000g0/_b      dhcp     ok         1.2.3.4/24
lo0/v6           static   ok         ::1/128
e1000g0/_a      addrconf ok         fe80::123:1234:fe27:360c/10
target# itadm list-target -v
TARGET NAME                                STATE    SESSIONS
iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-c3b36c123456  online  0
  alias:                                     -
  auth:                                       none (defaults)
  targetchapuser:                             -
  targetchapsecret:                           unset
  tpg-tags:                                    default
```

- ターゲットが静的に発見されるように構成します。

```
initiator# iscsiadm add static-config iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-c3b36c123456,
1.2.3.4
```

- 静的構成情報を確認します。

```
initiator# iscsiadm list static-config
Static Configuration Target: iqn.1986-03.com.sun:02:73d12edc-9bb9-cb44-efc4-c3b36c123456,
1.2.3.4:3260
```

発見方式が有効化されるまで iSCSI 接続は起動されません。次の手順を参照してください。

- 次のいずれかのターゲット発見方式を構成します。

- 動的に発見される (SendTargets) ターゲットを構成した場合は、SendTargets 発見方式を構成します。

```
initiator# iscsiadm add discovery-address 1.2.3.4
```

- 動的に発見される (iSNS) ターゲットを構成した場合は、iSNS 発見方式を構成します。

```
initiator# iscsiadm add isns-server 1.2.3.4
```

- 次のいずれかのターゲット発見方式を有効にします。

- 動的に発見される (SendTargets) ターゲットを構成した場合は、SendTargets 発見方式を有効にします。

```
initiator# iscsiadm modify discovery --sendtargets enable
```

- 動的に発見される (iSNS) ターゲットを構成した場合は、iSNS 発見方式を有効にします。

```
initiator# iscsiadm modify discovery --iSNS enable
```

- 静的ターゲットを構成した場合は、静的ターゲット発見方式を有効にします。

```
initiator# iscsiadm modify discovery --static enable
```

- 7 必要に応じて、`/dev` 名前空間を再構成して、iSCSI ディスクを認識させます。

```
initiator# devfsadm -i iscsi
```

## ▼ iSCSI ディスクにアクセスする方法

Oracle Solaris iSCSI イニシエータによってデバイスが発見されると、ログインネゴシエーションが自動的に発生します。Oracle Solaris iSCSI ドライバは、利用可能な LU の個数を判断し、デバイスノードを作成します。この時点で、iSCSI デバイスをほかのすべての SCSI デバイスと同様に扱えます。

ZFS ストレージプールを LU 上に作成してから、ZFS ファイルシステムを作成できます。

ローカルシステム上で iSCSI ディスクを表示するには、`format` ユーティリティーを使用します。

- 1 `format` の出力で iSCSI LU の情報を確認します。

```
initiator# format
0. c0t600144F0B5418B0000004DDAC7C10001d0 <SUN-COMSTAR-1.0 cyl 1022 alt 2 hd 128 sec 32>
   /scsi_vhci/disk@g600144f0b5418b0000004ddac7c10001
1. c8t0d0 <Sun-STK RAID INT-V1.0 cyl 17830 alt 2 hd 255 sec 63>
   /pci@0,0/pci10de,375ef/pci108e,286@0/disk@0,0
2. c8t1d0 <Sun-STK RAID INT-V1.0-136.61GB>
   /pci@0,0/pci10de,375ef/pci108e,286@0/disk@1,0
3. c8t2d0 <Sun-STK RAID INT-V1.0-136.61GB>
   /pci@0,0/pci10de,375ef/pci108e,286@0/disk@2,0
4. c8t3d0 <Sun-STK RAID INT-V1.0 cyl 17830 alt 2 hd 255 sec 63>
   /pci@0,0/pci10de,375ef/pci108e,286@0/disk@3,0
Specify disk (enter its number): 0
selecting c0t600144F0B5418B0000004DDAC7C10001d0
[disk formatted]
```

上記の出力では、ディスク 0 は MPxIO の制御下にある iSCSI LU です。ディスク 1-4 はローカルディスクです。

- 2 ZFS ストレージプールと ZFS ファイルシステムを iSCSI LU 上に作成できます。

```
initiator# zpool create pool-name c0t600144F0B5418B0000004DDAC7C10001d0
initiator# zfs create pool-name/fs-name
```

ZFS ファイルシステムは、作成時に自動的にマウントされ、ブート時に再マウントされます。

## ▼ 発見されたターゲットデバイスを削除する方法

次のいずれかの操作を実行すると、関連するターゲットがログアウトされます。

- 発見アドレスを削除する
- iSNS サーバーを削除する

- 静的構成を削除する
- 発見方式を無効にする

たとえば、これらの関連付けられたターゲットが引き続き使用され、ファイルシステムがマウントされている場合は、これらのデバイスのログアウトが失敗し、アクティブターゲットリスト上に残ります。

このオプションの手順では、ユーザーが現在ログインしているローカルシステム上で、ターゲットデバイスへのアクセスがすでに構成されているものとします。

- 1 管理者になります。
- 2 (省略可能) 次のいずれかを使って iSCSI ターゲット発見方式を無効にします。
  - SendTargets 発見方式を無効にする必要がある場合は、次のコマンドを使用します。
 

```
initiator# iscsiadm modify discovery --sendtargets disable
```
  - iSNS 発見方式を無効にする必要がある場合は、次のコマンドを使用します。
 

```
initiator# iscsiadm modify discovery --iSNS disable
```
  - 静的ターゲット発見方式を無効にする必要がある場合は、次のコマンドを使用します。
 

```
initiator# iscsiadm modify discovery --static disable
```
- 3 次のいずれかを使って iSCSI デバイス発見エンTRIES を削除します。
  - iSCSI SendTargets 発見エンTRIES を削除します。
 

例:

```
initiator# iscsiadm remove discovery-address 1.2.3.4:3260
```
  - iSCSI iSNS 発見エンTRIES を削除します。
 

例:

```
# iscsiadm remove isns-server 1.2.3.4:3205
```
  - 静的 iSCSI 発見エンTRIES を削除します。
 

例:

```
initiator# iscsiadm remove static-config eui.5000ABCD78945E2B,1.2.3.4
```

---

注-使用中の論理ユニット (LU) が関連付けられた発見エントリーを無効化または削除しようとする、次のメッセージが表示され、無効化または削除操作が失敗します。

logical unit in use

このエラーが発生した場合は、LU 上の関連付けられたすべての入出力を停止し、ファイルシステムのマウント解除を行います。そのあとで無効化または削除の操作を再び実行します。

---

#### 4 iSCSI ターゲットデバイスを削除します。

例:

```
target# itadm delete-target target-IQN
```

ターゲットが引き続きオンラインまたはビジー状態にある場合、このコマンドがエラーになることがあります。

```
The target is online or busy. Use the -f (force) option, or 'stmfadm offline-target  
iqn.1986-03.com.sun:02:99619b8a-a4dc-4cfb-93f0-ee3de1234567'  
itadm delete-target failed with error 16
```

## iSCSI ベースのストレージネットワークにおける認証の構成

iSCSI デバイスの認証設定はオプションです。

セキュリティー保護された環境では、信頼できるイニシエータだけがターゲットにアクセスできるため、認証は必要ありません。

セキュリティー保護の不十分な環境では、ターゲットは、接続要求が本当に指定されたホストからのものなのかを判断できません。そのような場合、ターゲットは、チャレンジハンドシェイク認証プロトコル (CHAP) を使ってイニシエータを認証できます。

CHAP 認証では「チャレンジ」と「応答」の概念が使用され、つまり、ターゲットがイニシエータに対して身元の証明を要求します。このチャレンジ/応答方式が機能するには、ターゲットがイニシエータの秘密鍵を知っており、かつイニシエータがチャレンジに応答するように設定されている必要があります。秘密鍵をアレイ上に設定する手順については、アレイのベンダーのドキュメントを参照してください。

iSCSI は、次の単方向および双方向の認証をサポートします。

- 「単方向認証」では、ターゲットがイニシエータの身元を認証できます。単方向認証はターゲットのために行われるもので、イニシエータを認証します。

- 「双方向認証」では、イニシエータがターゲットの身元を認証できるようにすることで、二次レベルのセキュリティーを追加します。双方向認証はイニシエータ主導で行われ、双方向認証を実行するかどうかはイニシエータにより制御されます。ターゲットに求められる設定は、CHAP ユーザーと CHAP シークレットを正しく定義することだけです。

## ▼ iSCSI イニシエータの CHAP 認証を構成する方法

この手順では、ユーザーが現在ログインしているローカルシステムから構成済みの iSCSI ターゲットデバイスに安全にアクセスするものとします。

- COMSTAR iSCSI ターゲットの CHAP 秘密鍵の長さは、12 文字以上かつ 255 文字以下にする必要があります。一部のイニシエータでは、サポートする秘密鍵の最大長がさらに短くなります。
- 自らの識別に CHAP を使用する各ノードは、ユーザー名とパスワードの両方が必要です。Oracle Solaris OS では、CHAP ユーザー名はデフォルトでイニシエータまたはターゲットのノード名（つまり、iqn 名）に設定されます。CHAP ユーザー名は、512 バイト未満の任意の長さのテキストに設定できます。512 バイトの長さの制限は Oracle Solaris の制限です。ただし、CHAP ユーザー名を設定しない場合は、初期化のときにノード名に設定されます。

集中管理認証サービスとして機能する他社製の RADIUS サーバーを使用することで、CHAP 秘密鍵の管理を簡略化できます。RADIUS を使用するとき、RADIUS サーバーにはノード名と、一致する CHAP 秘密鍵のセットが格納されます。認証を実行するシステムは、要求元のノード名および提供された要求者のシークレットを RADIUS サーバーに転送します。RADIUS サーバーは、秘密鍵が、指定されたモード名を認証するのに適切な鍵かどうかを確認します。iSCSI と iSER のどちらも、RADIUS サーバーの使用をサポートします。

他社製の RADIUS サーバーの使用に関する詳細は、71 ページの「[他社製の RADIUS サーバーを使用して iSCSI 構成内の CHAP 管理を単純化する](#)」を参照してください。

- 1 管理者になります。  
詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「[割り当てられている管理権限を使用する方法](#)」を参照してください。
- 2 単方向 CHAP または双方向 CHAP のどちらを構成するかを決定します。
  - 単方向認証（デフォルトの方式）では、ターゲットがイニシエータを検証できます。手順 3 から 5 のみを実行してください。
  - 双方向認証では、二次レベルのセキュリティーを追加する目的で、イニシエータがターゲットを認証することができます。手順 3 から 9 を実行してください。

**3 単方向 CHAP: イニシエータ上で秘密鍵を設定します。**

次のコマンドは、CHAP の秘密鍵を定義するためのダイアログを起動します。

```
initiator# iscsiadm modify initiator-node --CHAP-secret
Enter CHAP secret: *****
Re-enter secret: *****
```

**4 (オプション)単方向 CHAP: イニシエータ上で CHAP ユーザー名を設定します。**

デフォルトではイニシエータの CHAP ユーザー名は、イニシエータのノード名に設定されます。

次のコマンドを使用して、独自のイニシエータ CHAP ユーザー名を使用します。

```
initiator# iscsiadm modify initiator-node --CHAP-name new-CHAP-name
```

**5 単方向 CHAP – イニシエータ上で CHAP 認証を有効にします。**

```
initiator# iscsiadm modify initiator-node --authentication CHAP
```

CHAP 認証では、イニシエータノードにユーザー名とパスワードが必要です。ユーザー名は通常、渡されたユーザー名の秘密鍵をターゲットが検索するために使用されます。

**6 次のいずれかを選択して双方向 CHAP を有効または無効にします。**

- ターゲットとの接続用に双方向 CHAP を有効にします。

```
initiator# iscsiadm modify target-param -B enable target-iqn
```

- 双方向 CHAP を無効にします。

```
initiator# iscsiadm modify target-param -B disable target-iqn
```

**7 双方向 CHAP: ターゲットの認証方法を CHAP に設定します。**

```
initiator# iscsiadm modify target-param --authentication CHAP target-iqn
```

**8 双方向 CHAP: ターゲットを識別するターゲットデバイスの秘密鍵を設定します。**

次のコマンドは、CHAP の秘密鍵を定義するためのダイアログを起動します。

```
initiator# iscsiadm modify target-param --CHAP-secret target-iqn
```

**9 双方向 CHAP: ターゲットが代替 CHAP ユーザー名を使用する場合、ターゲットを識別する CHAP 名を設定します。**

デフォルトでは、ターゲットの CHAP 名はターゲット名に設定されます。

次のコマンドを使用して、ターゲットの CHAP 名を変更できます。

```
initiator# iscsiadm modify target-param --CHAP-name target-CHAP-name
```

## ▼ iSCSI ターゲットの CHAP 認証を構成する方法

この手順では、iSCSI ターゲットのあるローカルシステムにユーザーがログインしているものとします。

- 1 管理者になります。
- 2 単方向 CHAP または 双方向 CHAP のどちらを構成するかを決定します。
  - 単方向認証がデフォルト方式です。手順3から5のみを実行してください。
  - 双方向認証の場合。手順3から7を実行してください。
- 3 単方向/双方向 CHAP: イニシエータが CHAP を使用して自らを識別することを必要とするように、ターゲットを構成します。

```
target# itadm modify-target -a chap target-iqn
```

- 4 単方向/双方向 CHAP: イニシエータコンテキストを作成して、イニシエータについて記述します。

イニシエータのフルノード名とイニシエータの CHAP 秘密鍵を使って、イニシエータコンテキストを作成します。

```
target# itadm create-initiator -s initiator-iqn
Enter CHAP secret: *****
Re-enter secret: *****
```

- 5 単方向/双方向 CHAP: イニシエータが代替 CHAP 名を使用する場合、代替名を使ってイニシエータコンテキストを構成します。

```
target# itadm modify-initiator -u initiator-CHAP-name initiator-iqn
```

- 6 双方向 CHAP: このターゲットを識別するターゲットデバイスの秘密鍵を設定します。

```
target# itadm modify-target -s target-iqn
Enter CHAP secret: *****
Re-enter secret: *****
```

- 7 (オプション) 双方向 CHAP: ターゲットがターゲットノード名 (iqn) 以外の代替 CHAP ユーザー名を使用する場合、ターゲットを変更します。

```
target# itadm modify-target -u target-CHAP-name target-iqn
```

## 他社製の RADIUS サーバーを使用して iSCSI 構成内の CHAP 管理を単純化する

集中管理認証サービスとして機能する他社製の RADIUS サーバーを使用して、CHAP 秘密鍵の管理を簡略化できます。この方式では、各イニシエータノードでデフォルト

トの CHAP 名を使用することをお勧めします。すべてのイニシエータがデフォルトの CHAP 名を使用するという一般的な場合では、ターゲット上にイニシエータコンテキストを作成する必要はありません。

## ▼ RADIUS サーバーを iSCSI ターゲット用に構成する方法

集中管理認証サービスとして機能する他社製の RADIUS サーバーを使用して、CHAP 秘密鍵の管理を簡略化できます。この方式では、各イニシエータノードでデフォルトの CHAP 名を使用することをお勧めします。すべてのイニシエータがデフォルトの CHAP 名を使用するという一般的な場合では、ターゲット上にイニシエータコンテキストを作成する必要はありません。

この手順では、ユーザーが現在ログインしているローカルシステムから構成済みの iSCSI ターゲットデバイスに安全にアクセスするものとします。

- 1 管理者になります。
- 2 RADIUS サーバーの IP アドレスとポートを使用して、イニシエータノードを構成します。

デフォルトのポートは 1812 です。この構成は、ターゲットシステム上のすべての iSCSI ターゲットに対して 1 回で完了します。

```
initiator# itadm modify-defaults -r RADIUS-server-IP-address
Enter RADIUS secret: *****
Re-enter secret: *****
```

- 3 ターゲットシステムと RADIUS サーバー間の通信に使用される共有秘密鍵を構成します。

```
initiator# itadm modify-defaults -d
Enter RADIUS secret: *****
Re-enter secret: *****
```

- 4 RADIUS 認証を必要とするようにターゲットシステムを構成します。  
この構成は、個別のターゲットに実行することも、すべてのターゲットのデフォルトとして実行することもできます。

```
initiator# itadm modify-target -a radius target-iqn
```

- 5 次のコンポーネントを使用して RADIUS サーバーを構成します。
  - ターゲットノードの識別情報 (IP アドレスなど)
  - ターゲットノードが RADIUS サーバーとの通信に使用する共有秘密鍵
  - イニシエータの CHAP 名 (たとえば、iqn 名)、および認証に必要な各イニシエータの秘密鍵

## ▼ RADIUS サーバーを iSCSI イニシエータ用に構成する方法

集中管理認証サービスとして機能する他社製の RADIUS サーバーを使用して、CHAP 秘密鍵の管理を簡略化できます。この設定が役に立つのは、イニシエータが双方向 CHAP 認証を要求しているときだけです。それでも、イニシエータの CHAP 秘密鍵を指定する必要があります。ただし、RADIUS サーバーとの双方向認証を使用しているときは、イニシエータ上の各ターゲットの CHAP 秘密鍵を指定する必要はありません。RADIUS は、イニシエータまたはターゲットのいずれかの上に独立して構成できます。イニシエータおよびターゲットは、RADIUS を使用する必要はありません。

- 1 管理者になります。
- 2 RADIUS サーバーの IP アドレスとポートを使用して、イニシエータノードを構成します。  
デフォルトのポートは 1812 です。

```
# iscsiadm modify initiator-node --radius-server ip-address:1812
```

- 3 RADIUS サーバーの共有秘密鍵をイニシエータノードに構成します。  
iSCSI がサーバーとの対話処理を実行するためには、共有シークレットを使って RADIUS サーバーを構成する必要があります。

```
# iscsiadm modify initiator-node --radius-shared-secret
Enter secret:
Re-enter secret
```

- 4 RADIUS サーバーの使用を有効にします。  

```
# iscsiadm modify initiator-node --radius-access enable
```

- 5 CHAP 双方向認証に関するその他の設定を行います。  

