

Oracle® Solaris 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際、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	7
1 Solaris I/O マルチパス化の概要	11
Solaris I/O マルチパス化の新機能	11
Solaris I/O マルチパス化の概要	13
ファイバチャネルソフトウェアの機能	14
SAS ソフトウェア機能	15
Solaris I/O マルチパス化機能	15
サポートされるデバイス標準規格	17
2 ファイバチャネルマルチパス化構成の概要	19
FC デバイス構成タスクの概要	19
Solaris I/O マルチパス化構成の考慮事項	20
3 Solaris I/O マルチパス化機能の構成	23
マルチパス化 I/O 機能の構成	23
マルチパス化の考慮事項	24
マルチパス化の有効化と無効化	25
▼ マルチパス化を有効にする方法	25
▼ マルチパス化を無効にする方法	26
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	27
ポート構成の考慮事項	28
サードパーティー製ストレージデバイスの構成	30
サードパーティー製デバイスの構成の考慮事項	30
サードパーティー製ストレージデバイスの構成: 新規デバイス	31
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	32
デバイス名の変更の表示	33

自動フェイルバックの構成	34
▼ 自動フェイルバックの構成方法	34
4 マルチパス化デバイスの管理 (mpathadm)	35
マルチパス化デバイスの管理	35
▼ マルチパス化サポート情報を表示する方法	35
▼ 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	36
▼ 特定の LUN 情報を表示する方法	37
▼ 特定のターゲットポートに関連付けられたすべての LUN を表示する方法	38
▼ 特定の名前を持つ LUN を表示する方法	40
▼ 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	42
▼ LUN をフェイルオーバーする方法	43
▼ LUN パスを有効にする方法	46
▼ LUN パスを無効にする方法	47
5 SAN デバイスの構成	51
SAN デバイスの考慮事項	51
SAN デバイスの追加	52
▼ SAN デバイスを追加する方法	52
SPARC でのファブリックブートデバイスの構成	53
ファブリックブートデバイスの考慮事項	53
6 仮想ファイバチャネルポートの構成	55
NPIV の概要	55
NPIV の制限事項	55
NPIV ポートの作成	56
▼ NPIV ポートの作成方法	56
▼ NPIV ポートの削除方法	57
▼ NPIV ポートステータスの表示方法	57
7 FCoE ポートの構成	59
FCoE の概要	59
FCoE の制限事項	59
FCoE ポートの構成	60

▼FCoE ポートの作成方法	60
▼FCoE ポートの削除方法	61
▼FCoE ポートステータスの表示方法	61
▼FCoE ポートの再初期化を強制実行する方法	62
FCoE ハードウェアオフロードの構成	63
8 SAS ドメインの構成	65
SAS マルチパス化の考慮事項	65
SAS デバイスの動的発見	65
SAS ブートデバイスの構成	66
9 IPFC SAN デバイスの構成	67
IPFC の考慮事項	67
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	68
IPFC の呼び出しと構成	70
▼ネットワークインタフェースを手動で開始する方法	71
▼自動ネットワーク構成用にホストを構成する方法	71
10 x86 システムのファイバチャネルデバイスからの Solaris OS のブート	73
Oracle Solaris OS のセットアップ要件	74
Oracle Solaris OS のインストールの概要	74
Oracle Solaris OS のインストール手続き	75
▼Oracle Solaris OS のインストール方法	75
▼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	75
11 テープデバイス用の持続的なバインド	81
持続的なバインドの概要	81
テープリンクの作成	82
▼テープデバイスリンクの作成方法	83
A ファブリック接続されたデバイスの手動構成	85
FC デバイスの手動構成	85
▼FC デバイスを手動構成する方法	86
ファブリックデバイスノードの構成	86

LUN レベル情報が表示できることの確認	87
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	88
▼ 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	89
▼ 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	90
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	91
▼ マルチパス化された個々の FC デバイスを構成する方法	92
▼ マルチパス化された複数の FC デバイスを構成する方法	93
ファブリックデバイスの構成の解除	95
ファブリックデバイスの構成の解除	95
B サポートされる FC-HBA API	101
サポートされるファイバチャネル HBA API	101
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	105
stmsboot の実行中にシステムがブートに失敗する	105
▼ 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	105
stmsboot の実行中にシステムがクラッシュする	106
▼ システムクラッシュから回復する方法	106
索引	109

