



# SunOS リファレンスマニュアル ル7: デバイスとネットワーク インタフェース



Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054  
U.S.A.

Part No: 819-1218-12  
2006年5月

Sun Microsystems, Inc. (以下米国 Sun Microsystems 社とします) は、本書に記述されている製品に含まれる技術に関連する知的財産権を所有します。特に、この知的財産権はひとつかそれ以上の米国における特許、あるいは米国およびその他の国において申請中の特許を含んでいることがあります。また、それらに限定されるものではありません。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

U.S. Government Rights Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者によって開発された素材を含んでいることがあります。

本製品に含まれる HG-MinchoL、HG-MinchoL-Sun、HG-PMinchoL-Sun、HG-GothicB、HG-GothicB-Sun、および HG-PGothicB-Sun は、株式会社リコーがリコービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェースマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、Sun のロゴマーク、Solaris のロゴマーク、Java Coffee Cup のロゴマーク、docs.sun.com、JumpStart、Solaris Web Start、Power Management、Sun ONE Application Server、Solaris Flash、Solaris Live Upgrade、Java および Solaris は、米国およびその他の国における米国 Sun Microsystems 社の商標、登録商標もしくは、サービスマークです。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。Copyright(C) OMRON Co., Ltd. 1995-2000. All Rights Reserved. Copyright(C) OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK for Solaris」は、株式会社ジャストシステムの著作物であり、「ATOK for Solaris」にかかる著作権、その他の権利は株式会社ジャストシステムおよび各権利者に帰属します。

「ATOK」および「推測変換」は、株式会社ジャストシステムの登録商標です。

「ATOK for Solaris」に添付するフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド』に添付のものを使用しています。

「ATOK for Solaris」に含まれる郵便番号辞書(7桁/5桁)は日本郵政公社が公開したデータを元に制作された物です(一部データの加工を行なっています)。

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは、OPEN LOOK のグラフィカル・ユーザインタフェースを実装するか、またはその他の方法で米国 Sun Microsystems 社との書面によるライセンス契約を遵守する、米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

本書で言及されている製品や含まれている情報は、米国輸出規制法で規制されるものであり、その他の国の輸出入に関する法律の対象となる場合があります。核、ミサイル、化学あるいは生物兵器、原子力の海洋輸送手段への使用は、直接および間接を問わず厳しく禁止されています。米国が禁輸の対象としている国や、限定はされませんが、取引禁止顧客や特別指定国民のリストを含む米国輸出排除リストで指定されているものへの輸出および再輸出は厳しく禁止されています。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法(外為法)に定められる戦略物資等(貨物または役務)に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: man pages section 7: Device and Network Interfaces

Part No: 816-5177-11

Revision A

# 目次

---

はじめに .....	5
序章 .....	9
Intro(7) .....	10
デバイスとネットワークインタフェース .....	13
afb(7d) .....	14
bge(7D) .....	15
cpr(7) .....	19
cvc(7D) .....	21
cvcredir(7D) .....	22
dtrace(7D) .....	23
ecpp(7D) .....	24
fas(7D) .....	31
ffb(7D) .....	40
ge(7D) .....	41
glm(7D) .....	45
hme(7D) .....	51
m64(7D) .....	56
md(7D) .....	57
mpt(7D) .....	61
pm(7D) .....	66
qfe(7d) .....	73



# はじめに

---

## 概要

SunOS リファレンスマニュアルは、初めて SunOS を使用するユーザーやすでにある程度の知識を持っているユーザーのどちらでも対応できるように解説されています。このマニュアルを構成するマニュアルページは一般に参照マニュアルとして作られており、チュートリアルな要素は含んでいません。それぞれのコマンドを実行すると、どのような結果が得られるかについて、詳しく説明されています。なお、各マニュアルページの内容はオンラインでも参照することができます。