```
# iscsiadm modify initiator-node --authentication CHAP
# iscsiadm modify target-param --bi-directional-authentication enable target-iqn
# iscsiadm modify target-param --authentication CHAP target-iqn
```

- 6 次のコンポーネントを使用して RADIUS サーバーを構成します。
  - このノードの識別情報 (IP アドレスなど)
  - このノードが RADIUS サーバーとの通信に使用する共有秘密鍵
  - ターゲットの CHAP 名 (たとえば、iqn 名)、および認証に必要な各ターゲットの秘密鍵

## Oracle Solaris iSCSI と RADIUS サーバーに関するエラーメッセージ

このセクションでは、Oracle Solaris iSCSI と RADIUS サーバーの構成に関するエラーメッセージについて説明します。復旧に役立つと思われるソリューションも提供します。

empty RADIUS shared secret

原因: イニシエータ上で RADIUS サーバーが有効になっているにもかかわらず、RADIUS の共有秘密鍵が設定されていません。

対処方法: RADIUS の共有秘密鍵をイニシエータに構成します。詳細は、[72 ページの「RADIUS サーバーを iSCSI ターゲット用に構成する方法」](#)を参照してください。

WARNING: RADIUS packet authentication failed

原因: イニシエータによる RADIUS データパケットの認証が失敗しました。このエラーが発生する可能性があるのは、イニシエータノード上に構成された共有秘密鍵が RADIUS サーバー上の共有秘密鍵と異なっている場合です。

対処方法: 正しい RADIUS 共有シークレットをイニシエータに構成し直します。詳細は、[72 ページの「RADIUS サーバーを iSCSI ターゲット用に構成する方法」](#)を参照してください。

## Oracle Solaris での iSCSI マルチパスデバイスの設定

マルチパス I/O (MPxIO) を使用すると、I/O デバイスの単一のインスタンスから、複数のホストコントローラインタフェース経由で I/O デバイスにアクセスできます。

Oracle Solaris で iSCSI マルチパス (MPxIO) デバイスを使用するときは、次のガイドラインを考慮してください。

- **Oracle Solaris iSCSI および MPxIO - MpxIO** は、iSCSI イニシエータの MS/T (Multiple Sessions per Target) を構成する Oracle Solaris iSCSI 構成での、ターゲットポート集約と可用性をサポートします。
  - 複数の NIC を集約およびフェイルオーバーする場合は、IP ネットワークマルチパス (IPMP) を使用してください。
  - iSCSI ホストの基本的な構成は、iSCSI トラフィック専用の 2 つの NIC を備えたサーバーです。NIC は IPMP を使用して構成されます。パフォーマンスを最適化するために、iSCSI 以外のトラフィック用に追加の NIC が用意されます。
  - アクティブマルチパスは、Oracle Solaris の iSCSI MS/T 機能、および IPMP 構成のフェイルオーバーと冗長性を使用する場合のみ実現できます。
    - IPMP 構成で 1 つの NIC に障害が発生すると、IPMP はフェイルオーバーの処理を行います。MPxIO ドライバは障害を検出しません。IPMP 以外の構成では、MPxIO ドライバは障害状態になり、パスがオフラインになります。
    - IPMP 構成で 1 つのターゲットポートに障害が発生すると、MPxIO ドライバは障害を検出し、フェイルオーバーの処理を行います。IPMP 以外の構成では、MPxIO ドライバは障害を検出し、フェイルオーバーの処理を行います。

MS/T の構成方法については、75 ページの「ターゲットの複数の iSCSI セッションを有効にする方法」を参照してください。IPMP の構成の詳細については、『Oracle Solaris 11.1 ネットワークパフォーマンスの管理』の第 6 章「IPMP の管理(タスク)」を参照してください。

- **Oracle Solaris iSCSI**、ファイバチャネル (FC)、および MPxIO - より複雑な iSCSI/FC 構成では、MPxIO ドライバは次のように動作します。
  - FC SAN にデュアル iSCSI-FC ブリッジが存在する場合、iSCSI はターゲットパスを MPxIO に提示します。MPxIO は、一意の SCSI/LU 識別子を照合し、それらが同一である場合は 1 つのパスを iSCSI ドライバに提示します。
  - iSCSI と FC の両方を使用してターゲットを接続する構成の場合、MPxIO ドライバは同じデバイスに対して異なるトランスポートを提供できます。この構成では、MPxIO は両方のパスを使用します。
  - iSCSI と FC を MPxIO で使用する場合は、`/etc/driver/drv/fp.conf` ファイルおよび `/driver/drv/iscsi.conf` ファイルの MPxIO パラメータが、サポートしたい MPxIO 構成と一致することを確認します。たとえば、`fp.conf` では、MPxIO を HBA 全体で有効にするのか、ポート単位で有効にするのかを決定できません。
- 他社製ハードウェアに関する注意事項 - 他社製の HBA が Oracle Solaris iSCSI および MPxIO での使用に適しているかどうかを確認します。  
他社製の HBA を使用する場合は、そのベンダーに `/driver/drv/scsi_vhci.conf` ファイルの対称オプション情報を問い合わせることが必要になる場合があります。

## ▼ ターゲットの複数の iSCSI セッションを有効にする方法

この手順を使用して、単一のターゲットに接続する複数の iSCSI セッションを作成できます。このシナリオは、ログインのリダイレクションをサポートするか、または同じターゲットポータルグループに含まれている複数のターゲットのポータルを使用する iSCSI ターゲットデバイスの場合に役に立ちます。Oracle Solaris の SCSI マルチパス (MPxIO) 機能を利用して、ターゲットごとに複数の iSCSI セッションを使用します。また、ホスト側の複数の NIC を使用して同じターゲット上の複数のポータルに接続すると、より広い帯域幅を実現できます。

MS/T 機能は、イニシエータのセッション ID (ISID) を変えることで、ターゲット上に 2 つ以上のセッションを作成します。この機能を有効にすると、ネットワークに 2 つの SCSI レイヤーパスが作成され、複数のターゲットを iSCSI レイヤーから Oracle Solaris I/O レイヤーまで公開できるようになります。MPxIO ドライバは、これらのパスに対する予約を処理します。

iSCSI と MPxIO パスの相互動作の仕組みについては、74 ページの「Oracle Solaris での iSCSI マルチパスデバイスの設定」を参照してください。

iSCSI ターゲットの複数セッションを構成する前に、次の項目を確認してください。

- 通常の MS/T 構成には、2 つ以上の構成済みセッションがあります。  
ただし、ストレージが複数の TPGT をサポートしている場合には、構成済みセッションの数を 1 に設定できます。SendTarget 発見方式を使用している場合には、複数のパスが存在することが自動的に検出されて複数のターゲットセッションが作成されます。
- /etc/driver/drv/iscsi.conf ファイルで mxpio 構成パラメータが有効になっていることを確認します。

```
# cd /etc/driver/drv
# grep mxpio iscsi.conf iscsi.conf
iscsi.conf:mxpio-disable="no";
```

- IPMP を使用して複数のネットワーク接続が構成されていることを確認します。
- 複数のネットワーク接続が使用可能であることを確認します。

```
# ipadm show-addr
```

1 管理者になります。

2 iSCSI イニシエータとターゲットの現在のパラメータ値を一覧表示します。

a. iSCSI イニシエータの現在のパラメータ値を一覧表示します。

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zZR1200
.
.
.
Configured Sessions: 1
```

b. iSCSI ターゲットデバイスの現在のパラメータ値を一覧表示します。

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
Alias: -
.
.
.
Configured Sessions: 1
```

Configured Sessions (構成済みのセッション) の値は、ターゲットポータルグループ内の各ターゲット名用に作成される構成済みの iSCSI セッションの数です。

3 次のいずれかを選択し、構成済みのセッションの数をイニシエータノードで変更してすべてのターゲットに適用するか、ターゲットレベルで変更して特定のターゲットに適用します。

ターゲットのセッション数は 1 から 4 の間である必要があります。

- iSCSI イニシエータノードに必要なパラメータ値を適用します。

```
initiator# iscsiadm modify initiator-node -c 2
```

- iSCSI ターゲットに必要なパラメータ値を適用します。

```
initiator# iscsiadm modify target-param -c 2 iqn.1992-08.com.abcstorage:sn.84186266
```

- 構成済みセッションを1つ以上のローカル IP アドレスにバインドします。

構成済みのセッションは、ローカル IP アドレスにバインドすることもできます。この方法を使用する場合は、コンマ区切りのリストで1つ以上のローカル IP アドレスを指定します。各 IP アドレスは iSCSI セッションを表します。この方法は、initiator-node または target-param のレベルでも使用できます。例:

```
initiator# iscsiadm modify initiator-node -c 10.0.0.1,10.0.0.2
```

---

注-指定した IP アドレスがルーティング可能ではない場合、アドレスが無視され、デフォルトの Oracle Solaris ルートおよび IP アドレスがこのセッションで使用されます。

---

- 4 パラメータが変更されたことを確認します。

- a. イニシエータノードの更新された情報を表示します。

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
.
.
.
Configured Sessions: 2
```

- b. ターゲットノードの更新された情報を表示します。

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
Alias: -
.
.
.
Configured Sessions: 2
```

- 5 複数パスを一覧表示して、OS デバイス名が `iscsiadm list` の出力に一致していることと、パスの数が2以上であることを確認します。

```
initiator# mpathadm list lu
```

# iSCSI 構成の監視

`iscsiadm list` コマンドを使用して、iSCSI イニシエータとターゲットデバイスに関する情報を表示できます。

## ▼ iSCSI 構成情報を表示する方法

- 1 管理者になります。
- 2 iSCSI イニシエータに関する情報を表示します。

例:

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
    Login Parameters (Default/Configured):
        Header Digest: NONE/-
        Data Digest: NONE/-
    Authentication Type: NONE
    RADIUS Server: NONE
    RADIUS access: unknown
    Configured Sessions: 1
```

- 3 使用中の発見方式に関する情報を表示します。

例:

```
# iscsiadm list discovery
Discovery:
    Static: enabled
    Send Targets: enabled
    iSNS: enabled
```

### 例 6-1 iSCSI ターゲットの情報の表示

次の例では、特定の iSCSI ターゲットのパラメータ値を表示する方法を示します。

```
# iscsiadm list target-param iqn.1992-08.com.abcstorage:sn.33592219
Target: iqn.1992-08.com.abcstorage:sn.33592219
```

`iscsiadm list target-param -v` コマンドによって次の情報が表示されます。

- ターゲットの認証値
- ターゲットのログインパラメータのデフォルト値
- 各ログインパラメータに構成されている値

`iscsiadm list target-param -v` コマンドでは、/ 指定子の前にデフォルトのパラメータ値が表示され、/ 指定子の後に構成後のパラメータ値が表示されます。パラメータを構成していない場合は、値がハイフン (-) で表示されます。詳細は、次の例を参照してください。

```
# iscsiadm list target-param -v eui.50060e8004275511 Target: eui.50060e8004275511
Alias: -
Bi-directional Authentication: disabled
Authentication Type: NONE
Login Parameters (Default/Configured):
  Data Sequence In Order: yes/-
  Data PDU In Order: yes/-
  Default Time To Retain: 20/-
  Default Time To Wait: 2/-
  Error Recovery Level: 0/-
  First Burst Length: 65536/-
  Immediate Data: yes/-
  Initial Ready To Transfer (R2T): yes/-
  Max Burst Length: 262144/-
  Max Outstanding R2T: 1/-
  Max Receive Data Segment Length: 65536/-
  Max Connections: 1/-
  Header Digest: NONE/-
  Data Digest: NONE/-
Configured Sessions: 1
```

次の例では、ターゲットとイニシエータの間でネゴシエートされたあとのパラメータが出力されています。

```
# iscsiadm list target -v eui.50060e8004275511
Target: eui.50060e8004275511
TPGT: 1
ISID: 4000002a0000
Connections: 1
  CID: 0
    IP address (Local): 172.90.101.71:32813
    IP address (Peer): 172.90.101.40:3260
    Discovery Method: Static
    Login Parameters (Negotiated):
      Data Sequence In Order: yes
      Data PDU In Order: yes
      Default Time To Retain: 0
      Default Time To Wait: 3
      Error Recovery Level: 0
      First Burst Length: 65536
      Immediate Data: yes
      Initial Ready To Transfer (R2T): yes
      Max Burst Length: 262144
      Max Outstanding R2T: 1
      Max Receive Data Segment Length: 65536
      Max Connections: 1
      Header Digest: NONE
      Data Digest: NONE
```

# iSCSI イニシエータと iSCSI ターゲットのパラメータの変更

iSCSI イニシエータおよび iSCSI ターゲットデバイスのどちらでもパラメータは変更可能です。ただし、iSCSI イニシエータ上で変更可能なパラメータは、次のものだけです。

- iSCSI イニシエータのノード名 - イニシエータのノード名を別の名前に変更できます。イニシエータのノード名を変更した場合は、名前を変更したときの iSNS サーバー上の発見ドメインインストールの構成に応じて、iSNS によって発見されたターゲットがイニシエータのターゲットリストから削除されることがあります。詳細は、83 ページの「[iSCSI イニシエータと iSCSI ターゲットのパラメータを変更する方法](#)」を参照してください。
- ヘッダーダイジェスト - NONE、デフォルト値、または CRC32。
- データダイジェスト - NONE、デフォルト値、または CRC32。
- 認証と CHAP 秘密鍵 - 認証の設定方法の詳細は、69 ページの「[iSCSI イニシエータの CHAP 認証を構成する方法](#)」を参照してください。
- 構成済みセッション - 複数セッションの構成方法については、75 ページの「[ターゲットの複数の iSCSI セッションを有効にする方法](#)」を参照してください。

iSCSI ドライバは、iSCSI イニシエータと iSCSI ターゲットデバイスのパラメータのデフォルト値を提供します。iSCSI イニシエータのパラメータを変更すると、iSCSI ターゲットデバイスにすでに別の値が設定されている場合以外、その変更後のパラメータが iSCSI ターゲットデバイスに継承されます。



注意 - 変更対象のパラメータがターゲットソフトウェアによってサポートされていることを確認してください。サポートされていない場合、iSCSI ターゲットデバイスにログインできない可能性があります。サポートされているパラメータの一覧については、使用するアレイのドキュメントを参照してください。

iSCSI パラメータを変更できるのは、イニシエータとターゲット間の I/O が完了したあとだけです。iscsiadm modify コマンドを使って変更が実行されると、iSCSI ドライバはセッションを接続し直します。

## iSCSI パラメータのチューニング

iSCSI パラメータをチューニングして、iSCSI イニシエータのさまざまな応答や接続時間値を調整できます。イニシエータシステムが接続されているすべての

ターゲットでイニシエータのパラメータを調整するかどうか、または特定のターゲットのパラメータを調整するかどうかに応じて、iSCSI パラメータをチューニングできます。

特定の iSCSI ターゲットのパラメータ値を変更する場合は、次のコマンドを使用します。

```
iscsiadm modify target-param -T tunable-prop=value target-name
```

すべてのターゲットのパラメータ値を調整する場合は、次のコマンドを使用します。

```
iscsiadm modify initiator-node -T tunable-prop=value
```

次のチューニング可能パラメータは、アクティブな接続だけに適用され、iSCSI イニシエータおよびイニシエータに接続するターゲットの動作を変更します。パラメータを動的にチューニングする機能により、iSCSI イニシエータを柔軟に構成できます。

表 6-1 チューニング可能な iSCSI パラメータ

パラメータ名	説明	有効な値 (秒)	デフォルト値 (秒)
recv-login-rsp-timeout	セッションログインの応答時間 – iSCSI イニシエータが、指定された iSCSI ターゲットからの iSCSI セッションログイン要求の応答を待機する時間を指定します。	0 – 3600	60
conn-login-max	接続再試行の最大時間 – iSCSI イニシエータからターゲットへの入出力がタイムアウトするか、接続が失敗したあとで、iSCSI イニシエータがターゲットへの接続を試みる最大秒数を決定します。	0 – 3600	180
polling-login-delay	ログイン再試行間隔 – iSCSI イニシエータからターゲットへの入出力がタイムアウトするか、接続が失敗したあとの、各 iSCSI セッションのログイン試行の間隔を決定します。	0 – 3600	60

## ▼ iSCSI パラメータをチューニングする方法

- 1 チューニング可能な iSCSI パラメータをすべて表示します。  
すべてのターゲットの iSCSI パラメータ情報を表示します。

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:e00000000000.4e36d278
Initiator node alias: unknown
  Login Parameters (Default/Configured):
    Header Digest: NONE/-
    Data Digest: NONE/-
    Max Connections: 65535/-
  Authentication Type: NONE
  RADIUS Server: NONE
  RADIUS Access: disabled
  Tunable Parameters (Default/Configured):
    Session Login Response Time: 60/-
    Maximum Connection Retry Time: 180/-
    Login Retry Time Interval: 60/-
  Configured Sessions: 1
```

特定のターゲットの iSCSI パラメータ情報を表示します。

```
# iscsiadm list target-param [target-name]
```

例:

```
# iscsiadm list target-param -v iqn.1986-03.com.sun:02:47ac0
506-cd48-67f5-fc0d-ab7544d37538
Target: iqn.1986-03.com.sun:02:47ac0506-cd48-67f5-fc0d-ab7544d37538
Alias: -
Bi-directional Authentication: disabled
Authentication Type: NONE
  Login Parameters (Default/Configured):
    Data Sequence In Order: yes/-
    Data PDU In Order: yes/-
    Default Time To Retain: 20/-
    Default Time To Wait: 2/-
    Error Recovery Level: 0/-
    First Burst Length: 65536/-
    Immediate Data: yes/-
    Initial Ready To Transfer (R2T): yes/-
    Max Burst Length: 262144/-
    Max Outstanding R2T: 1/-
    Max Receive Data Segment Length: 8192/-
    Max Connections: 65535/-
    Header Digest: NONE/-
    Data Digest: NONE/-
  Tunable Parameters (Default/Configured):
    Session Login Response Time: 60/-
    Maximum Connection Retry Time: 180/-
    Login Retry Time Interval: 60/-
  Configured Sessions: 1
```

## 2 iSCSI パラメータをチューニングします。

たとえば、あるターゲットの接続再試行の最大時間を 90 秒に設定するには:

```
# iscsiadm modify target-param -T conn-login-max=90 iqn.1986-03.com.sun:02:47ac0506-cd48-67f5-fc0d-ab7544d37538
```

たとえば、すべてのターゲットの接続再試行の最大時間を 90 秒に設定するには:

```
# iscsiadm modify initiator-node -T conn-login-max=90
```

## ▼ iSCSI イニシエータと iSCSI ターゲットのパラメータを変更する方法

この手順の前半では、iSCSI イニシエータの変更されたパラメータが iSCSI ターゲットデバイスにどのように継承されるかを示します。この手順の後半では、iSCSI ターゲットデバイス上のパラメータを実際に変更する方法を示します。

この省略可能な手順では、ユーザーが現在ログインしているローカルシステム上で、iSCSI ターゲットデバイスへのアクセスがすでに構成されているものとします。

- 1 管理者になります。
- 2 iSCSI イニシエータとターゲットデバイスの現在のパラメータ値を一覧表示します。
  - a. iSCSI イニシエータの現在のパラメータ値を一覧表示します。

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zxr1200
Login Parameters (Default/Configured):
  Header Digest: NONE/-
  Data Digest: NONE/-
Authentication Type: NONE
RADIUS Server: NONE
RADIUS access: unknown
Configured Sessions: 1
```

- b. iSCSI ターゲットデバイスの現在のパラメータ値を一覧表示します。

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
Alias: -
Bi-directional Authentication: disabled
Authentication Type: NONE
Login Parameters (Default/Configured):
  Data Sequence In Order: yes/-
  Data PDU In Order: yes/-
  Default Time To Retain: 20/-
  Default Time To Wait: 2/-
  Error Recovery Level: 0/-
  First Burst Length: 65536/-
  Immediate Data: yes/-
```

```

Initial Ready To Transfer (R2T): yes/-
Max Burst Length: 262144/-
Max Outstanding R2T: 1/-
Max Receive Data Segment Length: 65536/-
Max Connections: 1/-
Header Digest: NONE/-
Data Digest: NONE/-
Configured Sessions: 1

```

iSCSI イニシエータと iSCSI ターゲットデバイスの両方で、ヘッダーダイジェストとデータダイジェストのパラメータがどちらも現在 NONE に設定されています。

iSCSI ターゲットデバイスのデフォルトパラメータ値を確認するには、例 6-1 の `iscsiadm list target-param` 出力を参照してください。

### 3 iSCSI イニシエータのパラメータを変更します。

たとえば、ヘッダーダイジェストを CRC32 に設定します。

```
initiator# iscsiadm modify initiator-node -h CRC32
```

イニシエータのノード名を変更した場合は、新しい名前がターゲットと同じ発見ドメインに属していないと、iSNS によって発見されたターゲットがログアウトおよび削除されることがあります。ただし、ターゲットが使用中の場合は、削除されません。たとえば、これらのターゲット上でファイルが開いている場合またはファイルシステムがマウントされている場合は、それらのターゲットは削除されません。

これらのターゲットと新しいイニシエータノードの名前が同じ発見ドメインに属している場合は、名前の変更後に新しいターゲットも表示されることがあります。

### 4 パラメータが変更されたことを確認します。

#### a. iSCSI イニシエータの更新済みパラメータ情報を表示します。

```

initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
Login Parameters (Default/Configured):
  Header Digest: NONE/CRC32
  Data Digest: NONE/-
Authentication Type: NONE
RADIUS Server: NONE
RADIUS access: unknown
Configured Sessions: 1

```

ヘッダーダイジェストは CRC32 に設定されています。

#### b. iSCSI ターゲットデバイスの更新済みパラメータ情報を表示します。例:

```

initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
Alias: -
Bi-directional Authentication: disabled
Authentication Type: NONE
Login Parameters (Default/Configured):
  Data Sequence In Order: yes/-

```

```

Data PDU In Order: yes/-
Default Time To Retain: 20/-
Default Time To Wait: 2/-
Error Recovery Level: 0/-
First Burst Length: 65536/-
Immediate Data: yes/-
Initial Ready To Transfer (R2T): yes/-
Max Burst Length: 262144/-
Max Outstanding R2T: 1/-
Max Receive Data Segment Length: 65536/-
Max Connections: 1/-
Header Digest: CRC32/-
Data Digest: NONE/-

```

Configured Sessions: 1

ヘッダーダイジェストはCRC32に設定されています。

## 5 iSCSI イニシエータが iSCSI ターゲットに接続し直されたことを確認します。

```

initiator# iscsiadm list target -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
  TPGT: 2
  ISID: 4000002a0000
  Connections: 1
    CID: 0
      IP address (Local): nnn.nn.nn.nnn:64369
      IP address (Peer): nnn.nn.nn.nnn:3260
      Discovery Method: SendTargets
      Login Parameters (Negotiated):
        .
        .
        .
      Header Digest: CRC32
      Data Digest: NONE

```

## 6 (オプション)iSCSI イニシエータまたは iSCSI ターゲットデバイスのパラメータをリセットします。

`iscsiadm modify` コマンドを使用して、パラメータをそのデフォルト値にリセットできます。または、`iscsiadm remove` コマンドを使って、すべてのパラメータをそれらのデフォルト値にリセットできます。

`iscsiadm modify target-param` コマンドは、コマンド行で指定されたパラメータのみを変更します。

次の例では、ヘッダーダイジェストを `NONE` にリセットする方法を示します。

```
initiator# iscsiadm modify target-param -h none iqn.1992-08.com.abcstorage:sn...
```

`iscsiadm remove target-param` コマンドについては、[iscsiadm\(1M\)](#) のマニュアルページを参照してください。

## iSCSI 構成に関する問題のトラブルシューティング

一般的な iSCSI 構成に関する問題のトラブルシューティングを行う際に利用可能なツールを、次に示します。

- snoop – このツールは更新され、iSCSI パケットをサポートするようになりました。
- Wireshark – この製品は <http://www.wireshark.org/> から入手できます。

どちらのツールも、ポート 3260 上の iSCSI パケットをフィルタできます。

次のセクションでは、iSCSI のさまざまな問題およびエラーメッセージのトラブルシューティングを行う方法を説明します。

### ローカルシステムから iSCSI ターゲットに接続できない

#### ▼ iSCSI の接続に関する問題のトラブルシューティングを行う方法

- 1 管理者になります。
- 2 iSCSI ターゲットの情報を一覧表示します。

例:

```
initiator# iscsiadm list target
Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01
TPGT: default
ISID: 4000002a0000
Connections: 0
```

- 3 `iscsiadm list target` の出力結果に接続が表示されていない場合は、接続が失敗したことについて、`/var/adm/messages` ファイル内で考えられる理由を調査します。

また、`ping` コマンドを使用するか、または、`telnet` コマンドを使ってストレージデバイスの iSCSI ポートに接続し、iSCSI サービスが利用可能かどうかを確認することによって、接続がアクセス可能かどうかを確認することもできます。デフォルトのポートは 3260 です。

さらに、ストレージデバイスのログファイルでエラーを確認します。

- 4 **iscsiadm list target** の出力結果に目的のターゲットが表示されていない場合は、`/var/adm/messages` ファイル内でその考えられる原因を調査します。

SendTargets を発見方式として使用している場合は、`-v` オプションを使って `discovery-address` の一覧表示を試みると、ホストが目的のターゲットを認識しているかどうかわかります。例:

```
initiator# iscsiadm list discovery-address -v 10.0.0.1
Discovery Address: 10.0.0.1:3260
  Target name: eui.210000203787dfc0
    Target address:      10.0.0.1:11824
  Target name: eui.210000203787e07b
    Target address:      10.0.0.1:11824
```

発見方式として iSNS を使用している場合は、iSNS 発見方式を有効にし、`v` オプションを使用して `-isns-server` を一覧表示することで、期待されるターゲットがホストに認識されることを確認します。例:

```
initiator# iscsiadm list isns-server -v
iSNS Server IP Address: 10.20.56.56:3205
  Target name: iqn.1992-08.com.xyz:sn.1234566
    Target address:  10.20.57.161:3260, 1
  Target name: iqn.2003-10.com.abc:group-0:154:abc-65-01
    Target address:  10.20.56.206:3260, 1
  Target name: iqn.2003-10.com.abc:group-0:154:abc-65-02
    Target address:  10.20.56.206:3260, 1
.
.
.
```

## ローカルシステム上で iSCSI デバイスまたは iSCSI ディスクが利用できない

### ▼ iSCSI デバイスまたは iSCSI ディスクが利用できない問題のトラブルシューティングを行う方法

- 1 管理者になります。
- 2 列挙中にこのターゲット上で発見された LUN を確認します。  
`-s` オプションを指定すると、列挙中にこのターゲット上で発見された LUN が表示されます。

例:

```
# iscsiadm list target -S
Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01
  TPGT: default
  ISID: 4000002a0000
  Connections: 1
  LUN: 0
```

```
Vendor: ABCSTOR
Product: 0010
OS Device Name: /dev/rdisk/c3t34d0s2
```

- 3 `/var/adm/messages` ファイルを確認して、エラーが報告されたかどうかを調べます。LUN が一覧表示されるべきなのにそうになっていない場合は、このログファイルを確認してください。
- 4 ストレージデバイスのログファイルにエラーが記録されていないか確認します。
- 5 ストレージデバイスのすべての LUN マスクが正しく構成されていることを確認します。

## iSNS 発見方式を使用する場合の LUN マスクの使用

特定のイニシエータに対するストレージの認証を制御する手段として iSNS 発見ドメインを使用しないでください。代わりに LUN マスクを使用して、承認されたイニシエータだけが LUN にアクセスできることを確認します。

ターゲットが使用中のときに発見ドメインからターゲットを削除しようとした場合、iSCSI イニシエータはこのターゲットからログアウトしません。このイニシエータがこのターゲット(および関連付けられた LUN)にアクセスしないようにする場合は、LUN マスクを使用する必要があります。ターゲットを発見ドメインから削除するだけでは不十分です。

## iSCSI の一般的なエラーメッセージ

このセクションでは、`/var/adm/messages` ファイルに見つかる可能性のある iSCSI メッセージと、回復のために適用できる解決方法について説明します。

メッセージの形式は次のとおりです。

```
iscsi TYPE (OID) STRING (STATUS-CLASS#/STATUS-DETAIL#)
```

**TYPE** 接続 (connection)、セッション (session) のいずれかです。

**OID** 接続またはセッションのオブジェクト ID です。この ID は、OS インスタンスに一意です。

**STRING** 状態の説明です。

**STATUS-CLASS#/STATUS-DETAIL#** これらの値は、RFC 3720 で定義されている iSCSI ログイン応答として返されます。

iscsi connection(OID) login failed - Miscellaneous iSCSI initiator errors.  
原因:何らかのイニシエータエラーにより、デバイスログインが失敗しました。

iscsi connection(OID) login failed - Initiator could not be successfully authenticated.  
原因:デバイスによるイニシエータの認証が成功しませんでした。

対処方法:該当する場合には、CHAP 名、CHAP パスワード、または RADIUS サーバーの設定が正しいことを確認します。

iscsi connection(OID) login failed - Initiator is not allowed access to the given target.

原因:イニシエータが iSCSI ターゲットデバイスにアクセスすることを、デバイスが許可できません。

対処方法:イニシエータ名を確認するとともに、その名前がストレージデバイスによって正しくマスクまたはプロビジョニングされていることを確認します。

iscsi connection(OID) login failed - Requested ITN does not exist at this address.

原因:ユーザーが要求する iSCSI ターゲット名 (ITN) へのアクセスを、デバイスが提供しません。

対処方法:イニシエータの発見情報が正しく指定されており、ストレージデバイスが正しく構成されていることを確認します。

iscsi connection(OID) login failed - Requested ITN has been removed and no forwarding address is provided.

原因:ユーザーが要求する iSCSI ターゲット名 (ITN) へのアクセスを、デバイスが提供しなくなりました。

対処方法:イニシエータの発見情報が正しく入力されており、ストレージデバイスが正しく構成されていることを確認します。

iscsi connection(OID) login failed - Requested iSCSI version range is not supported by the target.

原因:イニシエータの iSCSI バージョンがストレージデバイスによってサポートされていません。

iscsi connection(OID) login failed - No more connections can be accepted on this Session ID (SSID).

原因:ストレージデバイスは、このイニシエータノードから iSCSI ターゲットデバイスへの接続をこれ以上受け入れることができません。

iscsi connection(OID) login failed - Missing parameters (e.g., iSCSI initiator and/or target name).

原因:ストレージデバイスが、イニシエータ名またはターゲット名が正しく指定されていないと報告しています。

対処方法:iSCSI のイニシエータ名またはターゲット名を正しく指定します。

iscsi connection(OID) login failed - Target hardware or software error.

原因:ストレージデバイスでハードウェアエラーまたはソフトウェアエラーが発生しました。

対処方法:ストレージのドキュメントを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

iscsi connection(OID) login failed - iSCSI service or target is not currently operational.

原因:ストレージデバイスが現在動作していません。

対処方法:ストレージのドキュメントを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

iscsi connection(OID) login failed - Target has insufficient session, connection or other resources.

原因:ストレージデバイスのリソースが不足しています。

対処方法:ストレージのドキュメントを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

iscsi connection(OID) login failed - unable to initialize authentication

iscsi connection(OID) login failed - unable to set authentication

iscsi connection(OID) login failed - unable to set username

iscsi connection(OID) login failed - unable to set password

iscsi connection(OID) login failed - unable to set ipsec

iscsi connection(OID) login failed - unable to set remote authentication

原因:イニシエータが認証の初期化または設定を正しく行えませんでした。

対処方法:イニシエータの認証が正しく構成されていることを確認します。

iscsi connection(OID) login failed - unable to make login pdu

原因:イニシエータが、イニシエータまたはストレージデバイスの設定に基づいてログインのペイロードデータユニット (PDU) を作成できませんでした。

対処方法:任意のターゲットログインパラメータやその他のデフォルト以外の設定値のリセットを試みます。

iscsi connection(OID) login failed - failed to transfer login

iscsi connection(OID) login failed - failed to receive login response

原因: イニシエータが、ログインのペイロードデータユニット (PDU) をネットワーク接続経由で転送または受信できませんでした。

対処方法: ネットワーク接続が到達可能であることを確認します。

iscsi connection(OID) login failed - received invalid login response (OP CODE)

原因: ストレージデバイスがログインに対して予想外の応答を返しました。

iscsi connection(OID) login failed - login failed to authenticate with target

原因: イニシエータがストレージデバイスを認証できませんでした。

対処方法: イニシエータの認証が正しく構成されていることを確認します。

iscsi connection(OID) login failed - initiator name is required

原因: どのアクションを実行する場合も、イニシエータ名が構成されている必要があります。

対処方法: イニシエータ名が構成されていることを確認します。

iscsi connection(OID) login failed - authentication receive failed

iscsi connection(OID) login failed - authentication transmit failed

原因: イニシエータが認証情報を転送または受信できませんでした。

対処方法: 状況に応じて、ストレージデバイスまたはRADIUSサーバーとのネットワーク接続を確認します。

iscsi connection(OID) login failed - login redirection invalid

原因: ストレージデバイスが、イニシエータを無効な宛先にリダイレクトしようとしました。

対処方法: ストレージのドキュメントを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

iscsi connection(OID) login failed - target protocol group tag mismatch, expected <TPGT>, received <TPGT>

原因: イニシエータとターゲットのTPGT (ターゲットポータルグループタグ、Target Portal Group Tag) が一致しません。

対処方法: イニシエータ上またはストレージデバイス上のTPGT発見設定を確認します。

iscsi connection(OID) login failed - can't accept *PARAMETER* in security stage  
原因:ログインのセキュリティーフェーズで、デバイスがサポートされていないロ  
グインパラメータで応答しました。

対処方法:パラメータ名が参考のため記載されています。ストレージのドキュメン  
トを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

iscsi connection(OID) login failed - HeaderDigest=CRC32 is required, can't  
accept *VALUE*

iscsi connection(OID) login failed - DataDigest=CRC32 is required, can't accept  
*VALUE*

原因:このターゲットに対して、CRC32に設定されたHeaderDigestまたは  
DataDigestのみを受け入れるようにイニシエータが構成されています。デバイ  
スは値 *VALUE* を返しました。

対処方法:イニシエータとデバイスのダイジェスト設定に互換性があることを確認  
します。

iscsi connection(OID) login failed - HeaderDigest=None is required, can't  
accept *VALUE*

iscsi connection(OID) login failed - DataDigest=None is required, can't accept  
*VALUE*

原因:このターゲットに対して、NONEに設定されたHeaderDigestまたは  
DataDigestのみを受け入れるようにイニシエータが構成されています。デバイ  
スは値 *VALUE* を返しました。

対処方法:イニシエータとデバイスのダイジェスト設定に互換性があることを確認  
します。

iscsi connection(OID) login failed - can't accept *PARAMETER*

原因:イニシエータはこのパラメータをサポートしません。

iscsi connection(OID) login failed - can't accept MaxOutstandingR2T *VALUE*

原因:イニシエータは、記載された *VALUE* のMaxOutstandingR2Tを受け入れませ  
ん。

iscsi connection(OID) login failed - can't accept MaxConnections *VALUE*

原因:イニシエータは、記載された *VALUE* の最大接続数を受け入れません。

iscsi connection(OID) login failed - can't accept ErrorRecoveryLevel *VALUE*

原因:イニシエータは、記載された *VALUE* のエラー回復レベルを受け入れませ  
ん。

**iscsi session(OID) NAME offline**

原因: このターゲット *NAME* のすべての接続が、削除されたか、または失敗しました。

**iscsi connection(OID) failure - unable to schedule enumeration**

原因: イニシエータがこのターゲットの LUN を列挙できませんでした。

対処方法: LUN の列挙を強制実行するには、`devfsadm -i iscsi` コマンドを実行します。詳細は、[devfsadm\(1M\)](#) のマニュアルページを参照してください。

**iscsi connection(OID) unable to connect to target NAME (errno: ERRNO )**

原因: イニシエータによるネットワーク接続の確立が失敗しました。

対処方法: 接続エラーに関する特定の *ERRNO* については、`/usr/include/sys/errno.h` ファイルを参照してください。



## 仮想ファイバチャネルポートの構成

---

この章では、仮想ファイバチャネルポートとしても知られている、Nポート ID 仮想化 (NPIV) ポートを構成するために使用するステップについて説明します。

次の項目について説明します。

- 95 ページの「NPIV の概要」
- 96 ページの「NPIV の制限事項」
- 96 ページの「NPIV ポートの操作」

### NPIV の概要

NPIV は、1つのファイバチャネルアダプタが多数の Nポート ID を持つことを可能にするファイバチャネル機能です。各 Nポートは SAN 上で固有の識別子 (ポート WWN およびノード WWN) を持ち、ゾーン作成および LUN マスクに使用できます。ソフトゾーン作成は、ポート WWN によってポートをグループ化するために使用でき、ゾーン作成の優先される方法です。

## NPIVの制限事項

ファイバチャネルポートを仮想化するために使用する際のNPIVの制限事項は、次のとおりです。

- NPIVポートはブート用に使用できません。
- NPIVポートがSANでもっとも適切に使用されるのは、仮想ポートまたは物理ポートのいずれかのポート数が比較的少ない場合です。また、一部のターゲットは、NPIVが作成できる大量のポートを処理するための十分なリソースを持たない場合もあります。この制限事項が存在するのは、SANに大量のポートがある場合、SAN上で状態変更通知(SCN)を処理するのに著しく時間がかかるためです。可視のポートの数を制限できるゾーン機能を使用することによって、大規模なSANでこの制限事項を回避できます。
- MPxIOをNPIVと一緒に使用できますが、異なるパスが物理的に冗長であるようにする必要があります。
- NPIVはファブリックトポロジでのみサポートされ、FC-ALやポイントツーポイントトポロジではサポートされません。
- すべてのハードウェアがNPIVをサポートすることはかぎりません。スイッチおよびHBA(ターゲットではない)の両方が、SAN内でNPIVをサポートする必要があります。仕様では、HBAは最大255個の仮想ポートをサポートする必要がありますが、この能力はスイッチのリソースによって定義されます。NPIVサポートのために、スイッチを最新のファームウェアレベルに更新することが必要な場合もあります。

## NPIVポートの操作

仮想化されていない環境では、`fcadm` コマンドを使用してNPIVを構成できます。

ポートが`fcadm`から作成されたかどうかに関係なく、NPIVポートのステータスを判定するために、`fcinfo`および`fcadm`ステータスコマンドが使用できます。このコマンドは、物理ポートとそのポートにホストされている仮想ポートの間の関係を報告します。

`luxadm`や`cfgadm`などのほかのファイバチャネルコマンドはNPIV情報を報告しますが、仮想ポートと物理ポートの区別は行われません。

### ▼ NPIVポートの作成方法

始める前に 各仮想ポートはポート名とノード名を持つ必要があります。ポート名はSAN上で一意となる必要があります。名前を手動で割り当てるか、組み込みのランダムWWNジェネレータを使用できます。重複した名前を登録しようとする、ほとんどのスイッチは新しく登録されたWWNについてエラーステータスを報告し、スイッチは新しいWWNを登録しません。

受け入れ可能な名前前のフォーマットの詳細については、T11 標準: Fibre Channel Framing and Signaling (FC-FS 2) を参照してください。

NPIV をサポートしない HBA 上に NPIV ポートを作成しようとした場合、エラーが発生します。NPIV をサポートする HBA 上に NPIV ポートを作成しようとしたが、NPIV をサポートしないスイッチに接続されている場合、ポートはオフラインステータスで作成されます。ステータスは `fcinfo(1M)` 出力に報告されます。

- 1 管理者になります。
- 2 NPIV ポートを作成します。