はじめに

Oracle Solaris 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 OS 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方法についての説明も含まれています。

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。SANS および SAS ドメインの管理および保守について高いレベルの専門知識を持っていることが前提です。

お読みになる前に

本書を読む前に、最新の Oracle Solaris 11 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」	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」	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 (mpathadm)」	mpathadm コマンドを使用して、ANSI 標準のマルチパス管理 API を経由してマルチパス化管理を有効にする方法について説明します。
第 5 章「SAN デバイスの構成」	SAN 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。

章	説明
第 6 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に使用するステップについて説明します。
第 7 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 ポートの構成に使用されるステップについて説明します。
第 8 章「SAS ドメインの構成」	SAS ドメイン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。
第 9 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 10 章「x86 システムのファイバチャネルデバイスからの 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 11 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 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

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.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。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
AaBbCc123	変数を示します。実際に使用する特定の名前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。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第5章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep '^#define \ XV_VERSION_STRING'

Oracle Solaris OS に含まれるシェルで使用する、UNIX のデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solaris のリリースによって異なります。

- C シェル

```
machine_name% command y|n [filename]
```

- C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

- Bash シェル、Korn シェル、および Bourne シェル

```
$ command y|n [filename]
```

- Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー

```
# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

|は区切り文字(セパレータ)です。この文字で分割されている引数のうち1つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します(例:Shift キーを押します)。ただし、キーボードによってはEnter キーがReturn キーの動作をします。

ダッシュ(-)は2つのキーを同時に押すことを示します。たとえば、Ctrl-DはControl キーを押したままD キーを押すことを意味します。

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：

- 11 ページの「Solaris I/O マルチパス化の新機能」
- 13 ページの「Solaris I/O マルチパス化の概要」
- 17 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

- マルチパス化パッケージの変更 – Oracle Solaris 11 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、25 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 11 リリース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 6 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 7 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、[62 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` 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ン、受容体、および占有装置情報を表示する `croinfo` 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:

```
# croinfo -O occupant-compdev
c0t5000C50010420FEBd0
c0t5000C5000940F54Fd0
c0t5000C50007DD498Fd0
c0t5000C50002FB622Fd0
c0t5000C500103C9033d0
c0t5000C5000940F733d0
```

- 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。

- [14 ページの「ファイバチャネルソフトウェアの機能」](#)
- [15 ページの「SAS ソフトウェア機能」](#)
- [15 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、ファイバチャネルおよび SAS ストレージの両方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 の管理: デバイスとファイルシステム](#)』の第 14 章「[COMSTAR を使用したストレージデバイスの構成](#)」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。
共通の FC-HBA API のマニュアルページは `libhbaapi(3LIB)` マニュアルページセクションに含まれています。FC 仕様の追加情報については、<http://www.t11.org> を参照してください。
- 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 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。

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、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 以外のストレージについては、サポートを有効にし、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、ディスクドライバは 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 および x86 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 または x86 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 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 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す。

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、ファイバチャネル (FC) デバイスおよび Solaris I/O マルチパス化機能の構成について説明します。次の内容について説明します。

- 19 ページの「FC デバイス構成タスクの概要」
- 20 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、および各タスクを実行するための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にされます。 ■ 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 ■ SPARC プラットフォーム上では手動構成による ■ SAS デバイスについては手動構成による	マルチパス化を有効にするステップの概要は、第 3 章「 Solaris I/O マルチパス化機能の構成 」に記載されています。
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す。	第 5 章「 SAN デバイスの構成 」

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
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 6 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。	51 ページの「SAN 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 9 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。	

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` およびダンプ構成を自動的に更新できます。このソフトウェアは、SPARC デバイスについてはデフォルトで無効で、x86 デバイスについてはデフォルトで有効です。
- 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。

次の内容について説明します。

- 23 ページの「マルチパス化 I/O 機能の構成」
- 24 ページの「マルチパス化の考慮事項」
- 25 ページの「マルチパス化の有効化と無効化」
- 27 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」
- 30 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」
- 34 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は、サポートされるすべての FC HBA を制御するために構成できます。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、次の各セクションで説明する `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。
- eeprom ブートデバイスが、現在のブートデバイスからブートするように設定されていることを確認します。このことが必要なのは、stmsboot コマンドが操作を完了するのにマシンをリブートするためです。
- 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、[27 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、[24 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 mpxio status for driver fp
Checking mpxio 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、[24 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、[27 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 mpxio status for driver fp
Checking mpxio 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、[24 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。

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、マルチパス操作用に有効化されます。

次の手順は SPARC ベースおよび x86 ベースの両方のマシンに適用されます。

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。つまり、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

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。
たとえば、デバイスを選択するには、`/dev/cfg` ディレクトリに対して `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。

1 つまたは複数のポートを決定したら、次のステップに移動して、マルチパス化を明示的に有効または無効にします。

- 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。
 - 有効にする各 FC HBA ポートについて、次の行を /etc/driver/drv/fp.conf に追加します。