このマニュアルは、マニュアルページの内容によっていくつかのセクションに分かれています。各セクションについて以下に簡単に説明します。

- セクション1は、オペレーティングシステムで使えるコマンドを説明します。
- セクション1Mは、システム保守や管理用として主に使われるコマンドを説明します。
- セクション2は、すべてのシステムコールについて説明します。ほとんどのシステムコールに1つまたは複数のエラーがあります。エラーの場合、通常ありえない戻り値が返されます。
- セクション3は、さまざまなライブラリ中の関数について説明します。ただし、UNIX システムプリミティブを直接呼び出す関数については、セクション2で説明しています。
- セクション4は、各種ファイルの形式について説明します。また、ファイル形式を宣言するC構造体を適用できる場合には随時説明しています。
- セクション5は、文字セットテーブルなど他のセクションには該当しないものについて説明します。
- セクション7は、特殊なハードウェア周辺装置またはデバイスドライバに関するさまざまな特殊ファイルについて説明します。STREAMS ソフトウェアドライバ、モジュール、またはシステムコールの STREAMS 汎用セットについても説明します。
- セクション9は、カーネル環境でデバイスドライバを記述するのに必要な参照情報を提供します。ここでは、デバイスドライバインタフェース (DDI) とドライバ/カーネルインタフェース (DKI) という2つのデバイスドライバインタフェース仕様について説明します。
- セクション9Fは、デバイスドライバが使用できるカーネル関数について説明します。

以下に、このマニュアルの項目を表記されている順に説明します。ほとんどのマニュアルページが下記の項目からなる共通の書式で書かれていますが、必要でない項目については省略されています。たとえば、記述すべきバグがコマンドにない場合などは、「使用上の留意点」という項目はありません。各マニュアルページの詳細は各セクションの `intro` を、マニュアルページの一般的な情報については `man(1)` を参照してください。

名前	コマンドや関数の名称と概略が示されています。
形式	コマンドや関数の構文が示されています。標準パスにコマンドやファイルが存在しない場合は、フルパス名が示されます。字体は、コマンド、オプションなどの定数にはボールド体 ( <b>bold</b> ) を、引数、パラメータ、置換文字などの変数にはイタリック体 ( <i>Italic</i> ) または <日本語訳> を使用しています。オプションと引数の順番は、アルファベット順です。特別な指定が必要な場合を除いて、1文字の引数、引数のついたオプションの順に書かれています。
	以下の文字がそれぞれの項目で使われています。
	[ ] このかっこに囲まれたオプションや引数は省略できます。このかっこが付いていない場合には、引数を必ず指定する必要があります。
	... 省略符号。前の引数に変数を付けたり、引数を複数指定したりできることを意味します (例: 'filename...')。
	区切り文字 (セパレータ)。この文字で分割されている引数のうち1つだけを指定できます。
	{ } この大かっこに囲まれた複数のオプションや引数は省略できます。かっこ内を1組として扱います。
プロトコル	この項が使われているのは、プロトコルが記述されているファイルを示すサブセクション 3R だけです。パス名は常にボールド体 ( <b>bold</b> ) で示されています。
機能説明	コマンドの機能とその動作について説明します。実行時の詳細を説明していますが、オプションの説明や使用例はここでは示されていません。対話形式のコマンド、サブコマンド、リクエスト、マクロ、関数などに関しては「使用法」で説明します。
IOCTL	セクション 7 だけに使用される項です。 <code>ioctl(2)</code> システムコールへのパラメータは <code>ioctl</code> と呼ばれ、適切なパラメータを持つデバイスクラスのマニュアルページだけに記載されています。特定のデバイスに関する <code>ioctl</code> は、(そのデバイスのマニュアルページに) アルファベット順に記述されています。デバイスの特定のクラスに関する <code>ioctl</code> は、 <code>mtio(7I)</code> のように <code>io</code> で終わる名前が付いているデバイスクラスのマニュアルページに記載されています。