```
# fcadm create-npiv-port -p Virtual_Port_WWN -n Virtual_Node_WWN
PhysicalPort_port_WWN
```

-p および -n オプションがない場合、ランダム WWN が仮想ポートおよび仮想ノードにそれぞれ割り当てられます。

#### 例 7-1 NPIV ポートを作成する

次の例は、WWN が 210000e08b170f1c の物理 HBA ポート上に NPIV ポートを作成し、仮想ポート WWN は 2000000000000001 に設定され、仮想ノード WWN は 2100000000000001 に設定されます。

```
# fcadm create-npiv-port -p 2000000000000001 -n 2100000000000001 210000e08b170f1c
```

## ▼ NPIV ポートの削除方法

始める前に `fcinfo hba-port` コマンドを使用して、NPIV ポートの現在の WWN 値を表示します。

- 1 管理者になります。
- 2 NPIV ポートを削除します。

```
# fcadm delete-npiv-port -p Virtual_Port_WWN -n Virtual_Node_WWN
PhysicalPort_port_WWN
```

#### 例 7-2 NPIV ポートを削除する

次の例は、WWN が 210000e08b170f1c の物理 HBA ポート上の NPIV ポートを削除します。

```
# fcadm delete-npiv-port -p 2000000000000001 -n 2100000000000001 210000e08b170f1c
```

## ▼ NPIV ポートステータスの表示方法

- 1 管理者になります。
- 2 現在構成済みの NPIV ポートを表示します。

```
# fcinfo hba-port
```

NPIV ポート一覧には、現在構成済みの NPIV ポートの一覧が表示されています。

### 例 7-3 NPIV ポートのステータスを表示する

次の例は、HBA ポート 210000e08b84f7eb が 1 つの仮想ポートを持つことを示しています。

```
# fcinfo hba-port
HBA Port WWN: 210000e08b84f7eb
  Port Mode: Initiator
  Port ID: 10100
  OS Device Name: /dev/cfg/c7
  Manufacturer: QLogic Corp.
  Model: 375-3294-01
  Firmware Version: 04.04.00
  FCode/BIOS Version: BIOS: 1.4; fcode: 1.11; EFI: 1.0;
  Serial Number: 0402F00-0549112895
  Driver Name: qlc
  Driver Version: 20080430-0.00
  Type: N-port
  State: online
  Supported Speeds: 1Gb 2Gb 4Gb
  Current Speed: 4Gb
  Node WWN: 200000e08b84f7eb
  Max NPIV Ports: 63
  NPIV port list:
    Virtual Port1:
      Node WWN: 1110000000000000
      Port WWN: 1210000000000000
```

## FCoE ポートの構成

---

この章では、通常の Ethernet インタフェースにホストされている FCoE ポートの構成に使用されるステップについて説明します。この章は、コンバージドネットワークアダプタ (CNA) 上のハードウェア FCoE ポートには適用されません。

次の項目について説明します。

- 99 ページの「FCoE の概要」
- 99 ページの「FCoE の制限事項」
- 100 ページの「FCoE ポートの構成」

### FCoE の概要

FCoE は、カプセル化されたファイバチャネルフレームを拡張された Ethernet 上で転送する新しい T11 標準です。FCoE は、データセンターにおけるネットワークコンバージェンスとコスト効果の高い SAN 拡張を可能にするように設計されています。

### FCoE の制限事項

Solaris FCoE イニシエータは、通常の Ethernet コントローラと一緒に動作するように設計されたソフトウェア実装です。ただし、次に示す Solaris FCoE 実装の制限事項に従う必要があります。

- FCoE ポートはブート用に使用できません。
- FCoE ポートは Oracle VM Server for SPARC または Oracle VM Server 3.0 for x86 ゲストオペレーティングシステムで構成できません。
- FCoE はファブリックトポロジおよびポイントツーポイントトポロジでサポートされています。

FCoE はすべてのハードウェアでサポートされているわけではありません。FCoE は、802.3x PAUSE およびジャンボフレームをサポートし、GLDv3 ドライバを持つ Ethernet コントローラで動作します。

## FCoE ポートの構成

FCoE ポートは `fcadm` コマンドを使用して構成できます。FCoE ポートのステータスを判別するために `fcinfo` および `fcadm` コマンドを使用します。これらのコマンドは、Ethernet インタフェースとそのインタフェースにホストされている FCoE ポートの間の関係も報告します。

`luxadm` や `cfgadm` などのほかのファイバチャネルコマンドは FCoE 情報を報告しますが、FCoE とネイティブ FC ポートの区別は行われません。

### ▼ FCoE ポートの作成方法

始める前に この手順を開始する前に、次のタスクを実行する必要があります。

- Ethernet インタフェース上で 802.3x (PAUSE と呼ばれる) 設定を有効にします。この設定により、損失のない Ethernet トランスポートが保証されます。
- Ethernet インタフェース上のジャンボフレーム (2.5K バイトより大きい) を有効にします。ファイバチャネルデータフレームは、2136 バイトになることがあります。

これらの設定は、Ethernet ハードウェアおよびドライバにより異なる場合があります。ほとんどの場合、Ethernet インタフェースの `driver.conf` ファイルを変更してから、リポートする必要があります。これらの機能を有効にする方法の詳細は、Ethernet インタフェースの `driver.conf` ファイルを参照してください。

各仮想ポートはポート名とノード名を持つ必要があります。ポート名は SAN 上で一意となる必要があります。名前を手動で割り当てるか、組み込みの WWN ジェネレータを使用できます。重複した名前を登録しようとする、スイッチは新しく登録された WWN についてエラーステータスを報告し、スイッチは新しい WWN を登録しません。受け入れ可能な名前のフォーマットの詳細については、T11 標準: Fibre Channel Framing and Signaling (FC-FS 2) を参照してください。

FCoE をサポートしないネットワークインタフェース上に FCoE ポートを作成しようとした場合、エラーが発生して FCoE ポートは作成されません。

- 次のサービスを有効にします。

```
# svcadm enable svc:/system/fcoe_target:default
# svcadm enable svc:/system/stmf:default
```

- 1 管理者になります。

## 2 FCoE ポートを作成します。

```
# fcadm create-fcoe-port -i -p Port_WWN -n Node_WWN Ethernet_Interface
```

選択された Ethernet インタフェースが複数ユニキャストアドレスをサポートしない場合、そのインタフェース上でプロミスキュー (promiscuous) モードを明示的に有効にすることを求めるメッセージが表示されます。

```
# fcadm create-fcoe-port -i -f Ethernet_Interface
```

例:

```
# fcadm create-fcoe-port -i nxge0
```

## ▼ FCoE ポートの削除方法

始める前に `fcadm list-fcoe-ports` コマンドを使用して、FCoE ポートをホストする Ethernet インタフェースを表示できます。

- 1 管理者になります。
- 2 FCoE ポートを削除します。

```
# fcadm delete-fcoe-port network_interface
```

例:

```
# fcadm delete-fcoe-port nxge0
```

## ▼ FCoE ポートステータスの表示方法

- 1 管理者になります。
- 2 現在構成済みの FCoE ポートのステータスを表示します。

```
# fcinfo hba-port -e
```

例:

```
# fcinfo hba-port -e
HBA Port WWN: 200000144fc1f5c8
  Port Mode: Initiator
  Port ID: 9a0042
  OS Device Name: /dev/cfg/c6
  Manufacturer: Sun Microsystems, Inc.
  Model: FCoE Virtual FC HBA
  Firmware Version: N/A
  FCode/BIOS Version: N/A
  Serial Number: N/A
  Driver Name: SunFC FCoEI v20090422-1.00
```

```
Driver Version: v20090422-1.00
Type: N-port
State: online
Supported Speeds: 1Gb 10Gb
Current Speed: 10 Gb
Node WWN: 100000144fc1f5c8
```

システム内のすべての FCoE ポートについての FC 固有情報を一覧表示します。

```
# fcadm list-fcoe-ports
```

例:

```
# fcadm list-fcoe-ports
HBA Port WWN: 200000144fc1f5c8
  Port Type: Initiator
  MAC Name: nxge0
  MTU Size: 9194
  Primary MAC Address: 00144fc1f5c8
  Current MAC Address: 0efc009a0042
  Promiscuous Mode: Off
```

## ▼ FCoE ポートの再初期化を強制実行する方法

FCoE ポートを再初期化する必要がある場合は、次のステップを使用します。FC SAN に新規デバイスが追加された場合や、正しく動作しないデバイスが SAN に存在することが原因で、FCoE ポートの再初期化を強制することが必要な場合があります。多くの場合、この操作は FC-SAN の問題を解決できます。

このコマンドをターゲットポート側で発行すると、ターゲットポートがリセットされます。このコマンドをホストポート側から発行すると、ホストポートがリセットされます。

FC スイッチが接続されている場合、SAN のほかの FC ポートはリモート状態変更通知 (RSCN) を受け取ります。また、この操作の後、ほかのイニシエータは常にポートを再発見し、FC ログインセッションが確立されるか再使用されます。このコマンドは I/O を中断させるものですが、I/O は継続します。このコマンドはデータ損失を発生させないため非破壊的です。

- 1 管理者になります。
- 2 ポートに接続されているリンクの再初期化を強制的に実行します。

例:

```
# fcadm force-lip 200000144fc2d508
```

## FCoEハードウェアオフロードの構成

表7-1に一覧表示されているチューニング可能な設定を使用して、FCoEポートを持つシステムのCPU使用率を削減し、パフォーマンスを向上させることができます。これらのチューニング可能な設定はIntel 10 Gb Ethernet デバイスによってサポートされ、`ixgbe.conf` ファイルに設定されます。

`/kernel/drv/ixgbe.conf` を `/etc/driver/drv/ixgbe.conf` にコピーし、FCoE環境のためのチューニング可能な値を変更します。

表8-1 FCoEハードウェアオフロードのチューニング可能パラメータ

チューニング可能パラメータ	説明	範囲	デフォルト値
<code>fcoe_txcrc_enable</code>	ixgbe ドライバが、送信する FCoE パケットの FC CRC トランザクションをオフロードするかどうかを制御します。	0 - FC CRC トランザクションのオフロードを無効にします 1 - FC CRC トランザクションのオフロードを有効にします	1
<code>fcoe_lso_enable</code>	ixgbe ドライバが、送信する FCoE パケットの FC 大量送信 トランザクションをオフロードするかどうかを制御します。	0 - FC LSO トランザクションのオフロードを無効にします 1 - FC LSO トランザクションのオフロードを有効にします	1
<code>fcoe_rxcrc_enable</code>	ixgbe ドライバが、受信する FCoE パケットの RC CRC トランザクションをオフロードするかどうかを制御します。	0 - FC RX CRC トランザクションのオフロードを無効にします 1 - FC RX CRC トランザクションのオフロードを有効にします	1
<code>fcoe_lro_enable</code>	ixgbe ドライバが、受信する FCoE パケットの FC 大量受信 トランザクションをオフロードするかどうかを制御します。	0 - FC LRO トランザクションのオフロードを無効にします 1 - FC LRO トランザクションのオフロードを有効にします	0



## SAS ドメインの構成

---

この章では、SAS ドメインの考慮事項、SAS デバイス発見、および SAS ブートデバイス構成について説明します。

次の項目について説明します。

- 105 ページの「SAS マルチパス化の考慮事項」
- 105 ページの「SAS デバイスの動的発見」
- 106 ページの「SAS ブートデバイスの構成」

### SAS マルチパス化の考慮事項

- SAS マルチパス化は、バンドルされている mpt ドライバを使用するときに Oracle Solaris リリース内でサポートされます。
- SAS エクスパンダは Oracle Solaris リリースではサポートされません。
- LUN マスクは Oracle Solaris リリースの SAS 接続されたデバイスではサポートされません。
- 1つのサーバーがデバイスの電源を切断しようとしているときに別のサーバーがアクセスを取得しようとしたときの予期しない結果を回避するために、SAS ドメインに接続されているシステムの電源管理を無効にします。電源管理についての詳細は、[poweradm\(1M\)](#) を参照してください。

### SAS デバイスの動的発見

mpt ドライバを使用しているときは、SAS デバイスの追加および削除は動的に実行されます。mpt ドライバで接続されている特定のターゲットおよび LUN をシステムで検出できるようにするために `sd.conf` ファイルを編集する必要はなくなりました。詳細は、[mpt\(7D\)](#) および [mpt\\_sas\(7D\)](#) を参照してください。

SAS ドメイン内でデバイスを追加または削除した場合、その存在または削除を示すメッセージが `/var/adm/messages` ファイルに書き込まれます。デバイスが追加された場合、デバイスは `format` コマンドによって表示でき、利用可能になります。

## SAS ブートデバイスの構成

Oracle Solaris OS を実行するシステムは、マルチパス化された SAS デバイスからブートしたり、SAS コントローラに接続された SATA デバイスからブートしたりできます。Oracle Solaris リリースの SAS マルチパス化については、バンドルされている `mpt` ドライバのみがサポートされています。

一部の SAS および SATA デバイスはマルチパス化をサポートしないこともあります。マルチパス化されていないデバイスとして機能します。詳細は、My Oracle サポートサイトを参照してください。

# IPFC SAN デバイスの構成

---

この章では、ホストシステムでの IPFC (Internet Protocol over Fibre Channel) の構成情報を示し、IPFC デバイスの認識と、SAN における FC 経由の IP の実装について説明します。IPFC ドライバは RFC 2625 をベースとし、IP トラフィックを FC 経由で実行できます。

次の項目について説明します。

- 107 ページの「IPFC の考慮事項」
- 110 ページの「IPFC の呼び出しと構成」

## IPFC の考慮事項

次の表は、IPFC について使用できるサポートされている機能を示しています。

表 10-1 IPFC (NFS/NAS および SNDR)

機能	サポートされています
カスケード	はい、ファブリックゾーンのみ
ゾーンのタイプ	HBA が F-port ポイントツーポイント接続として構成されたファブリックゾーン
ゾーンあたりのデバイスポートの最大数	253

次の制限が適用されます。

- IPFC は Oracle の 1G ビットスイッチではサポートされません。
- プロミスキヤス (promiscuous) モードはサポートされません。snoop ユーティリティは使用できません。
- マルチキャストはブロードキャスト経由でのみサポートされます。

- IPFCを使用するネットワークカードはルーターとして使用できません。Oracle Solaris OS では、IP 転送はデフォルトで無効です。
- IPFCを接続したあと、すべての標準のネットワークコマンドを使用できます。これらのコマンド (telnet、ping、または ftp) は、Ethernet セットアップ内と同じ方法でこの環境内で使用されます。

## ファイバチャネルアダプタのポートインスタンスの判別

このセクションでは、IPFC 向けの目的のホストシステムを構成する方法について説明します。ここでは、ポートインスタンスを判別して IPFC インスタンスを plumb するための手順も含まれています。

### ▼ ポートインスタンスを判別する方法

- 1 **HBA PCI アダプタスロットおよび I/O ボード PCI スロットを判別します。**

この情報は、108 ページの「ファイバチャネルアダプタのポートインスタンスの判別」の計算を実行するために必要です。

この手順の例では、HBA カードが PCI アダプタスロット 5 に配置され、PCI アダプタが I/O ボードのスロット 1 にあるアレイが存在すると仮定します。

- 2 **インスタンス番号を判別します。**

- a. **/etc/path\_to\_inst** ファイルの **fp** ドライババインド名を検索します。

---

注-サーバーハードウェアマニュアルに記載されているハードウェアパスを見つけて正しいエントリを判別します。

---

- b. **ステップ 1** の I/O ボードとスロット情報を使用して検索を絞り込みます。

---

注-次に示す、サーバーの物理的な位置から HBA のデバイスパスを導出する方法は、Oracle のすべての Sun サーバーハードウェアについて成功するとはかぎりません。

---

- i. **PCI アダプタスロット番号にアダプタポートの数を乗算します。**

たとえば、HBA が 2 つのポートを持つ場合、2 を乗算します。PCI アダプタスロット 5 の HBA でアレイを使用する場合、5 に 2 を乗算して 10 を導出します。

- ii. PCIアダプタ I/O ボードのスロット番号を、ステップ i で導出した数値に加算します。

PCIアダプタスロット 5 の HBA と、I/O ボードの PCI スロット 1 を使用して、10 に 1 を加算して合計は 11 となります。

- iii. ステップ ii で導出した数値を 16 進数に変換します。

数値 11 は 16 進数の「b」に変換されます。

- iv. `pci@hex` を使用して、`fp` エントリを検索します (`hex` はステップ iii で導出した数値です)。

次の表に、次のパスを持つ PCI シングル FC ネットワークアダプタデバイスについてのデバイスパスの各要素を示します。