```
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 mpxio-disable="yes";
```

例:

```
name="fp" parent="/pci@6,2000/SUNW,qlc@2" port=0 mpxio-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、[24 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、次のことに注意してください。

- 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。これが自動的に実行されるデフォルトの方法は、`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す。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ストレージデバイスとして定義されます。

▼ サードパーティー製デバイスの構成方法

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「,」文字で区切られ、最後のベンダー/製品エントリは「;」文字で終了します。

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。

「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には、次を追加します。

```
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、[25 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、INQUIRY データからの特定のベンダー 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 `/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", "NONE",
"VendorID2ProductID2", "NONE",
...
"VendorIDnProductIDn", "NONE";
```

前の例にあるエントリは「,」文字で区切られ、最後のベンダー/製品エントリは「;」文字で終了します。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、[25 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ブートする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の間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/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
```

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` ファイルを編集します。
- 4 **auto-failback** エントリを次のように変更して、自動フェイルバック機能を無効にします。

```
auto-failback="disable";
```


auto-failback エントリを次のように変更して、自動フェイルバック機能を有効にします (以前無効にした場合)。

```
auto-failback="enable";
```
- 5 ファイルを保存して終了します。
- 6 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 (mpathadm)

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。この章で、パス、イニシエータポート、ターゲットポート、および LUN を指すために使用されている用語は、T10 仕様と一貫性が保たれています。

マルチパス化デバイスの管理

任意の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を表示する方法

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を使用して識別されます。

- 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。
- 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 list mpath-support
mpath-support: libmpscsi_vhci.so

# 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` 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特定します。

```

# 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 LUN 情報を表示する方法

mpathadm コマンドを使用して、LUN のリストと、各 LUN のプロパティーを表示します。表示される LUN のリストには、特定の LUN のプロパティー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 LUN のリストを表示します。

```
# mpathadm list lu
/dev/rdsd/c0t600144F08069703400004E828EE10004d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdsd/c0t600144F08069703400004E8183DF0002d0s2
    Total Path Count: 8
    Operational Path Count: 8
/dev/rdsd/c0t600A0B800026D63A0000A4994E2342D4d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsd/c0t600A0B800029065C00007CF54E234013d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsd/c0t600A0B800026D63A0000A4984E234298d0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsd/c0t600A0B800029065C00007CF44E233FCFd0s2
    Total Path Count: 4
    Operational Path Count: 4
/dev/rdsd/c0t600A0B800026D63A0000A4974E23424Ed0s2
    Total Path Count: 4
    Operational Path Count: 4
.
.
.
```

- 2 特定の LUN についての構成情報を表示します。

```
# mpathadm show lu /dev/rdsd/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 LUN 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 LUN の一覧を表示します。

```

# 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

```

2 特定の LUN 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/rdsd/c0t600A0B800029065C00007CF54E234013d0s2
```

```
Logical Unit: /dev/rdsd/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/rdsd/c0t600A0B800026D63A0000A4994E2342D4d0s2
                Total Path Count: 4
                Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
                /dev/rdsd/c0t600A0B800029065C00007CF54E234013d0s2
                Total Path Count: 4
                Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
                /dev/rdsd/c0t600A0B800026D63A0000A4984E234298d0s2
                Total Path Count: 4
                Operational Path Count: 4
mpath-support: libmpscsi_vhci.so
                /dev/rdsd/c0t600A0B800029065C00007CF44E233FCFd0s2
                Total Path Count: 4
                Operational Path Count: 4
.
.
.
```