オプション	各オプションがどのように実行されるかを説明しています。「形式」で示されている順に記述されています。オプションの引数はこの項目で説明され、必要な場合はデフォルト値を示します。
オペランド	コマンドのオペランドを一覧表示し、各オペランドがコマンドの動作にどのように影響を及ぼすかを説明しています。
出力	コマンドによって生成される出力 (標準出力、標準エラー、または出力ファイル) を説明しています。
戻り値	値を返す関数の場合、その値を示し、値が返される時の条件を説明しています。関数が 0 や -1 のような一定の値だけを返す場合は、値と説明の形で示され、その他の場合は各関数の戻り値について簡単に説明しています。void として宣言された関数はこの項では扱いません。
エラー	失敗の場合、ほとんどの関数はその理由を示すエラーコードを <code>errno</code> 変数の中に設定します。この項ではエラーコードをアルファベット順に記述し、各エラーの原因となる条件について説明します。同じエラーの原因となる条件が複数ある場合は、エラーコードの下にそれぞれの条件を別々のパラグラフで説明しています。
使用法	この項では、使用する際の手がかりとなる説明が示されています。特定の決まりや機能、詳しい説明の必要なコマンドなどが示されています。組み込み機能については、以下の小項目で説明しています。
	コマンド 修飾子 変数 式 入力文法
使用例	コマンドや関数の使用例または使用方法を説明しています。できるだけ実際に入力するコマンド行とスクリーンに表示される内容为例にしています。例の中には必ず <code>example%</code> のプロンプトが出てきます。スーパーユーザーの場合は <code>example#</code> のプロンプトになります。例では、その説明、変数置換の方法、戻り値が示され、それらのほとんどが「形式」、「機能説明」、「オプション」、「使用法」の項からの実例となっています。
環境	コマンドや関数が影響を与える環境変数を記述し、その影響について簡単に説明しています。
終了ステータス	コマンドが呼び出しプログラムまたはシェルに返す値と、その状態を説明しています。通常、正常終了には 0 が返され、0 以外の値はそれぞれのエラー状態を示します。

ファイル	マニュアルページが参照するファイル、関連ファイル、およびコマンドが作成または必要とするファイルを示し、各ファイルについて簡単に説明しています。
属性	属性タイプとその対応する値を定義することにより、コマンド、ユーティリティ、およびデバイスドライバの特性を一覧しています。詳細は <code>attributes(5)</code> を参照してください。
関連項目	関連するマニュアルページ、当社のマニュアル、および一般の出版物が示されています。
診断	エラーの発生状況と診断メッセージが示されています。メッセージはボールド体 ( <b>bold</b> ) で、変数はイタリック体 ( <i>Italic</i> ) または <日本語訳> で示されており、C ロケール時の表示形式です。
警告	作業に支障を与えるような現象について説明しています。診断メッセージではありません。
注意事項	それぞれの項に該当しない追加情報が示されています。マニュアルページの内容とは直接関係のない事柄も参照用に扱っています。ここでは重要な情報については説明していません。
使用上の留意点	すでに発見されているバグについて説明しています。可能な場合は対処法も示しています。



参照

序章

## 名前 Intro - 特殊ファイルの序章

### 機能説明

本セクションは、システム上で利用可能なさまざまなデバイスとネットワークインタフェースについて説明します。説明されているインタフェースの種類には、文字型デバイス、ブロック型デバイス、STREAMS モジュール、ネットワークプロトコル、ファイルシステム、ドライバサブシステムとドライバクラスの `ioctl` 要求、などがあります。

本セクションは次のように分けられています。

(7D) システムは、さまざまなハードウェアデバイス (磁気テープ、シリアル通信回線、マウス、フレームバッファ、疑似ターミナルや疑似ウィンドウなどの仮想デバイス) 用のドライバを提供します。

特定のハードウェア周辺装置やデバイスドライバを参照する特殊ファイル、STREAMS デバイスドライバについて説明しています。ハードウェアデバイスおよびそれに対応するデバイスドライバの特徴についても、適宜説明しています。

アプリケーションは、デバイスの特殊ファイルを介してデバイスにアクセスします。デバイスドライバを使用するために必要なアプリケーションプログラムインタフェース (API) 情報と共に、デバイスの特殊ファイルは、デバイスにアクセスするために使用されるものとして指定されています。

デバイス特殊ファイルはすべて `/devices` ディレクトリに置かれています。`/devices` ディレクトリ階層は、システム上に構成されているシステムバス、コントローラ、デバイスの階層構造を反映しようとしています。`/devices` ディレクトリにある特殊ファイルの論理デバイス名は、`/dev` に置かれています。`/devices` に置かれているすべての特殊ファイルが、`/dev` ディレクトリに論理デバイス名を持っているわけではありませんが、アプリケーションは、可能な場合は論理デバイス名を使用してデバイスを参照します。論理デバイス名は、そのデバイスのマニュアルページの「ファイル」の項目に記載されています。