```
"/pci@b,2000/SUNW,qlc@2/fp@0,0" 7 "fp"
```

デバイス名	値
物理名	/pci@b,2000/SUNW,qlc@2/fp@0,0
インスタンス名	7
ドライババインド名	fp

- c. 各 FP インスタンスを手動で作成します。

この例では、`interface-number` の値は `fcip7` です。

```
# ipadm create-ip fcip7
```

コマンドに成功すると、メッセージがコンソールとメッセージファイルの両方に表示されます。例:

```
Sep 13 15:52:30 bytownite ip: ip: joining multicasts failed (7) on fcip0 -
will use link layer brocasts for multicast
```

## ▼ IPFC インスタンスの作成方法

システムの各 FP インスタンスは `/dev/fc` 内にエントリを持ちます。HBA が除去されると、一部の無効なリンクが存在することがあります。IPFC をロードして作成するにはこの手順を使用します。

- 1 `/dev/fc` ファイルの各エントリについて、その HBA ポートを経由して可視となるすべてのデバイスを表示します。

```
# luxadm -e dump_map /dev/fc/fp0
```

```
Pos  Port_ID Hard_Addr Port_WWN      Node_WWN      Type
0    610100 0          210000e08b049f53 200000e08b049f53 0x1f (Unknown Type)
1    620d02 0          210000e08b02c32a 200000e08b02c32a 0x1f (Unknown Type)
2    620f00 0          210000e08b03eb4b 200000e08b03eb4b 0x1f (Unknown Type)
3    620e00 0          210100e08b220713 200100e08b220713 0x1f (Unknown Type,Host Bus Adapter)
```

```
# luxadm -e dump_map /dev/fc/fp1
No FC devices found. - /dev/fc/fp1
```

- 2 デバイスのリストに基づき、IPFC通信を確立する必要があるリモートホストから可視の宛先HBAを判別します。

この手順の例では、宛先HBAはポートID 610100 および 620d02 を持ちます。発信HBAのポートIDは 620e00 です。

- 3 宛先HBAポートを確認できる発信HBAポートの物理パスを一覧表示します。ここで、*originating-hba-link* はステップ2で判別したリンクの変数です。

```
# ls -l /dev/fc/fp originating-hba-link
```

次の例で、*originating-hba-link* の数値は 0 です。

```
# ls -l /dev/fc/fp 0
lrwxrwxrwx 1 root root 51 Sep 4 08:23 /dev/fc/fp0 ->
../../devices/pci@8,600000/SUNW,q1c@1/fp@0,0:devctl
```

- 4 ステップ3で識別した物理パスを検索します。

パス名の出力から、先頭の *../../devices* を削除する必要があります。例:

```
# grep pci@8,600000/SUNW,q1c@1/fp@0,0 /etc/path_to_inst
"/pci@8,600000/SUNW,q1c@1/fp@0,0" 0 "fp"
```

- 5 ステップ4のコマンドの出力から、発信HBAポートの *fp* インスタンスを判別します。

インスタンス番号は、出力の「fp」の前に表示されます。次の出力例で、インスタンス番号は 0 です。

```
"/pci@8,600000/SUNW,q1c@1/fp@0,0" 0 "fp"
```

- 6 ステップ5のインスタンス番号を使用してIPFCをロードし、IPFCインタフェースを作成します。

この例で、インスタンスは 0 です。

```
# ipadm create-ip fcip 0
```

## IPFCの呼び出しと構成

インストール後すぐに、*ipadm* コマンドを使用してIPFCを手動で開始します。後続のリポートでIPFCネットワークインタフェースが自動的に開始するようにホストを構成できます。このセクションでは、ネットワークインタフェースを手動で開始し、リポート時に自動 *plumb* するようホストを構成する手順について説明します。

## ▼ ネットワークインタフェースを手動で開始する方法

特定のネットマスク値でIPFCをplumbし、IPFCインタフェースを動作させるには、この手順を使用します。

- 1 管理者になります。
- 2 適切なネットワークインタフェースを構成します。  
ネットワーク管理者に、適切なIPアドレスおよびネットマスク情報を尋ねます。たとえば、fpインスタンスが0でIPアドレスが192.9.201.10のIPFCインタフェースを有効にするには、次のように入力します。

```
# ipadm create-ip fcip0
# ipadm create-addr -T static -a 192.9.201.10 fcip0/ipv4
```

詳細は、[ipadm\(1M\)](#)を参照してください。

- 3 ネットワークが動作していることを確認します。

```
# ipadm show-if
```

## ▼ 自動ネットワーク構成用にホストを構成する方法

システムのホスト名はsvc:/system/identity:nodeサービスに設定されています。たとえば、ホスト名はsys-Aで、IPFCネットワークインタフェースはfcip0です。

- 1 管理者になります。
- 2 ホスト名が設定されていることを確認します。

```
# svccfg -s identity:node
svc:/system/identity:node> listprop config/nodename
config/nodename astring sys-A
```

ホスト名を設定する必要がある場合、次のような構文を使用します。

```
# svccfg -s identity:node setprop config/nodename = "sys-A"
```

- 3 `/etc/inet/hosts` ファイルに追加入力を行います。

インストールプログラムは最低限のエントリで `/etc/inet/hosts` ファイルを作成します。テキストエディタを使用して、手動で追加入力する必要があります。追加情報については、[hosts\(4\)](#) マニュアルページ参照してください。

`/etc/inet/hosts` ファイルにはホストデータベースが格納されています。このファイルには、ホスト名とプライマリネットワークインタフェースのIPアドレスのほか

に、システムに接続されているほかのネットワークインタフェースのIPアドレスと、マシンが知っておく必要があるほかのネットワークインタフェースのIPアドレスが含まれています。

次の例はetc/inet/host ファイルを示しています。

```
127.0.0.1    localhost    loghost
192.9.200.70 neol        #This is the local host name
192.9.201.10 fcip0      #Interface to network 192.9.201.10
```

- 4 ネームサービスのSMFサービスに、ホストの **files nis** が構成されていることを確認します。

```
# svccfg
svc:> select network/nis/client:default
svc:/network/nis/client:default> select name-service/switch
svc:/system/name-service/switch> listprop config/host
config/host astring      "files nis"
svc:/system/name-service/switch> quit
```

# x86 ベースシステムのファイバチャネルデバイスからの Oracle Solaris OS のブート

---

この章では、2Gビットおよび4Gビットのファイバチャネル(FC) HBA ドライバを備えた Solaris マルチパス化 I/O 機能を含む Oracle Solaris OS を、x86 ベースシステムに手動でインストールする方法について説明します。Solaris インストールプログラムのディスク選択部分で、ブート元となるファイバチャネル(FC) デバイスを選択できます。

次の項目について説明します。

- 113 ページの「Oracle Solaris OS のセットアップ要件」
- 114 ページの「Oracle Solaris OS のインストールの概要」
- 115 ページの「Oracle Solaris OS のインストール手続き」

## Oracle Solaris OS のセットアップ要件

インストールするには次の項目が必要です。

- Oracle Solaris OS 11 または 11.1 インストール DVD - x86 ベースのシステム上に FC デバイスを構成するには、次のインストール方法が使用できます。
  - 自動インストール - 複数のクライアントシステムをネットワーク上にインストールします。単一システムをインストールするためにメディアからブートできますが、複数のクライアントシステム用にインストールをカスタマイズする場合はインストールサーバーが必要です。
  - テキストインストール - メディアまたはインストールサーバーから単一システムをインストールします。
- x86 ベースシステムに接続されている FC HBA
- ネットワークベースの OS インストール用の 10/100/1000 Mbps Ethernet ネットワーク

---

注-

Oracle の 1 Gb HBA は、x86 ベースシステム上での SAN 経由のブートをサポートしません。Oracle の 4 Gb HBA は、x86 ベースシステム上での SAN 経由のブートをサポートします。ほとんどの 2 Gb HBA は、x86 ベースシステム上での SAN 経由のブートをサポートしますが、次の場合を除きます。

- Oracle の StorageTek 2 Gb Enterprise Class Dual-Port Fibre HBA、SG-XPCI2FC-QF2-Z
  - Oracle の StorageTek 2 Gb PCI Dual-Port Fibre HBA、SG-XPCI2FC-QF2
  - Oracle の StorageTek 2 Gb FC PCI Single-Channel Network Adapter、X6767A
  - Oracle の StorageTek 2 Gb FC PCI Dual-Channel Network Adapter、X6768A
  - Oracle の StorageTek 2 Gb FC PCI-X Enterprise Single-Port HBA、SG-XPCI1FC-QL2
  - Oracle の StorageTek 2 Gb FC PCI Single-Port Host Adapter、SG-XPCI1FC-QF2
- 

## Oracle Solaris OS のインストールの概要

インストールを正しく完了するためには、インストールのディスク選択部分で FC ベースのデバイスを使用します。対話式インストールの最後に、x86 BIOS および FC HBA BIOS を変更して、マルチパス化されたリモートディスクから Oracle Solaris をブートするために使用する FC イニシエータを特定する必要があります。

OS のインストール後、リブートの前に、新しくインストールされた論理ユニット番号 (LUN) に対して `luxadm` コマンドを発行して構成情報を収集します。`luxadm` コマンド出力には `c#t#d#` からアレイ WWN および LUN へのマップが提供されます。HBA WWN (World Wide Name) およびアレイ WWN ポート情報を記録します。

リブート中に、WWN および LUN を使用して、各 HBA の HBA BIOS が、同じアレイ上の各 LUN からのブートに使用されるように設定します。CD-ROM またはネットワークでなくディスクからブートを実行するようにシステム BIOS を変更します。

---

注 - HBA およびシステム BIOS を変更するにはコンソールアクセスが必要です。

---

さらに、Oracle Solaris OS のインストール中は次のことに注意してください。

- カスタムのディスクレイアウトを使用中の場合、オーバーラップ (s2) パーティションを削除しないでください。x86 ブートインストーラはこのパーティションに依存します。
- デフォルトでは、Solaris I/O マルチパス化機能は単一のブートデバイスへの複数パスを持つ FC ブートデバイスを管理します。

# Oracle Solaris OS のインストール手続き

## ▼ Oracle Solaris OS のインストール方法

- 1 HBAハードウェアを設置します。

<http://www.oracle.com/technetwork/documentation/oracle-storage-networking-190061.html>にある該当する

Oracle HBA インストールガイドの指示に従います。

- 2 Oracle Solaris OS をインストールします。

自動インストールを選択し、インストール中にインストール先となる特定のデバイスを選択するには、『Oracle Solaris 11.1 システムのインストール』を参照してください。

## ▼ DVD またはネットワークベースの OS インストールを実行する方法

HBA を設置したあと、x86 ベースのシステム上で Oracle Solaris OS の DVD インストールまたはネットワークベースのインストールを行うには、次のステップを実行します。

詳細は、『Oracle Solaris 11.1 システムのインストール』を参照してください。

- 1 ネットワークでなく DVD-ROM からインストールする場合、Oracle Solaris ソフトウェア DVD を DVD-ROM ドライブに挿入します。
- 2 初期の電源投入時に、システム BIOS を準備し、ネットワークまたは DVD-ROM のいずれか該当する方からブートするよう構成します。
- 3 次のいずれかの方法を選択して Oracle Solaris OS をインストールします。
  - 自動インストール - x86 ベースシステムの BIOS からネットワークブートを選択することによって、自動インストールを開始できます。自動インストールを選択する場合、ステップ7にスキップします。
  - テキストインストール - メディアからブートするか、インストールサーバーからブートする場合に GRUB メニューから次のオプションを選択することによって、テキストインストールを開始できます。

Oracle Solaris 11.1 Text Installer and command line

インストールが完了したあと、インストーラを終了してデバイスを構成できません。

- 4 テキストインストール-目的のレイとレイに関連付けられた LUN を選択します。
- 5 テキストインストール-各インストールメニューから目的のインストールオプションを選択することによって、インストールを続行します。
- 6 テキストインストール-インストール画面の最後で選択内容を確認し、Oracle Solaris OS のインストールを開始します。
- 7 インストールが完了したあと、自動インストールまたはテキストインストールのいずれを実行したかに基づき、次のいずれかを選択します。
  - 自動インストール-デフォルトでは、システムはインストール後にリブートされません。これは、`/usr/share/auto_install/default.xml` ファイルに次のマニフェストキーワードがあるためです。これは、システムがリブートする前にデバイスを構成できることを意味します。

```
<auto_install>
  <ai_instance name="default">
  .
  .
  .
```

以前のインストールで次のキーワード値を `true` に設定した場合、この値を `false` に変更して、システムがリブートする前にデバイスを構成できるようにします。

```
<auto_install>
  <ai_instance name="default" auto_reboot="true">
  .
  .
  .
```

- テキストインストール-インストールが完了したら、デバイスを構成するために `Quit` オプションを選択してインストーラを終了します。
- 8 インストールが完了したあと、リブートする前に、インストール中に選択された LUN について `luxadm display` コマンドを発行します。

図 11-1 を参照してください。

```
# luxadm display /dev/rdisk/c0t600015D000202800000000000001142d0s2
DEVICE PROPERTIES for disk: /dev/rdisk/c0t600015D000202800000000000001142d0s2
Vendor:      SUN
Product ID:  SE6920
Revision:    0202
Serial Num:  00500057
Unformatted capacity: 10240.000 MBytes
Read Cache:  Enabled
  Minimum prefetch: 0x0
  Maximum prefetch: 0xffff
Device Type:  Disk device
Path(s):

/dev/rdisk/c0t600015D000202800000000000001142d0s2
/devices/scsi_vhci/disk@g600015d000202800000000000001142:c,raw
```

```

Controller    /dev/cfg/c4
Device Address 213600015d207200,0
Host controller port WWN      210100e08b206812
Class          primary
State          ONLINE
Controller    /dev/cfg/c11
Device Address 213600015d207200,0
Host controller port WWN      210100e08b30a2f2
Class          primary
State          ONLINE

```

図 11-1 luxadm display コマンドおよび出力

```

QLogic FastUTIL
=====Selected Adapter=====
Adapter Type I/O Address Slot Bus Device Function
QLA2462      3400          02  03  01    1
=====Adapter Settings=====
BIOS Address:           CF000
BIOS Revision:          1.05
Adapter Serial Number:  A04712
Interrupt Level:        5
Adapter Port Name:      210100E08B206812
Host Adapter BIOS:      Enabled
Frame Size:             2048
Loop Reset Delay:       5
Adapter Hard Loop ID:   Disabled
Hard Loop ID:           0
Spinup Delay:           Disabled
Connection Options:     2
Fibre Channel Tape Support: Enabled
Data Rate:              2
=====
Use <Arrow keys> and <Enter> to change settings, <Esc> to exit

```

図の例にある次の luxadm コマンド出力は、MPxIO ベースの c#t#d# を HBA WWN およびアレイ WWN にマップするために使用できます。

- MPxIO c#t#d# = c0t600015d0002028000000000000001142d0
- アレイ WWN = 213600015d207200, LUN 0
- HBA WWNs = 210100e08b206812 and 210100e08b30a2f2

- 9 リブートプロセス中に、1 番目の HBA の BIOS 画面に入ることができるようにモニターを監視し、ブートデバイスが、Oracle Solaris OS を先ほどインストールした FC LUN になるように指定します。

マルチパス化に使用する各 HBA についてこのステップに従い、ブートデバイスが、Oracle Solaris OS をインストールした FC LUN になるように指定します。図 11-2 および図 11-3 を参照してください。

- QLogic HBA BIOS について、次のステップを実行します。
  - a. ホストのリブート中に、Ctrl-Q キーを押して HBA BIOS 画面を表示します。
  - b. ブートデバイスとして有効にしてブートを有効にする HBA を選択します。
  - c. ブートデバイスを次のように構成します。
    - i. 「Configuration Settings」を選択します。
    - ii. 「Selectable Boot Settings」を選択します。
    - iii. 「Selectable Boot」が有効に設定されていることを確認します。  
このメニューでは、ブートデバイス/LUN をアレイ WWPN によって選択できます。
    - iv. HBA BIOS 画面を保存して終了します。
- Emulex HBA BIOS について、次のステップを実行します。
  - a. ホストのリブート中に、Alt-E キーを押して HBA BIOS 画面を表示します。
  - b. ブートデバイスとして有効にしてブートを有効にする HBA を選択します。
  - c. 「Configure Boot Devices」を選択します。
  - d. ブートエントリを選択する。
  - e. 目的のブートデバイスの WWPN を選択します。
  - f. LUN 番号を入力します。
  - g. ブート LUN を選択します。
  - h. ブートデバイスとアレイ WWPN を選択します。
  - i. HBA BIOS 画面を保存して終了します。

図 11-2 HBA WWN についての HBA BIOS 画面