▼ 特定の名前を持つ LUN を表示する方法

マルチパス化された LUN 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 LUN 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 LUN のリストを表示します。

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


2 選択された LUN の構成情報を表示します。

```
# mpathadm show lu /dev/rdsd/c0t600A0B800026D63A0000A4994E2342D4d0s2
Logical Unit: /dev/rdsd/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 LUN 情報を表示します。

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

▼ 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

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。

- 1 管理者になります。
- 2 サポートされるマルチパスドライバ情報を表示します。
- 3 サポートされるマルチパ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フェイルバック表示は、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。[34 ページ](#)の「自動フェイルバックの構成」を参照してください。

▼ LUN 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 LUN 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 LUN 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
.
.
.

```

2 選択された LUN の構成情報を表示します。

```

# 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: 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 LUN 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

```

▼ LUN パスを有効にする方法

LUN 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 LUN 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 LUN のリストを表示します。

```

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

```

- 2 選択された LUN の構成情報を表示します。

```

# mpathadm show lu
Logical Unit: /dev/rdsd/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 選択された LUN パスを有効にします。

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

▼ LUN パスを無効にする方法

この操作は LUN のパスの動作状態にかかわらず、LUN 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 LUN 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 選択された LUN 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 LUN パスを無効にします。

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


SAN デバイスの構成

この章では、SAN 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。

次の内容について説明します。

- [51 ページの「SAN デバイスの考慮事項」](#)
- [52 ページの「SAN デバイスの追加」](#)
- [53 ページの「SPARC でのファブリックブートデバイスの構成」](#)

SAN 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。
- 手動のデバイス構成を選択しないかぎり、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 およびダンプ構成を自動的に更新できます。このソフトウェアは、SPARC デバイスについてはデフォルトで無効で、x86 デバイスについてはデフォルトで有効です。

SAN デバイスの追加

SAN デバイスを追加および削除するには、次のコマンドについての知識が必要です。

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。	cfgadm(1M) および cfgadm_fp(1M)
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。	format(1M)
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。	luxadm(1M)

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。

▼ SAN デバイスを追加する方法

- 1 SAN デバイスに目的の LUN を作成します。
- 2 必要な場合、SAN 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、SAN 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

Solaris I/O マルチパス化機能によって、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す。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の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。

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。

- 55 ページの「NPIV の概要」
- 55 ページの「NPIV の制限事項」
- 56 ページの「NPIV ポートの作成」

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

次のリストは、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 の制限事項を示したものです。

- 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ます。これは 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、これ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 ポートの作成

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といった、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。

```
# fcdm 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 6-1 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 6-2 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 6-3 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。

次の内容について説明します。

- 59 ページの「FCoE の概要」
- 59 ページの「FCoE の制限事項」
- 60 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純粋な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` コマンドが使用できます。このコマンドは、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 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、リブ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
```

このコマンドは、システム内のすべての FCoE ポートについての FC 固有情報のリストを取得します。

```
# 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
```

このコマンドは、システム内のすべての FCoE ポートについての FCoE 固有情報のリストを取得します。

▼ 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

次に示す4つのチューニング可能な設定は、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ために使用できます。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` ファイルに設定できます。

`/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 7-1 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 ブートデバイスの構成の概要を説明します。

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、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` コマンドから表示でき、利用可能になります。

SAS ブートデバイスの構成

Solaris マルチパス化 I/O 機能によって、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

ホストシステム上での IPFC (Internet Protocol over Fibre Channel) の構成では、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。

この章の内容は、次のとおりです。

- 67 ページの「IPFC の考慮事項」
- 70 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 9-1 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サポートの最大数	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 11 リリースでは、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。ここには、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、68 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。

たとえば、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,qlc@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,qlc@1/fp@0,0 /etc/path_to_inst
"/pci@8,600000/SUNW,qlc@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,qlc@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。後続のリブートで 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、リブート時に自動 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 システムのファイバチャネルデバイスからの Solaris OS のブート

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。

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。

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 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 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 は、DVD-ROM またはネットワークから x86 ベースのシステムにインストールできます。

この章の内容は、次のとおりです。

- 74 ページの「Oracle Solaris OS のセットアップ要件」
- 74 ページの「Oracle Solaris OS のインストールの概要」
- 75 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 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

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 番号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 番号を使用して、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 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 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 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 11/11 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す。

- 4 テキストインストール–目的のアレイとア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 10-1 を参照してください。