セクション7D には、該当する場合にはドライバ構成についても説明しています。多くのデバイスドライバには、`driver_name.conf` という形式のドライバ構成ファイルがあります (`driver.conf(4)` を参照)。ドライバ構成ファイルに記述されている構成情報は、ドライバやデバイスを構成するために使用されます。ドライバ構成ファイルは、ディレクトリ `/kernel/drv` および `/usr/kernel/drv` にあります。プラットフォームに依存するドライバの構成ファイルは、`/platform/'uname -i'/kernel/drv` ディレクトリにあります。`'uname -i` は、`-i` オプション付きの `uname(1)` コマンドの出力結果です。

いくつかのドライバ構成ファイルには、ユーザーが構成できる属性が含まれています。ドライバ構成ファイルに加えた変更内容は、システムをリポートするかまたはドライバをいったん削除して再度追加すると、有効になります (`rem_drv(1M)` および `add_drv(1M)` を参照)。

- (7FS) SunOS でサポートされている複数のファイルシステムのプログラムインタフェースについて説明します。
- (7I) クラスまたはサブドライバに適用される `ioctl` 要求について説明します。たとえば、大部分のテープデバイスに適用される `ioctl` 要求については、`mtio(7I)` で説明しています。特定のデバイスにのみ適用される `ioctl` 要求については、そのデバイスのマニュアルページで説明しています。そのようなデバイスのマニュアルページは、セクション 7I に記載されている `ioctl` 要求に対する例外についても説明しています。
- (7M) STREAMS モジュールについて説明します。STREAMS ドライバについてはセクション 7D で説明します。`streamio(7I)` には、STREAMS モジュールとインタフェースを STREAMS フレームワークで操作するために使用される `ioctl` 要求のリストが含まれています。STREAMS モジュールに固有の `ioctl` 要求については、そのモジュールのマニュアルページで説明しています。
- (7P) SunOS で利用できるさまざまなネットワークプロトコルについて説明しています。

SunOS は、ソケットに基づくネットワーク通信と、STREAMS に基づくネットワーク通信をサポートしています。`inet(7P)` に記述されているインターネットファミリは、システムがその他多数のプロトコルをサポートしている場合でも、SunOS でサポートされる主プロトコルです。`raw` インタフェースによって、パケット分割や再アセンブリ、ルーティング、アドレス指定、ソケットに基づく実装における基本転送など、下位レベルのサービスが提供されます。通常、インターネットファミリプロトコルを使用する通信機能にアクセスするには、ソケットを結合する時に `AF_INET` アドレスファミリを指定します。詳細は、`socket(3SOCKET)` を参照してください。

インターネットにおける主要プロトコルは、次のとおりです。

- インターネットプロトコル (IP; Internet Protocol)。共通のデータグラム形式をサポートします。詳細は `ip(7P)` で説明されています。これは、`AF_INET` ドメイン中の `SOCK_RAW` 型ソケットのデフォルトのプロトコルです。
- 伝送制御プロトコル (TCP; Transmission Control Protocol)。`tcp(7P)` を参照してください。これは、`SOCK_STREAM` 型ソケットのデフォルトのプロトコルです。
- ユーザーデータグラムプロトコル (UDP; User Datagram Protocol)。`udp(7P)` を参照してください。これは、`SOCK_DGRAM` 型ソケットのデフォルトのプロトコルです。
- アドレス解決プロトコル (ARP; Address Resolution Protocol)。`arp(7P)` を参照してください。
- インターネット制御メッセージプロトコル (ICMP; Internet Control Message Protocol)。`icmp(7P)` を参照してください。

関連項目 [add\\_drv\(1M\)](#), [rem\\_drv\(1M\)](#), [intro\(3\)](#), [ioctl\(2\)](#), [socket\(3SOCKET\)](#), [driver.conf\(4\)](#), [arp\(7P\)](#), [icmp\(7P\)](#), [inet\(7P\)](#), [ip\(7P\)](#), [mtio\(7I\)](#), [st\(7D\)](#), [streamio\(7I\)](#), [tcp\(7P\)](#), [udp\(7P\)](#)