```
QLogic Fast!UTIL
=====Selected Adapter=====
Adapter Type I/O Address Slot Bus Device Function
QLA2462      3400          02  03  01    1
=====Adapter Settings=====
BIOS Address:                CF000
BIOS Revision:               1.05
Adapter Serial Number:      A04712
Interrupt Level:            5
Adapter Port Name:          210100E08B206812
Host Adapter BIOS:          Enabled
Frame Size:                  2048
Loop Reset Delay:           5
Adapter Hard Loop ID:       Disabled
Hard Loop ID:                0
Spinup Delay:                Disabled
Connection Options:         2
Fibre Channel Tape Support: Enabled
Data Rate:                   2
=====
Use <Arrow keys> and <Enter> to change settings, <Esc> to exit
```

図 11-3 FCLUN からのブートを有効にするための HBA BIOS 画面

```
QLogic FastUTIL
=====Selected Adapter=====
| Adapter Type I/O Address Slot Bus Device Function |
| QLA2462      3000          02  03  01  0          |
=====
=====Selectable Boot Settings=====
| Selectable Boot: Enabled |
| (Primary) Boot Port Name, Lun: 213600015D207200, 0 |
| Boot Port Name, Lun: 213600015D207200, 1 |
| Boot Port Name, Lun: 213600015D207200, 2 |
| Boot Port Name, Lun: 213600015D207200, 3 |
| Press "C" to clear a Boot Port Name entry |
=====
Use <Arrow keys> and <Enter> to change settings, <Esc> to exit
```

注-

この図では次の変更を示しています。

- Selectable boot = Enabled
- ARRAY WWN = 213600015d207200
- ARRAY LUN = 0
- HBA WWN = 210100e08b206812

- 10 すべての HBA およびすべての LUN について適切な変更を繰り返します。
- 11 ベンダーのアクセス方法に従ってシステム BIOS を入力し、ブートデバイスが Oracle Solaris OS をインストールした FCLUN になるように指定します。
- 12 システム BIOS 内で指定された FC LUN を使用して、新しくインストールされた Oracle Solaris OS をリブートします。

# テープデバイス用の持続的なバインド

---

この章では、`/dev` エントリが SAN の複数のサーバーにわたって同一であるようにするために、テープデバイス用の持続的なバインドを作成する方法について説明します。

次の項目について説明します。

- 121 ページの「持続的なバインドの概要」
- 122 ページの「テープリンクの作成」

## 持続的なバインドの概要

SAN ベースのデータセンター内でサーバーの管理を簡素化するために、Oracle の Solaris OS の StorageTek SAN Foundation ソフトウェアスタックは SAN 内のデバイスを動的に検出し、ユーザーによる構成ファイルの編集を必要とすることなく、関連する `/dev` ツリーエントリを構築します。

ほとんどの場合、このプロセスは SAN 管理を大幅に簡素化します。ただし、テープデバイスについては、`/dev` エントリが作成される方法を明示的に指定し、`/dev` エントリが SAN の複数のサーバーにわたって同一であるようにする機能を求める場合もあります。この章では、ディスクベースのデバイスについての自動発見のメリットを保持しながら Solaris OS でこのテープバインドを指定する方法について説明します。

`/dev/rmt` ディレクトリには、テープデバイスについての物理デバイスへのリンクが `/devices` の下に含まれています。システムによって認識される各テープ LUN は、`/dev/rmt/N`、`/dev/rmt/Nb`、および `/dev/rmt/Nbn` という形式の 24 個のマイナーノードによって表現されます。ここで、`N` は 0 から始まる整数カウンタです。この数値は新規デバイスの列挙中に `devfsadm` によって選択されます。`devfsadm` によって検出される新しいテープ論理ユニット番号 (LUN) は、`/dev/rmt` 内で次に使用できる番号を取得します。

`/dev/rmt` の名前はデバイスツリー内でデバイスが表示される順序に依存するため、システム間で変化します。複数の異なるシステムによって認識される特定のテープドライブについて、`/dev/rmt` リンクはこれらのシステムごとに異なる可能性があります。この違いは、Symantec (VERITAS) NetBackup (SSO オプション) のほとんどの一般的な使用方法について問題となる可能性があります。また、ドライブが交換された場合、ドライブのポート World Wide Name (PWPN) を保持する方法をベンダーが提供しないかぎり、リンクは変更されます。

## テープリンクの作成

`/etc/devlink.tab` ファイルはデフォルトデバイステーブルファイルと呼ばれます。これは、`/dev` ディレクトリ内でリンクを作成するために `devfsadm` が使用するルールを指定します。このファイルにはテープのエントリが含まれません。この理由は、`devfsadm` はテープドライブのリンクを作成できますが、ルールが追加されて、テープリンクを作成するためのデフォルト動作が変更されることがあるためです。詳細は、[devlinks\(1M\)](#) を参照してください。

Oracle Solaris OS から可視だが `devlink` ファイル内に指定されないテープドライブについて、`devfsadm` は 0 から始まるマイナーノード番号を自動的に割り当てます。これらのマイナーノード番号は、`/etc/devlink.tab` 内で手動で割り当てた低い番号と競合するため、競合を回避するための十分な高さの番号を割り当ててください。

このアプローチでは、重複するリンクが `/dev/rmt` 内に容易に生成されます。エントリが `/etc/devlink.tab` 内に指定されるよりも前に発見されたテープは、リンクが自動的に作成されます。エントリが追加されて `devfsadm` が実行される時、元のリンクは `/dev/rmt` 内に残るため、結果として重複するリンクとなります。`/dev/rmt` 内の元のリンクを削除するには、`rm /dev/rmt/*` コマンドを実行してから `devfsadm` を実行します。

このアプローチは、複数の HBA ポートに接続された複数ポートのテープドライブでは使用できません。同じテープ LUN に対して複数の HBA ポートが接続されている場合、システムは 1 つでなく 2 つのテープドライブを検出します。`prtconf` の出力で最後に表示された方が、`/etc/devlink.tab` によって生成されたリンクを受け取ります。

次の例は、`devlink.tab` ファイル内でのテープについてのサンプルエントリを示しています。

```
type=ddi_byte:tape;addr=PWPN,LUN-number; rmt/rmt-number\M0
```

`rmt #` は、`/dev/rmt/ N` が必要とするものに変更します。次に、目的のテープデバイスと一致するように PWPN および LUN を変更します。この値は次に示すように、既存の `/dev/rmt/` リンクに対して `ls -l` コマンドを実行することによって取得できます。

```
# ls -l /dev/rmt/4
lrwxrwxrwx 1 root root 69 Oct 6 14:57 /dev/rmt/4 ->
../../../../devices/pci@1f,700000/SUNW,qlc@2/fp@0,0/st@w5005076300617717,0:
```

たとえば、`/dev/rmt/`番号を40にする場合、次の例のようなエントリを `/etc/devlink.tab` 内に作成します。

```
# type=ddi_byte:tape;addr=w5005076300617717,0; rmt/40\M0
```

次に、このドライブを使用する SAN 上の Solaris サーバーの `devlink` ファイルにこの行を追加します。これによって、ドライブは常にマイナーノード40として表示されます。

## ▼ テープデバイスリンクの作成方法

- 1 管理者になります。
- 2 [122 ページの「テープリンクの作成」](#)で説明されているように、`/etc/devlink.tab` 内にエントリを作成します。  
`devfsadm` が以前デバイスを発見した場合、既存のリンク上で `ls -l` コマンドを実行してデバイスアドレスを判別する必要があります。

---

注-前に説明したように、自動的に構成されたデバイスとの競合を回避するように `/dev/rmt/N` 番号を割り当ててください。

---

- 3 `rm /dev/rmt/*` コマンドを実行することによって、`/dev/rmt` から既存のリンクを削除します。
- 4 `devfsadm` を実行します。  
このコマンドは、不特定のデバイスのリンクを自動的に作成するほか、`/etc/devlink.tab` のエントリのとおりリンクを作成します。



# ファブリック接続されたデバイスの手動構成

---

この付録では、Oracle Solaris OS のファブリックデバイスを構成したり、構成を解除したりする方法について説明します。ここでは、ホスト上で可視のファブリックデバイスを検出し、マルチパス化ソフトウェアを有効に構成したり有効にしないように構成したりする方法について説明します。

次の項目について説明します。

- 125 ページの「FC デバイスの手動構成」
- 126 ページの「ファブリックデバイスノードの構成」
- 128 ページの「マルチパス化が有効にされていないデバイスノードの構成」
- 131 ページの「Solaris マルチパス化が有効にされているデバイスノードの構成」
- 135 ページの「ファブリックデバイスの構成の解除」

## FC デバイスの手動構成

Oracle Solaris リリースでは、ファブリック接続されたデバイスは、自動的に Oracle Solaris システムから使用可能になります。

ファブリック接続されたデバイスを手動で構成する場合、次のステップを使用してデフォルトの動作を変更します。

---

注-デフォルトの動作を変更すると、ファブリック接続されたすべてのデバイスが使用できなくなり、ブート時に使用可能となることが必要なファブリック接続されたデバイスに問題が生じる可能性があります。

---

## ▼ FC デバイスを手動構成する方法

- 1 管理者になります。
- 2 `/kernel/drv/fp.conf` ファイルを `/etc/driver/drv/fp.conf` ファイルにコピーします。
- 3 `/etc/driver/drv/fp.conf` ファイルの次の行をコメント解除することによって、手動構成を有効にします。  

```
manual_configuration_only=1;
```

この設定に関する追加情報については、`fp(7D)` のマニュアルページと `cfgadm_fp(1M)` のマニュアルページを参照してください。
- 4 システムをリブートします。
- 5 ファブリック接続された各デバイスを使用可能にするには、**Solaris I/O** マルチパス化機能を使用しているかどうかに応じて、次のいずれかのタスクを選択します。
  - 128 ページの「マルチパス化が有効にされていないデバイスノードの構成」
  - 131 ページの「Solaris マルチパス化が有効にされているデバイスノードの構成」

ファブリック接続されたデバイスの元のデフォルト動作を使用する場合、次のステップを参照してください。
- 6 `/etc/driver/drv/fp.conf` ファイルの次の行をコメント化して、手動構成を無効にします。  

```
# manual_configuration_only=1;
```

この設定に関する詳細は、`cfgadm_fp(1M)` および `fp(7d)` を参照してください。
- 7 システムをリブートします。  

```
# init 6
```

## ファブリックデバイスノードの構成

直接接続システムまたは SAN 内でハードウェアを構成したあと、システムがデバイスを認識するようになる必要があります。このセクションでは、SAN の 24 ビット FC アドレス指定デバイスとしても知られているファブリックデバイスのホスト認識について説明します。SAN のデバイス、ポート、およびゾーンを構成したあと、システムがデバイスを認識するようにします。FC サポートを持つ 1 つの SAN には最大 1,600 万個のファブリックデバイスを一緒に接続できます。

このセクションでは、Oracle Solaris OS の視点から必要となる操作に限定します。ここでは、デバイスの使用条件やデバイス固有の管理などのほかの側面については扱

いません。デバイスがボリュームマネージャーなどのほかのソフトウェアによって管理されている場合、追加の指示についてはボリュームマネージャーの製品ドキュメントを参照してください。

## LUN レベル情報が表示できることの確認

### ▼ LUN レベル情報が表示できることを確認する方法

- 1 管理者になります。
- 2 LUN レベル情報を識別します。

```
# cfgadm -al -o show_SCSI_LUN
```

`cfgadm -al -o show_SCSI_LUN controller-ID` コマンドをシステムのブート直後に発行した場合、ファイバチャネルプロトコル (FCP) SCSI LUN レベル情報が出力に表示されないことがあります。この情報が表示されない理由は、`ssd` や `st` ドライバなどのストレージデバイスドライバが、実行中のシステムにまだロードされていないためです。

- 3 ドライバがロードされたかどうかを判定します。

例:

```
# modinfo | grep ssd
```

ドライバがロードされたあと、LUN レベル情報は `cfgadm` 出力で表示されます。

### ▼ システム上で可視のファブリックデバイスを検出する方法

このセクションでは、FC ホストポート `c0` および `c1` を使用するファブリックデバイスを検出するための手順の例を示します。この手順では、`cfgadm` コマンドで表示されるデバイス構成情報についても示します。

---

注- 次の例では、フェイルオーバーパスの接続点 ID (Ap\_Id) のみが一覧表示されます。システムに表示される Ap\_Id は、システム構成に依存します。

---

- 1 管理者になります。
- 2 システムの接続点についての情報を表示します。

```
# cfgadm -l
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	fc-fabric	connected	unconfigured	unknown
c1	fc-private	connected	configured	unknown

この例で、c0はファブリック接続されたホストポートを表し、c1はループ接続されたプライベートホストポートを表しています。cfgadm コマンドを使用して、ファブリック接続されたホストポート上のデバイス構成を管理します。

デフォルトでは、ループ接続されたプライベートホストポート上でのデバイス構成は、Oracle Solaris OS を実行中のシステムによって管理されます。

- 3 ホストポートおよびそれらに接続されているデバイスについての情報を表示します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant  Condition
c0             fc-fabric    connected   unconfigured unknown
c0::50020f2300006077 disk         connected   unconfigured unknown
c0::50020f23000063a9 disk         connected   unconfigured unknown
c0::50020f2300005f24 disk         connected   unconfigured unknown
c0::50020f2300006107 disk         connected   unconfigured unknown
c1             fc-private   connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708ba7d832 disk         connected   configured  unknown
c1::220203708b8d45f2 disk         connected   configured  unknown
c1::220203708b9b20b2 disk         connected   configured  unknown
```

注 - cfgadm -l コマンドは、FC ホストポートについての情報を表示します。cfgadm -al コマンドを使用しても FC デバイスについての情報を表示できます。c0 に関連付けられたポート WWN (World Wide Name) を Ap\_Id フィールドに含む行はファブリックデバイスを表します。これらのデバイスを管理し、Oracle Solaris OS を使用中のシステムでこれらが使用できるようにするには、cfgadm configure および unconfigure コマンドを使用します。ポート WWN が c1 に属する Ap\_Id デバイスは、c1 ホストポート経由で構成されたプライベートループデバイスを表します。

## マルチパス化が有効にされていないデバイスノードの構成

このセクションでは、マルチパス化が有効にされていないシステム上でのファブリックデバイス構成タスクについて説明します。

このセクションの手順では、システム上で可視のファブリックデバイスを検出する方法と、ファブリックデバイスを構成して Oracle Solaris OS を実行中のシステムで使用できるようにする方法を示します。このセクションの手順では、特定のデバイスを例として使用して、cfgadm コマンドを使用してファブリックデバイスを検出および構成するための方法を示します。

提供するデバイス情報と、cfgadm コマンドによって表示されるデバイス情報は、システム構成に依存します。

## ▼ マルチパス化されていないFCデバイスを手動構成する方法

このサンプル手順では、ファブリック接続されたホストポート c0 に接続されたファブリックデバイスを構成する方法について説明します。

- 1 管理者になります。
- 2 構成するデバイスを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric     connected   unconfigured unknown
c0::50020f2300006077 disk         connected   unconfigured unknown
c0::50020f23000063a9 disk         connected   unconfigured unknown
c0::50020f2300005f24 disk         connected   unconfigured unknown
c0::50020f2300006107 disk         connected   unconfigured unknown
c1             fc-private    connected   configured   unknown
c1::220203708b69c32b disk         connected   configured   unknown
c1::220203708ba7d832 disk         connected   configured   unknown
c1::220203708b8d45f2 disk         connected   configured   unknown
c1::220203708b9b20b2 disk         connected   configured   unknown
```

- 3 ファブリックデバイスを構成します。

```
# cfgadm -c configure c0::50020f2300006077
```

- 4 選択されたファブリックデバイスが構成されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric     connected   configured   unknown
c0::50020f2300006077 disk         connected   configured   unknown
c0::50020f23000063a9 disk         connected   unconfigured unknown
c0::50020f2300005f24 disk         connected   unconfigured unknown
c0::50020f2300006107 disk         connected   unconfigured unknown
c1             fc-private    connected   configured   unknown
c1::220203708b69c32b disk         connected   configured   unknown
c1::220203708ba7d832 disk         connected   configured   unknown
c1::220203708b8d45f2 disk         connected   configured   unknown
c1::220203708b9b20b2 disk         connected   configured   unknown
```

c0 および c0::50020f2300006077 の両方の Occupant 列は configured と表示されており、これは c0 ポートには構成済みの占有装置があり、c0::50020f2300006077 デバイスが構成済みであることを示しています。

- 5 マルチ LUN SCSI デバイスについての FCP SCSI LUN 情報を表示します。

次のコーディング例は、Ap\_Id c0:50020f2300006077 を経由して接続されている物理デバイスに4つのLUNが構成されていることを示しています。

```
# cfgadm -al -o show_SCSI_LUN c0
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric     connected   configured   unknown
c0::50020f2300006077,0 disk         connected   configured   unknown
```

```

c0::50020f2300006077,1 disk      connected  configured  unknown
c0::50020f2300006077,2 disk      connected  configured  unknown
c0::50020f2300006077,3 disk      connected  configured  unknown

```

デバイスは、Oracle Solaris OS を実行中のシステム上で使用できるようになっています。パスは、c0::50020f2300006077 によって表現される物理デバイス内の各 SCSI LUN を示します。

## ▼ マルチパス化されていない複数の FC デバイスを構成する方法

127 ページの「LUN レベル情報が表示できることの確認」の手順を使用して、最初にシステムから可視のデバイスを識別します。この手順では、ファブリック接続されたホストポートに接続された、構成が解除されているすべてのファブリックデバイスを構成する方法について説明します。例として使用するポートは c0 です。

- 1 管理者になります。
- 2 構成するデバイスを確認します。