```
# luxadm display /dev/rdisk/c0t600015D0002028000000000000001142d0s2
DEVICE PROPERTIES for disk: /dev/rdisk/c0t600015D0002028000000000000001142d0s2
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/c0t600015D0002028000000000000001142d0s2
/devices/scsi_vhci/disk@g600015d0002028000000000000001142: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 10-1 luxadm display コマンドおよび出力

```

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# = c0t600015d00020280000000000000001142d0
- 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 10-2 および図 10-3 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 10-2 HBA WWN についての HBA BIOS 画面

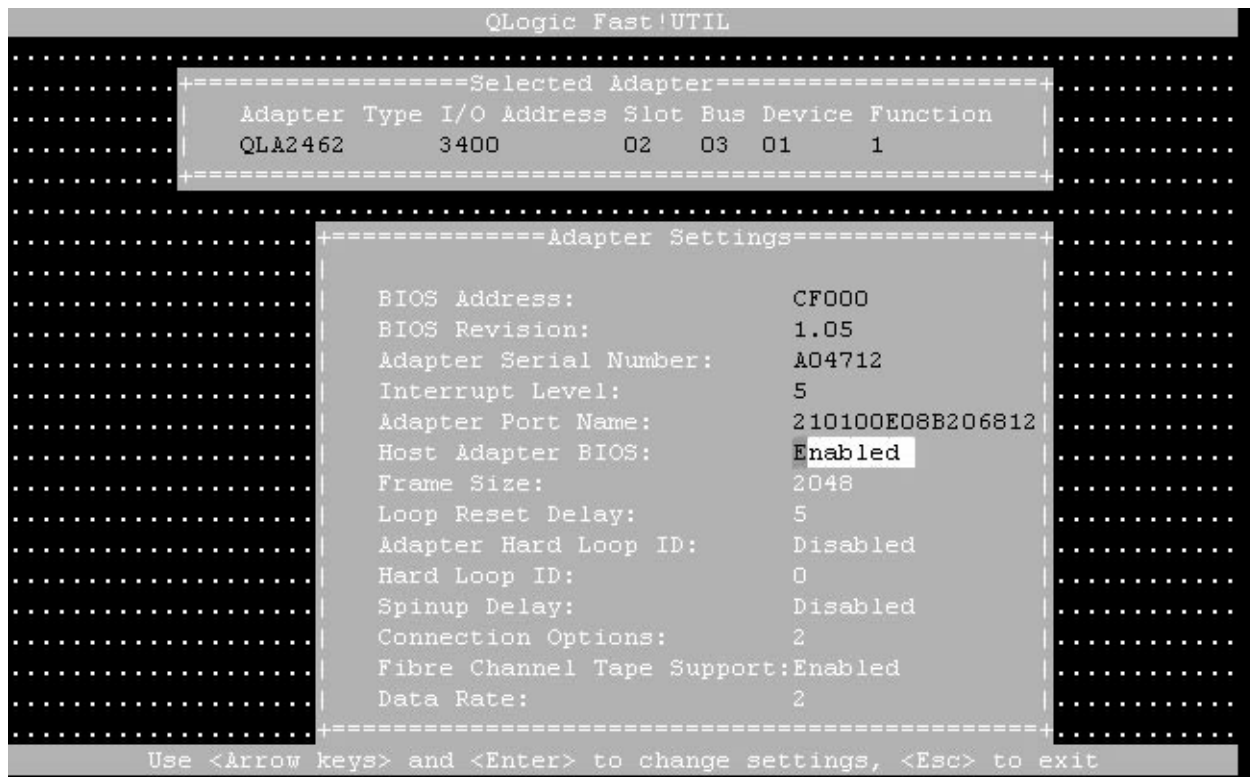
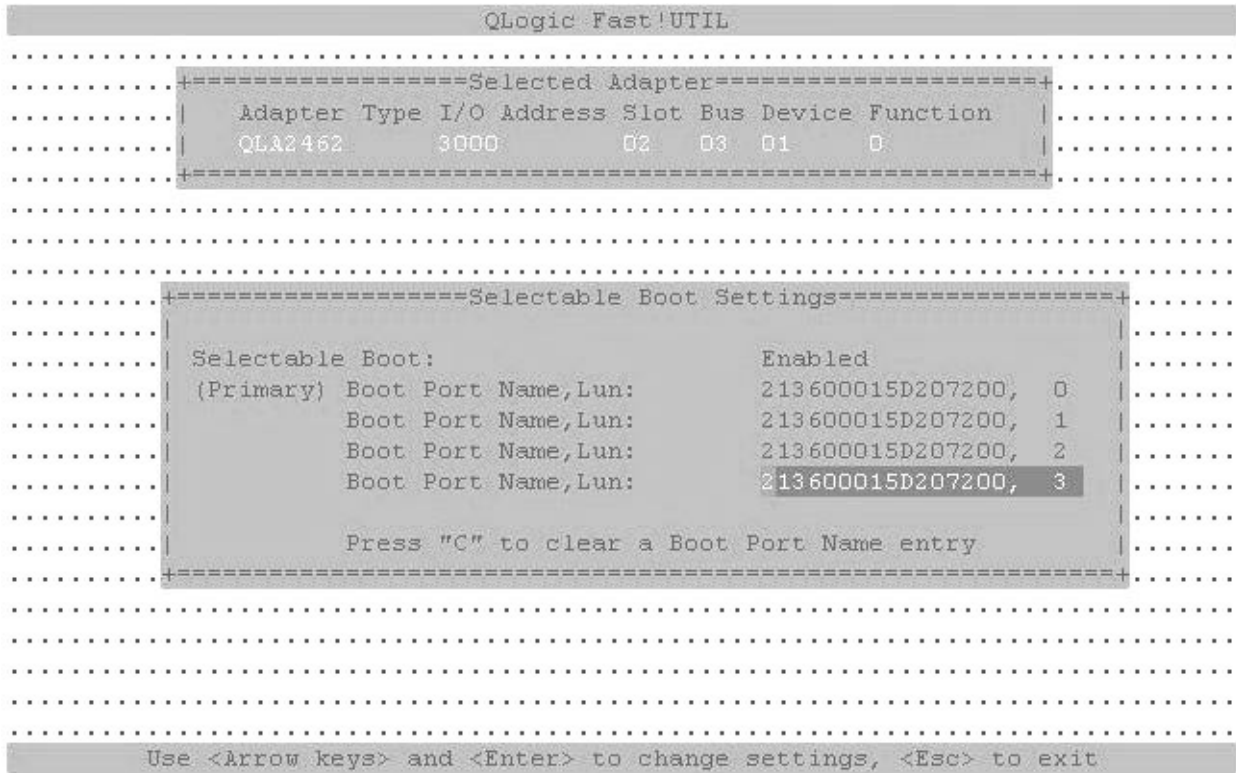


図 10-3 FCLUN からのブートを有効にするための HBA BIOS 画面



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 を実行します。

テープデバイス用の持続的なバインド

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。

持続的なバインドの概要

`/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` 内で次を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 (PWWN) 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 が実行されるとき、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=PWWN,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 PWWN 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 82 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。

この付録は、次のセクションから構成されています。

- 85 ページの「FC デバイスの手動構成」
- 86 ページの「ファブリックデバイスノードの構成」
- 88 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」
- 91 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」
- 95 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。
 - 88 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」
 - 91 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 構成するデバイを確認します。

```
# cfmadm -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。

```
# cfmadm -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。

```
# cfmadm -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。

```
# cfmadm -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

87 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え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、87 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ときのみ作成されます。

- 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/rdisk/c6t60020F20000061073AC8B52D000B74A3d0s2
```

```
/dev/rdisk/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。

```
# cfmadm -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。

```
# cfmadm-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。

```
# cfgadm -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 インタフェースとサポートされない 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、[13 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

表 B-1 サポートされる FC-HBA インタフェースとサポートされない FC-HBA インタフェース

FC-HBA インタフェース	Solaris I/O マルチパス化のサポート
HBA_GetVersion	サポートされています
HBA_LoadLibrary	サポートされています
HBA_FreeLibrary	サポートされています
HBA_GetNumberOfAdapters	サポートされています
HBA_GetAdapterName	サポートされています
HBA_OpenAdapter	サポートされています
HBA_CloseAdapter	サポートされています
HBA_GetAdapterAttributes	サポートされています
HBA_GetAdapterPortAttributes	サポートされています
HBA_GetDiscoveredPortAttributes	サポートされています
HBA_GetPortAttributesbyWWN	サポートされています
HBA_SendCTPassThru	サポートされています
HBA_SendCTPassThruV2	サポートされています

表 B-1 サポートされる FC-HBA インタフェースとサポートされない FC-HBA インタフェース (続き)

FC-HBA インタフェース	Solaris I/O マルチパス化のサポート
HBA_RefreshInformation	サポートされています
HBA_GetFcpTargetMapping	サポートされています
HBA_SendScsiInquiry	サポートされています
HBA_SendReportLuns	サポートされています
HBA_SendReadCapacity	サポートされています
HBA_GetPortStatistics	サポートされています
HBA_ResetStatistics	サポートされていません
HBA_GetFcpPersistentBinding	サポートされていません
HBA_GetEventBuffer	サポートされていません
HBA_SetRNIDMgmtInfo	サポートされています
HBA_GetRNIDMgmtInfo	サポートされています
HBA_SendRNID	サポートされています
HBA_SendRNIDV2	サポートされています
HBA_ScsiInquiryV2	サポートされています
HBA_ScsiReportLUNsV2	サポートされています
HBA_ScsiReadCapacityV2	サポートされています
HBA_OpenAdapterByWWN	サポートされています
HBA_RefreshAdapterConfiguration	サポートされています
HBA_GetVendorLibraryAttributes	サポートされています
HBA_GetFcpTargetMappingV2	サポートされています
HBA_SendRPL	サポートされていません
HBA_SendRPS	サポートされていません
HBA_SendSRL	サポートされていません
HBA_SendLIRR	サポートされていません
HBA_SendRLS	サポートされています
HBA_RemoveCallback	サポートされています
HBA_RegisterForAdapterEvents	サポートされています

表 B-1 サポートされる FC-HBA インタフェースとサポートされない FC-HBA インタフェース (続き)

FC-HBA インタフェース	Solaris I/O マルチパス化のサポート
HBA_RegisterForAdapterAddEvents	サポートされています
HBA_RegisterForAdapterPortEvents	サポートされています
HBA_RegisterForAdapterPortStatEvents	サポートされていません
HBA_RegisterForTargetEvents	サポートされています
HBA_RegisterForAdapterLinkEvents	サポートされていません
HBA_RegisterForAdapterTargetEvents	サポートされています
HBA_GetFC4Statistics	サポートされていません
HBA_GetFCPStatistics	サポートされていません
HBA_GetBindingCapability	サポートされていません
HBA_GetBindingSupport	サポートされていません
HBA_SetBindingSupport	サポートされていません
HBA_SetPersistentBindingV2	サポートされていません
HBA_GetPersistentBindingV2	サポートされていません
HBA_RemovePersistentBinding	サポートされていません
HBA_RemoveAllPersistentBindings	サポートされていません

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。

この付録は、次のセクションから構成されています。

- 105 ページの「**stmsboot** の実行中にシステムがブートに失敗する」
- 106 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 11/11 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, 86