```

# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric    connected   unconfigured unknown
c0::50020f2300006077 disk         connected   unconfigured unknown
c0::50020f23000063a9 disk         connected   unconfigured unknown
c0::50020f2300005f24 disk         connected   unconfigured unknown
c0::50020f2300006107 disk         connected   unconfigured unknown
c1             fc-private   connected   configured   unknown
c1::220203708b69c32b disk         connected   configured   unknown
c1::220203708ba7d832 disk         connected   configured   unknown
c1::220203708b8d45f2 disk         connected   configured   unknown
c1::220203708b9b20b2 disk         connected   configured   unknown

```

- 3 選択されたポート上で構成が解除されているすべてのデバイスを構成します。

```
# cfgadm -c configure c0
```

---

注 - この操作は、個々のデバイスの configure 操作を、c0 上のすべてのデバイスに対して繰り返します。c0 上のデバイス数が多い場合、時間がかかることがあります。

---

- 4 c0 上のすべてのデバイスが構成済みであることを確認します。

```

# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric    connected   configured   unknown
c0::50020f2300006077 disk         connected   configured   unknown
c0::50020f23000063a9 disk         connected   configured   unknown
c0::50020f2300005f24 disk         connected   configured   unknown
c0::50020f2300006107 disk         connected   configured   unknown
c1             fc-private   connected   configured   unknown

```

```

c1::220203708b69c32b disk      connected  configured  unknown
c1::220203708ba7d832 disk      connected  configured  unknown
c1::220203708b8d45f2 disk      connected  configured  unknown
c1::220203708b9b20b2 disk      connected  configured  unknown

```

##### 5 マルチ LUN SCSI デバイスについての FCP SCSI LUN 情報を表示します。

次のコーディング例は、c0::50020f2300006077 および c0::50020f2300006107 によって表現される物理デバイスは、それぞれ4つのLUNが構成されていることを示しています。c0::50020f23000063a9 および c0::50020f2300005f24 によって表現される物理デバイスは、それぞれ2つのLUNが構成されています。

```

# cfgadm -al -o show_SCSI_LUN c0
Ap_Id          Type      Receptacle  Occupant    Condition
c0              fc-fabric connected    configured  unknown
c0::50020f2300006077,0 disk      connected    configured  unknown
c0::50020f2300006077,1 disk      connected    configured  unknown
c0::50020f2300006077,2 disk      connected    configured  unknown
c0::50020f2300006077,3 disk      connected    configured  unknown
c0::50020f23000063a9,0 disk      connected    configured  unknown
c0::50020f23000063a9,1 disk      connected    configured  unknown
c0::50020f2300005f24,0 disk      connected    configured  unknown
c0::50020f2300005f24,1 disk      connected    configured  unknown
c0::50020f2300006107,0 disk      connected    configured  unknown
c0::50020f2300006107,1 disk      connected    configured  unknown
c0::50020f2300006107,2 disk      connected    configured  unknown
c0::50020f2300006107,3 disk      connected    configured  unknown

```

## Solaris マルチパス化が有効にされているデバイスノードの構成

このセクションでは、マルチパス化機能が有効にされているシステム上でファブリックデバイス構成ステップを実行する方法について説明します。

ファブリック接続されたホストポートに接続されているデバイスは、デフォルトで構成されていないため、システムから使用できません。cfgadm configure および cfgadm unconfigure コマンドを使用して、ファブリックデバイスのためのデバイスノード作成を管理します。詳細は、[cfgadm\\_fp\(1M\)](#)を参照してください。このセクションの手順では、システム上で可視のファブリックデバイスを検出し、ファブリックデバイスをマルチパス化デバイスとして構成してシステムで使用できるようにするステップを示します。

提供するデバイス情報と、cfgadm コマンドによって表示されるデバイス情報は、システム構成に依存します。

## ▼ マルチパス化された個々の FC デバイスを構成する方法

このサンプル手順では、ファブリック接続されたホストポート `c0` および `c2` を使用して、マルチパスソフトウェアが有効にされたシステム上でファブリックデバイスをマルチパス化されたデバイスとして構成します。

ファブリックデバイス用の `cfgadm -c configure` コマンドは、マルチパス化が有効にされるかどうかにかかわらず同じです。

- 1 管理者になります。
- 2 マルチパス化されたデバイスとして構成されるデバイスのポート **WWN** を識別します。

ファブリック接続されたホストポート上にある、`fc-fabric` とマークされているデバイスを探します。これらのデバイスは、`cfgadm -c configure` コマンドを使用して構成できるデバイスです。

```
# cfgadm -al
Ap_Id                Type      Receptacle Occupant  Condition
c0                   fc-fabric connected unconfigured unknown
c0::50020f2300006077 disk      connected unconfigured unknown
c0::50020f23000063a9 disk      connected unconfigured unknown
c1                   fc-private connected configured unknown
c1::220203708b69c32b disk      connected configured unknown
c1::220203708ba7d832 disk      connected configured unknown
c1::220203708b8d45f2 disk      connected configured unknown
c1::220203708b9b20b2 disk      connected configured unknown
c2                   fc-fabric connected unconfigured unknown
c2::50020f2300005f24 disk      connected unconfigured unknown
c2::50020f2300006107 disk      connected unconfigured unknown
```

上の例で、`c0::50020f2300006077` および `c2::50020f2300006107` `Ap_Id` は、複数のストレージデバイスコントローラについて異なるポート **WWN** を持つ同一のストレージデバイスを表します。`c0` および `c2` ホストポートはマルチパス化のために有効にされています。

- 3 ファブリックデバイスを構成して、デバイスをシステムから使用できるようにします。

```
# cfgadm -c configure c0::50020f2300006077 c2::50020f2300006107
```

- 4 選択されたデバイスが構成されていることを確認します。

```
# cfgadm -al
Ap_Id                Type      Receptacle Occupant  Condition
c0                   fc-fabric connected configured unknown
c0::50020f2300006077 disk      connected configured unknown
c0::50020f23000063a9 disk      connected unconfigured unknown
c1                   fc-private connected configured unknown
c1::220203708b69c32b disk      connected configured unknown
c1::220203708ba7d832 disk      connected configured unknown
```

```

c1::220203708b8d45f2 disk      connected  configured  unknown
c1::220203708b9b20b2 disk      connected  configured  unknown
c2                fc-fabric  connected  configured  unknown
c2::50020f2300005f24 disk      connected  unconfigured unknown
c2::50020f2300006107 disk      connected  configured  unknown

```

c0 および c0::50020f2300006077 の Occupant 列は configured と指定されており、これは c0 ポートには少なくとも 1 つの構成済みの占有装置があり、c0::50020f2300006077 デバイスが構成済みであることを示しています。c2 および c2::50020f2300006107 についても同じ変更が実行されています。

構成操作がエラーを出さずに完了した場合、マルチパス化されたデバイスがシステム上に作成されます。c0::50020f2300006077 および c2::50020f2300006107 で表現される物理デバイスに複数の SCSI LUN が構成された場合、それぞれの LUN がマルチパス化されたデバイスとして構成されます。次の例は、c0::50020f2300006077 および c2::50020f2300006107 を使用して 2 つの LUN が構成されることを示しています。各 Ap\_Id はマルチパス化されたデバイスへのパスに関連付けられます。

```

# cfgadm -al -o show_SCSI_LUN c0::50020f2300006077\ c2::50020f2300006107
Ap_Id      Type      Receptacle  Occupant  Condition
c0::50020f2300006077,0 disk      connected   configured unknown
c0::50020f2300006077,1 disk      connected   configured unknown
c2::50020f2300006107,0 disk      connected   configured unknown
c2::50020f2300006107,1 disk      connected   configured unknown

```

上の例は、c0::50020f2300006077 および c2::50020f2300006107 によって表現されるデバイスについて次の 2 つのマルチパス化されたデバイスが作成されることを示しています。

## ▼ マルチパス化された複数の FC デバイスを構成する方法

デバイスノードを構成または削除する前に、127 ページの「LUN レベル情報が表示できることの確認」の手順を使用して、最初にファブリックデバイスを識別するようにしてください。

この例で、ファブリック接続されたホストポート上の Ap\_Id は、マルチパス化されたデバイスへのパスです。たとえば、c2 を経由したパスを持つすべてのデバイスは構成されますが、c0 を経由するものは構成されません。c2 はシステムからファブリックへの接続点で、c2::50020f2300006107 はストレージからファブリックへの接続点です。システムは、そのシステムが構成される目的のファブリック内のすべてのストレージデバイスを検出します。

すでに構成されたデバイス上の Ap\_Id を、別の Ap\_Id を使用して構成すると、以前構成されたデバイスにパスが追加されます。この場合は新しいデバイスノードは作成されません。デバイスノードは、対応するデバイスへの Ap\_Id が初めて構成される時のみ作成されます。

- 1 管理者になります。
- 2 構成されるファブリック接続されたホストポートを識別します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric    connected   configured  unknown
c0::50020f2300006077 disk         connected   configured  unknown
c0::50020f23000063a9 disk         connected   configured  unknown
c1             fc-private   connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708ba7d832 disk         connected   configured  unknown
c1::220203708b8d45f2 disk         connected   configured  unknown
c1::220203708b9b20b2 disk         connected   configured  unknown
c2             fc-fabric    connected   unconfigured unknown
c2::50020f2300005f24 disk         connected   unconfigured unknown
c2::50020f2300006107 disk         connected   unconfigured unknown
```

Ap\_Id が c0::50020f2300006077 および c2::50020f2300006107 で表現されるデバイスは、同じ物理デバイスへの2つのパスで、c0::50020f2300006077 はすでに構成されています。選択されたポート上で構成が解除されているデバイスを構成します。この操作は、個々のデバイスの configure コマンドを、c2 上のすべてのデバイスに対して繰り返します。c2 上のデバイス数が多い場合、時間がかかることがあります。

```
# cfgadm -c configure c2
```

- 3 c2 上のすべてのデバイスが構成済みであることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric    connected   configured  unknown
c0::50020f2300006077 disk         connected   configured  unknown
c0::50020f23000063a9 disk         connected   configured  unknown
c1             fc-private   connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708ba7d832 disk         connected   configured  unknown
c1::220203708b8d45f2 disk         connected   configured  unknown
c1::220203708b9b20b2 disk         connected   configured  unknown
c2             fc-fabric    connected   configured  unknown
c2::50020f2300005f24 disk         connected   configured  unknown
c2::50020f2300006107 disk         connected   configured  unknown
```

c2 と、c2 の下にあるすべてのデバイスの Occupant 列に configured のマークが付いていることに注目してください。

show\_SCSI\_LUN コマンドは、複数の LUN SCSI デバイスについての FCP SCSI LUN 情報を表示します。次のコーディング例は、c2::50020f2300006107 および c2::50020f2300005f24 によって接続される物理デバイスで、それぞれ2つの LUN が構成されていることを示しています。

```
# cfgadm -al -o show_SCSI_LUN c2
Ap_Id          Type          Receptacle  Occupant    Condition
c2             fc-fabric    connected   configured  unknown
c2::50020f2300005f24,0 disk         connected   configured  unknown
c2::50020f2300005f24,1 disk         connected   configured  unknown
```

```
c2::50020f2300006107,0 disk    connected    configured    unknown
c2::50020f2300006107,1 disk    connected    configured    unknown
```

## ファブリックデバイスの構成の解除

このセクションでは、ファブリックデバイスのマルチパス化機能の構成を解除することについて説明します。

### ファブリックデバイスの構成の解除

ファブリックデバイスの構成を解除する前に、デバイスに対するすべての活動を停止し、ファブリックデバイス上のファイルシステムをアンマウントします。アンマウントの指示については、Oracle Solaris 管理のドキュメントを参照してください。デバイスがほかのボリュームマネージャーの制御下にある場合、デバイスの構成を解除する前にボリュームマネージャーのドキュメントを参照してください。

#### ▼ FCデバイスの構成を手動で解除する方法

この手順では、ファブリック接続されたホストポート `c0` に接続されたファブリックデバイスの構成を解除する方法について説明します。

- 1 管理者になります。
- 2 構成を解除するデバイスを確認します。  
ファブリック接続されたホストポート上のデバイスのみ構成を解除できます。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric     connected   configured  unknown
c0::50020f2300006077 disk         connected   configured  unknown
c0::50020f23000063a9 disk         connected   configured  unknown
c1             fc-private    connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708ba7d832 disk         connected   configured  unknown
```

- 3 ファブリックデバイスの構成を解除します。

```
# cfgadm -c unconfigure c0::50020f2300006077
```

- 4 選択されたファブリックデバイスの構成が解除されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric     connected   configured  unknown
c0::50020f2300006077 disk         connected   unconfigured  unknown
c0::50020f23000063a9 disk         connected   configured  unknown
c1             fc-private    connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708ba7d832 disk         connected   configured  unknown
```

## ▼ FC ホストポート上のすべてのFCデバイスの構成を解除する方法

この手順では、ファブリック接続されたホストポートに接続された、構成されているすべてのファブリックデバイスの構成を解除する方法について説明します。

- 1 管理者になります。
- 2 構成を解除するファブリックデバイスを確認します。  
ファブリック接続されたホストポート上のデバイスのみ構成を解除できます。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric     connected   configured  unknown
c0::50020f2300006077 disk         connected   configured  unknown
c0::50020f23000063a9 disk         connected   configured  unknown
c1             fc-private    connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708ba7d832 disk         connected   configured  unknown
```

- 3 選択されたポート上の各ファブリックデバイスに対するすべての活動を停止し、各ファブリックデバイス上のファイルシステムをアンマウントします。  
デバイスがほかのボリュームマネージャーの制御下にある場合、デバイスの構成を解除する前にボリュームマネージャーのドキュメントを参照してください。

```
# cfgadm -c unconfigure c0
```

- 4 選択されたポート上で構成されているすべてのファブリックデバイスの構成を解除します。

---

注 - この操作は、個々のデバイスの unconfigure 操作を、c0 上のすべてのデバイスに対して繰り返します。c0 上のデバイス数が多い場合、この処理は時間がかかることがあります。

---

- 5 c0 上のすべてのデバイスの構成が解除されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric     connected   unconfigured unknown
c0::50020f2300006077 disk         connected   unconfigured unknown
c0::50020f23000063a9 disk         connected   unconfigured unknown
c1             fc-private    connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708ba7d832 disk         connected   configured  unknown
```

c0 と、これに接続されているすべてのファブリックデバイスの Occupant 列が unconfigured と表示されていることに注目してください。

## ▼ マルチパス化された FC デバイスの構成を解除する方法

この手順では、マルチパス化されたデバイスに関連付けられたファブリックデバイスの構成を解除する方法を示すために、ファブリック接続されたホストポート c0 および c2 を示します。

- 1 管理者になります。
- 2 構成を解除するファブリックデバイスのポート WWN を確認します。

```
# cfgadm -al
Ap_Id                Type          Receptacle  Occupant    Condition
c0                   fc-fabric    connected   configured  unknown
c0::50020f2300006077 disk          connected   configured  unknown
c0::50020f23000063a9 disk          connected   configured  unknown
c1                   fc-private   connected   configured  unknown
c1::220203708b69c32b disk          connected   configured  unknown
c1::220203708ba7d832 disk          connected   configured  unknown
c2                   fc-fabric    connected   configured  unknown
c2::50020f2300005f24 disk          connected   configured  unknown
c2::50020f2300006107 disk          connected   configured  unknown
```

この例では、c0::50020f2300006077 および c2::50020f2300006107 の Ap\_Id は、1つのマルチパス化されたデバイスに関連付けられた同一デバイスについての異なるポート WWN を表しています。c0 および c2 ホストポートは、使用が有効にされています。

- 3 選択されたポート上の各ファブリックデバイスに対するすべてのデバイス活動を停止し、各ファブリックデバイス上のファイルシステムをアンマウントします。  
デバイスがほかのボリュームマネージャーの制御下にある場合、ファブリックデバイスを維持するために、ボリュームマネージャーのドキュメントを参照してください。
- 4 デバイスに関連付けられたファブリックデバイスの構成を解除します。  
ファブリック接続されたホストポート上のデバイスのみが、`cfgadm -c unconfigure` コマンドで構成を解除できます。

```
# cfgadm -c unconfigure c0::50020f2300006077 c2::50020f2300006107
```

---

注 - `cfgadm -c unconfigure c0::1111, c1::2222, c3::3333` というコマンド例で示すように、デバイスを最大 8 つのパスから個別に削除できます。代替の方法として、`cfgadm -c unconfigure c0` という例で示すように、パスのセットすべてをホストから削除できます。

---

- 5 選択されたデバイスの構成が解除されていることを確認します。

```
# cfgadm -al
Ap_Id                Type          Receptacle  Occupant    Condition
c0                   fc-fabric    connected   configured  unknown
c0::50020f2300006077 disk          connected   unconfigured unknown
```

```

c0::50020f23000063a9 disk      connected  configured  unknown
c1                fc-private connected  configured  unknown
c1::220203708b69c32b disk    connected  configured  unknown
c1::220203708ba7d832 disk    connected  configured  unknown
c2                fc-fabric  connected  configured  unknown
c2::50020f2300005f24 disk    connected  configured  unknown
c2::50020f2300006107 disk    connected  unconfigured unknown

```

Ap\_Id の c0::50020f2300006077 および c2::50020f2300006107 の構成が解除されていることに注目してください。c0 および c2 の Occupant 列は、引き続きこれらのポートを configured と表示していますが、この理由は、これらのポートがほかの構成済みの占有装置を持つためです。

Ap\_Id c0::50020f2300006077 および c2::50020f2300006107 に関連付けられているマルチパス化されたデバイスは、システムから使用できなくなりました。次の2つのデバイスはシステムから削除されました。

```
/dev/rdsk/c6t60020F20000061073AC8B52D000B74A3d0s2
```

```
/dev/rdsk/c6t60020F20000061073AC8B4C50004ED3Ad0s2
```

## ▼ マルチパス化された FC デバイスへの 1 つのパスの構成を解除する方法

前のセクションの手順に対して、この手順では、c2::50020f2300006107 に関連付けられた1つのデバイスの構成を解除し、ほかのデバイス 50020f2300006077 を構成したままにする方法を示します。ファブリック接続されたホストポート上のデバイスのみが、`cfgadm unconfigure` コマンドで構成を解除できます。

- 1 管理者になります。
- 2 構成を解除するマルチパス化されたデバイスの **Ap\_Id** を確認します。