A

ANSI 標準マルチパス管理 API, 35-49

Ap_Id, 87

C

cfgadm

-c, 89

-l, 87, 88

cfgadm -al -o show_SCSI_LUN, 87

cfgadm -c unconfigure, 97

cfgadm(1M), 88

cfgadm configure, 88

cfgadm_fp(1M), 91

F

fc-fabric, 92

FCAL, 14

FCoE

制限事項, 59-60

ポートの構成, 60-62

ポートの削除, 61

ポートの作成, 60-61

ポートのステータスのレポート, 61-62

I

Internet Protocol over Fibre Channel, 「IPFC」を参照
IPFC, 考慮事項, 67-70

L

LUN

SCSI デバイス, 94

認識, 53

マスク, 51

luxadm(1M), 34

M

modinfo, 87

mpathadm コマンド, 35-49

イニシエータポート

プロパティ, 36

自動フェイルバック、構成, 42

自動フェイルバックの構成, 42

特定の名前を持つ論理ユニット, 40

パスを無効にする, 47-49

パスを有効にする, 46-47

論理ユニット

ターゲットポートに関連付けられた, 38

フェイルオーバー, 43-46

プロパティおよび関連したパス, 37

mpt ドライバ, 65

N

NPIV ポート, 構成, 55–58

O

Oracle Solaris x86 ベースのシステム FC OS ブート
手順, 73–80

R

REPORT LUNS コマンド, 30

S

SAS デバイス

構成, 66

動的発見, 65

マルチパス化の考慮事項, 65

sd ドライバ, 65

Serial Attached SCSI, 「SAS」を参照

show_SCSI_LUN, 89, 94

snoop(1M), 67

ssd driver, 87

st driver, 87

stmsboot コマンド, 25

SAN デバイスの考慮事項, 52

考慮事項, 20

システムのクラッシュ, 106

デバイスパスの判別, 24

ブートの考慮事項, 53

マルチパス化の有効化および無効化, 17

T

T10 標準, 35–49

T11 標準, 14

い

イニシエータポート, プロパティの表示, 36

こ

構成

FCoE, 60–62

NPIV ポート, 55–58

SAN デバイス, 51–53

SAS デバイス, 66

仮想ファイバチャネルポート, 55–58

サードパーティー製デバイス, 30, 31

自動フェイルバック, 34

手動, 85–86

タスク, 19

ダンプ, 24

デバイス, 87

ファイバチャネルの概要, 19–21

ファブリック接続されたデバイス, 51–53

ファブリックデバイス, 88

マルチパス化, 24

マルチパス化機能, 23, 51–53

考慮事項

IPFC, 67–70

SAN デバイス, 51–52

SAS デバイス, 65

stmsboot コマンド, 20

StorageTek Traffic Manager, 20–21

サードパーティー製デバイスの構成, 30

デバイス固有, 24

デバイス名の変更, 24

ファブリックブートデバイス, 53

ポート単位の構成, 28–29

マルチパス化, 24

し

自動フェイルバック, 構成, 34

す

ストレージデバイス, 34

せ

制限事項, FCoE, 59–60

て

テープデバイスの持続的なバインド, 81-83

デバイス

構成, 88

ストレージ, 16

非対称, 16

デバイス管理, 17

デバイス、持続的なネーミング、テープ, 14

と

トラブルシューティング、マルチパス化, 105-107

は

発見

SAS デバイス, 65

動的, 65

ふ

ファイバチャネル

HBA, 14

IPFC SAN デバイスの構成, 67-72

IPFC の考慮事項, 67-70

LUN レベル情報の一覧表示, 87-88

Oracle Solaris x86 ベースシステムの

ブート, 73-80

アダプタのポートインスタンスの判別, 68-70

構成の概要, 19-21

サポートされる HBA インタフェース, 101-103

調停ループ, 14

マルチパス化構成, 23-24

マルチパス化の有効化と無効化, 25-27

ファイバチャネルポート、構成, 55-58

ファブリック接続されたホストポート, 88, 92

ファブリックデバイス構成, 88

ブート, 14

ブートディスク, 53

物理デバイス, 89

ブロードキャスト, 67

プロミスキャス (promiscuous) モード, 67

ほ

ポート単位の構成

考慮事項, 28-29

マルチパス化の有効化と無効化, 28-29

ま

マルチパス化

mpathadm コマンド, 35-49

SAS 考慮事項, 65

SAS ブートデバイスの構成, 66

機能, 15

サードパーティー製ストレージデバイスの構成, 30-33

サポート, 35-49

自動フェイルバックの構成, 34

手動構成, 85-86

デバイスノードの構成, 91-95

トラブルシューティング, 105-107

ファブリックデバイスノードの構成, 86

ブートの考慮事項, 53

ポート単位ベースでの有効化または無効化, 27-29

無効化、複数デバイスの構成, 90-91

無効、デバイスノードの構成, 88

有効化と無効化, 25-27

マルチパス化、有効化と無効化, 25-27

マルチパス管理 API プラグイン

プロパティ, 35

ライブラリ, 35

む

無効化、マルチパス化, 25-27

ゆ

有効化、マルチパス化, 25-27