```

# cfgadm -al
Ap_Id      Type      Receptacle  Occupant  Condition
c0         fc-fabric connected   configured unknown
c0::50020f2300006077 disk      connected   configured unknown
c0::50020f23000063a9 disk      connected   configured unknown
c1         fc-private connected   configured unknown
c1::220203708b69c32b disk    connected   configured unknown
c1::220203708ba7d832 disk    connected   configured unknown
c2         fc-fabric  connected   configured unknown
c2::50020f2300005f24 disk    connected   configured unknown
c2::50020f2300006107 disk    connected   configured unknown

```

この例では、c0::50020f2300006077 および c2::50020f2300006107 の Ap\_Id は、同一デバイスの異なるポート WWN を表しています。

- 3 デバイスに関連付けられた **Ap\_Id** の構成を解除します。

注-Ap\_Idが、デバイスに対して構成された最後のパスを表す場合、パスに対するすべての活動を停止し、パス上のファイルシステムをアンマウントします。マルチパス化デバイスがほかのボリュームマネージャーの制御下にある場合、ファブリックデバイスを維持するために、ボリュームマネージャーのドキュメントを参照してください。

次の例で、c2::50020f2300006107で示すパスは構成が解除され、c0::50020f2300006077は構成されたままになっており、マルチパス化デバイスの複数パスのいずれか1つの構成のみ解除する方法を示しています。

```
# cfgadm -c unconfigure c2::50020f2300006107
```

- 4 選択されたパスc2::50020f2300006107の構成が解除されていることを確認します。

```
# cfgadm-al
Ap_Id                Type      Receptacle  Occupant    Condition
c0                   fc-fabric connected    configured  unknown
c0::50020f2300006077 disk      connected    configured  unknown
c0::50020f23000063a9 disk      connected    configured  unknown
c1                   fc-private connected    configured  unknown
c1::220203708b69c32b disk      connected    configured  unknown
c1::220203708ba7d832 disk      connected    configured  unknown
c2                   fc-fabric connected    configured  unknown
c2::50020f2300005f24 disk      connected    configured  unknown
c2::50020f2300006107 disk      connected    unconfigured unknown
```

そのAp\_Idに関連付けられたデバイスは、c0::50020f2300006077で表現される別のパスを経由して、引き続きシステムから使用できます。1つのデバイスは複数のAp\_Idに接続でき、1つのAp\_Idは複数のデバイスに接続できます。

```
/dev/rdisk/c6t60020F20000061073AC8B52D000B74A3d0s2
and
/dev/rdisk/c6t60020F20000061073AC8B4C50004ED3Ad0s2
```

## ▼ マルチパス化されたすべてのFCデバイスの構成を解除する方法

ファブリック接続されたホストポート上のAp\_Idは、マルチパス化されたデバイスへのパスです。

マルチパス化されたデバイスに対して、複数のAp\_Idが接続されている場合、1つのAp\_Idの構成を解除したあともデバイスは引き続きシステムから使用できます。最後のAp\_Idの構成を解除したあとは、ほかのパスが残っていないため、デバイスはシステムから使用できなくなります。ファブリック接続されたホストポート上のデバイスのみ構成を解除できます。

- 1 管理者になります。

- 2 構成を解除するデバイスを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric    connected   configured  unknown
c0::50020f2300006077 disk         connected   configured  unknown
c0::50020f23000063a9 disk         connected   configured  unknown
c1             fc-private   connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708b9b20b2 disk         connected   configured  unknown
c2             fc-fabric    connected   configured  unknown
c2::50020f2300005f24 disk         connected   configured  unknown
```

- 3 選択されたポート上で構成されているすべてのデバイスの構成を解除します。

```
# cfigadm -c unconfigure c2
```

---

注- この操作は、個々のデバイスの unconfigure コマンドを、c2 上のすべてのデバイスに対して繰り返します。c2 上のデバイス数が多い場合、この処理は時間がかかることがあります。

---

- 4 c2 上のすべてのデバイスの構成が解除されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             fc-fabric    connected   configured  unknown
c0::50020f2300006077 disk         connected   configured  unknown
c1             fc-private   connected   configured  unknown
c1::220203708b69c32b disk         connected   configured  unknown
c1::220203708ba7d832 disk         connected   configured  unknown
c2             fc-fabric    connected   unconfigured unknown
c2::50020f2300005f24 disk         connected   unconfigured unknown
c2::50020f2300006107 disk         connected   unconfigured unknown
```

Occupant 列では、c2 と、c2 に接続されたすべてのデバイスが unconfigured と表示されていることに注目してください。

## サポートされる FC-HBA API

---

この付録には、サポートされる FC-HBA インタフェースのリストが含まれています。API に関する追加の情報は、15 ページの「Solaris I/O マルチパス化の概要」を参照してください。

### サポートされるファイバチャネル HBA API

Solaris I/O マルチパス化では、次の FC-HBA インタフェースがサポートされています。

- HBA\_GetVersion
- HBA\_LoadLibrary
- HBA\_FreeLibrary
- HBA\_GetNumberOfAdapters
- HBA\_GetAdapterName
- HBA\_OpenAdapter
- HBA\_CloseAdapter
- HBA\_GetAdapterAttributes
- HBA\_GetAdapterPortAttributes
- HBA\_GetDiscoveredPortAttributes
- HBA\_GetPortAttributesbyWWN
- HBA\_SendCTPassThru
- HBA\_SendCTPassThruV2
- HBA\_RefreshInformation
- HBA\_GetFcpTargetMapping
- HBA\_SendScsiInquiry
- HBA\_SendReportLuns
- HBA\_SendReadCapacity
- HBA\_SetRNIDMgmtInfo
- HBA\_GetRNIDMgmtInfo
- HBA\_SendRNID

- HBA\_SendRNIDV2
- HBA\_ScsiInquiryV2
- HBA\_ScsiReportLUNsv2
- HBA\_ScsiReadCapacityV2
- HBA\_OpenAdapterByWWN
- HBA\_RefreshAdapterConfiguration
- HBA\_GetVendorLibraryAttributes
- HBA\_GetFcpTargetMappingV2
- HBA\_SendRRLS
- HBA\_RegisterForAdapterEvents
- HBA\_RegisterForAdapterAddEvents
- HBA\_RegisterForAdapterPortEvents
- HBA\_RegisterForTargetEvents
- HBA\_RegisterForAdapterTargetEvents

## マルチパス化されたデバイスの問題のトラブルシューティング

---

この付録では、マルチパス化機能の実行中に発生する可能性がある問題の解決方法を提供します。

次の項目について説明します。

- 143 ページの「`stmsboot` の実行中にシステムがブートに失敗する」
- 144 ページの「`stmsboot` の実行中にシステムがクラッシュする」

### `stmsboot` の実行中にシステムがブートに失敗する

`stmsboot` の有効化 (-e)、無効化 (-d)、または更新 (-u) 操作の後にシステムがブートに失敗した場合、シングルユーザーモードで回復するための次のステップを実行します。

#### ▼ シングルユーザーモードでブートの失敗から回復する方法

- 1 `root` ユーザーパスワードを入力してシングルユーザーモードを開始します。
- 2 `mpxio-upgrade` サービスを再起動します。

```
# svcadm restart svc:/system/device/mpxio-upgrade:default
```

このコマンドが成功しない場合、次のセクションにある指示に従って元の構成を回復します。

## stmsbootの実行中にシステムがクラッシュする

stmsbootの有効化(-e)、無効化(-d)、または更新(-u)操作の後、システムにハングアップ、パニック、またはブートの失敗が発生した場合、元のマルチパス化構成を回復するための次のステップを実行します。

### ▼ システムクラッシュから回復する方法

- 1 別のブート可能ディスクかインストールDVDを使用するか、またはネットワーク経由でシステムをブートします。

- SPARC: インストールメディアまたはネットワーク上のインストールサーバーからブートする場合、テキストインストールを選択します。インストールサーバーからブートする場合、次のコマンドを使用します。

```
ok boot net:dhcp
```

- x86: インストールメディアまたはネットワーク上のインストールサーバーからブートする場合、GRUBメニューから次のテキストインストールオプションを選択します。

```
Oracle Solaris 11.1 Text Installer and command line
```

- 次のメニューからオプション 3 Shell を選択します。

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun-color)
5 Reboot
```

```
Please enter a number [1]: 3
To return to the main menu, exit the shell
```

- 2 ZFS ルートプールをインポートします。

```
# zpool import -f rpool
```

- 3 ルートBEをマウントします。

```
# mkdir /a
# beadm mount solaris /a
```

- 4 元の `fp.conf` ファイル(FCマルチパス化の場合)または `mpt.conf` (SASマルチパス化の場合)を次のようにして復元します。

- `stmsboot -e` コマンドまたは `stmsboot -d` コマンドを実行した場合、次のようにします。

– SASマルチパス化の場合:

```
# cp /a/etc/mpxio/mpt.conf /a/etc/driver/drv/mpt.conf
```

- FCマルチパス化の場合:

```
# cp /a/etc/mpxio/fp.conf /a/etc/driver/drv/fp.conf
```

- **stmsboot -u** コマンドを実行して、**fp.conf** ファイルまたは **mpt.conf** を変更した場合、**/a/etc/driver/drv/fp.conf** または **/a/etc/driver/drv/mpt.conf** ファイルのいずれかを編集することによって、このファイルに対して実行した変更を元に戻します。

- 5 **stmsboot** コマンドを実行する前にマルチパス化構成に対して実行したほかの変更を元に戻します。

たとえば、**scsi\_vhci.conf** ファイルを変更した場合、**/a/etc/driver/drv/scsi\_vhci.conf** ファイルを編集することによって、このファイルに対して実行した変更を元に戻します。

ストレージレイのデバイス設定を変更した場合、元の設定を復元します。

- 6 元の **/etc/vfstab** ファイルを復元します。

```
# cp /a/etc/mpxio/vfstab /a/etc/vfstab
```

**/a/etc/mpxio/vfstab** ファイルは **stmsboot** コマンドが **vfstab** ファイルを更新する前に保存した、元の **/etc/vfstab** ファイルのコピーです。 **stmsboot** コマンドが **vfstab** ファイルを変更しなかった場合、**/a/etc/mpxio/vfstab** ファイルは存在しません。

- 7 システムが **x86** ベースシステムの **Oracle Solaris OS** 上で実行中の場合、次のステップを実行します。

- a. 元の **/boot/solaris/bootenv.rc** ファイルを復元します。

```
# cp /a/etc/mpxio/bootenv.rc /a/boot/solaris/bootenv.rc
```

**/a/etc/mpxio/bootenv.rc** ファイルは、**bootenv.rc** ファイルを更新する前に **stmsboot** コマンドが保存した元の **/boot/solaris/bootenv.rc** ファイルのコピーです。 **stmsboot** コマンドが **bootenv.rc** ファイルを変更しなかった場合、**/a/etc/mpxio/bootenv.rc** ファイルは存在しません。

- b. ブートアーカイブを更新します。

```
# bootadm update-archive -R /a
```

- 8 **mpxio-upgrade** サービスを無効にします。

```
# /usr/sbin/svccfg -f /a/etc/mpxio/svccfg_recover
```

- 9 **BE** をアンマウントします。

```
# beadm umount solaris
```

- 10 システムをリブートします。



# 索引

---

## 数字・記号

24 ビット FC アドレス指定デバイス, 126

## A

ANSI 規格マルチパス管理 API, 39

Ap\_Id, 127

## C

cfgadm

-c, 129

-l, 127, 128

cfgadm -al -o show\_SCSI\_LUN, 127

cfgadm -c unconfigure, 137

cfgadm(1M), 128

cfgadm configure, 128

cfgadm\_fp(1M), 131

## F

fc-fabric, 132

FCAL, 16

FCoE

制限事項, 99–100

ポートの構成, 100–102

ポートの削除, 101

ポートの作成, 100–101

ポートのステータスのレポート, 101

## I

Internet Protocol over Fibre Channel, 「IPFC」を参照  
IPFC, 考慮事項, 107–110

iSCSI

iSCSI 構成に関する問題のトラブル

シューティング (方法), 86

iSCSI 構成の監視 (方法), 78

iSCSI ディスクへのアクセス (方法), 66

iSCSI のイニシエータとターゲットのパラ  
メータの変更 (方法), 83

一般的な iSCSI エラーメッセージ, 88

静的および動的ターゲット発見, 63

単方向または双方向 CHAP 認証の構成 (方  
法), 69

発見された iSCSI ターゲットの削除 (方法), 67

iscsiadm list, iSCSI 構成情報の表示 (例), 78

iscsiadm modify コマンド

CHAP を有効にする (例), 70

静的または動的ターゲットの有効化または無効  
化 (例), 67

iscsiadm remove コマンド, 静的または動的  
ターゲットの削除 (例), 67

## L

LUN

SCSI デバイス, 134

認識, 57

マスク, 55

luxadm(1M), 36

**M**

modinfo, 127  
mpathadm コマンド, 39  
  イニシエータポート  
  プロパティ, 40-41  
  自動フェイルバック、構成, 46  
  自動フェイルバックの構成, 46  
  特定の名前を持つ論理ユニット, 44  
  パスを無効にする, 52-53  
  パスを有効にする, 50-52  
  論理ユニット  
    ターゲットポートに関連付けられた, 42  
    フェイルオーバー, 47-50  
    プロパティおよび関連したパス, 41  
mpt ドライバ, 105-106

**N**

NPIV ポート、構成, 95-98

**R**

REPORT LUNS コマンド, 32

**S**

SAS デバイス  
  構成, 106  
  動的発見, 105-106  
  マルチパス化の考慮事項, 105  
sd ドライバ, 105-106  
Serial Attached SCSI, 「SAS」を参照  
show\_SCSI\_LUN, 129, 134  
snoop(1M), 107  
ssd driver, 127  
st driver, 127  
stmsboot コマンド, 26  
  FC デバイスの考慮事項, 56  
  考慮事項, 22  
  システムのクラッシュ, 144  
  デバイスパスの判別, 26  
  ブートの考慮事項, 57

stmsboot コマンド (続き)

  マルチパス化の有効化および無効化, 20

**T**

T10 規格, 39  
T11 標準, 16

**X**

x86 ベースシステムの OS ブート手順, ファイバ  
  チャンネル, 113-120

**あ**

アクセス, iSCSI ディスク (方法), 66

**い**

イニシエータポート, プロパティの表示, 40-41

**え**

エラーメッセージ, iSCSI, 88

**か**

監視, iSCSI 構成 (方法), 78

**こ****構成**

  FCoE, 100-102  
  FC デバイス, 55-57  
  iSCSI の単方向または双方向 CHAP 認証 (方  
    法), 69  
  NPIV ポート, 95-98  
  SAS デバイス, 106

## 構成 (続き)

- 仮想ファイバチャネルポート, 95-98
- サードパーティー製デバイス, 32, 33
- 自動フェイルバック, 36-37
- 手動, 125-126
- タスク, 21
- ダンプ, 26
- デバイス, 127
- ファイバチャネルの概要, 21-23
- ファブリック接続されたデバイス, 55-57
- ファブリックデバイス, 128
- マルチパス化, 26
- マルチパス化機能, 25, 55-57

## 考慮事項

- IPFC, 107-110
- SAN デバイス, 55-56
- SAS デバイス, 105
- stmsboot コマンド, 22
- StorageTek Traffic Manager, 22-23
- サードパーティー製デバイスの構成, 32
- デバイス固有, 26
- デバイス名の変更, 26
- ファブリックブートデバイス, 57
- ポート単位の構成, 30-32
- マルチパス化, 26

## さ

- 削除, 発見された iSCSI ターゲット (方法), 67

## し

- 自動フェイルバック, 構成, 36-37

## す

- ストレージデバイス, 36

## せ

- 制限事項, FCoE, 99-100

## て

- テープデバイスの永続的なバインド, 121
- デバイス
  - 構成, 128
  - ストレージ, 19
  - 非対称, 19
- デバイス、永続的なネーミング, テープ, 16
- デバイス管理, 20

## と

- トラブルシューティング
  - iSCSI 構成に関する問題 (方法), 86
  - マルチパス化, 143-145

## は

## 発見

- SAS デバイス, 105-106
- 動的, 105-106

## ふ

## ファイバチャネル

- HBA, 16
- IPFC SAN デバイスの構成, 107-112
- IPFC の考慮事項, 107-110
- LUN レベル情報の一覧表示, 127-128
- x86 ベースシステムのブート, 113-120
- アダプタのポートインスタンスの判別, 108-110
- 構成の概要, 21-23
- サポートされる HBA インタフェース, 141-142
- 調停ループ, 16
- マルチパス化構成, 25-26
- マルチパス化の有効化と無効化, 26-30
- ファイバチャネルポート, 構成, 95-98
- ファブリック接続されたホストポート, 128, 132
- ファブリックデバイス構成, 128
- ブート, 16
- ブートディスク, 57
- 物理デバイス, 129

ブロードキャスト, 107  
プロミスキャス (promiscuous) モード, 107

む  
無効化, マルチパス化, 26-30

へ  
変更, iSCSI のイニシエータとターゲットのパラ  
メータ (方法), 83

ゆ  
有効化, マルチパス化, 26-30

ほ  
ポート単位の構成  
考慮事項, 30-32  
マルチパス化の有効化と無効化, 30-32

ま  
マルチパス化  
mpathadm コマンド, 39  
SAS 考慮事項, 105  
SAS ブートデバイスの構成, 106  
機能, 18  
サードパーティー製ストレージデバイスの構  
成, 32  
サポート, 39  
自動フェイルバックの構成, 36-37  
手動構成, 125-126  
デバイスノードの構成, 131-135  
トラブルシューティング, 143-145  
ファブリックデバイスノードの構成, 126  
ブートの考慮事項, 57  
ポート単位ベースでの有効化または無効  
化, 30-32  
無効化、複数デバイスの構成, 130-131  
無効、デバイスノードの構成, 128  
有効化と無効化, 26-30  
マルチパス化、有効化と無効化, 26-30  
マルチパス管理 API プラグイン  
プロパティ, 39  
ライブラリ, 39