『Solaris のシステム管理 (IP サービス)』

『STREAMS Programming Guide』

『Writing Device Drivers』

コマンド一覧	名前	説明
	<a href="#">Intro(7)</a>	特殊ファイルの序章
	<a href="#">afb(7d)</a>	Elite3D グラフィックスアクセラレータドライバ
	<a href="#">bge(7D)</a>	Broadcom BCM5704 用 SUNW,bge Gigabit Ethernet ドライバ
	<a href="#">cpr(7)</a>	保存停止および復元再開モジュール
	<a href="#">cvc(7D)</a>	仮想コンソールドライバ
	<a href="#">cvcredir(7D)</a>	仮想コンソール入出力先変更ドライバ
	<a href="#">ecpp(7D)</a>	IEEE 1284 互換の平行ポートドライバ
	<a href="#">fas(7D)</a>	FAS SCSI ホストバスアダプタドライバ
	<a href="#">ffb(7D)</a>	24 ビット UPA カラーフレームバッファとグラフィックスアクセラレータ
	<a href="#">ge(7D)</a>	GEM GigabitEthernet デバイスドライバ
	<a href="#">glm(7D)</a>	GLM SCSI ホストバスアダプタドライバ
	<a href="#">hme(7D)</a>	SUNW,hme Fast-Ethernet デバイスドライバ
	<a href="#">m64(7D)</a>	PGX、PGX24、PGX64 フレームバッファデバイスドライバ
	<a href="#">mpt(7D)</a>	SCSI ホストバスアダプタドライバ
	<a href="#">pm(7D)</a>	電源管理システムのドライバ
	<a href="#">qfe(7d)</a>	SUNW,qfe Quad Fast-Ethernet デバイスドライバ

参照

デバイスとネットワークインタフェース

名前	afb - Elite3D グラフィックアクセラレータドライバ
機能説明	afb ドライバは、Sun Elite3D グラフィックアクセラレータのデバイスドライバです。 afbdaemon プロセスは、システムの起動時と、一時停止・再開サイクルの再開処理中に、afb マイクロコードを読み込みます。
ファイル	<code>/dev/fbs/afb<i>n</i></code> デバイス特殊ファイル  <code>/usr/lib/afb.unicode</code> afb マイクロコード  <code>/usr/sbin/afbdaemon</code> afb マイクロコード読み込みプログラム
関連項目	afbconfig(1M)

名前	bge – Broadcom BCM57xx 用 SUNW,bge Gigabit Ethernet ドライバ
形式	/dev/bge*
機能説明	<p>bge Gigabit Ethernet ドライバは、マルチスレッド化された、ロード可能かつ複製可能な GLD ベースの STREAMS ドライバであり、システムマザーボードに装着された Broadcom BCM57xx (BCM5703/5704、および x86 上の BCM5700/5701/5705/5782) Gigabit Ethernet コントローラでの Data Link Provider Interface (dLpi(7P)) をサポートします。BCM5700/BCM5701/BCM5704S を除いてこれらのデバイスには MAC と PHY 機能が組み込まれており、RJ-45 コネクタで 3 種類の通信速度での (銅線による) Ethernet 動作を行うことができます。</p> <p>(BCM5700/BCM5701/BCM5704S の MAC チップセットには PHY は統合されていません)。</p> <p>bge ドライバ機能には、コントローラの初期設定、フレームの送受信、プロミスキュアスとマルチキャストのサポート、エラーの回復と報告が含まれます。</p> <p>bge ドライバとハードウェアは、1000Base-T 標準規格で指定されたプロトコルである「自動ネゴシエーション」に対応しています。自動ネゴシエーションにより、各デバイスではそのデバイスに備わっている機能を通知し、そのピア (接続相手) を検出できるようになります。手動で構成を行うことなく、双方の接続相手でサポートされる最適な共通モードが自動的に選択されて、最大のスループットが得られます。bge ドライバを使用すると、接続相手側で通知される機能とは関係なく、通知する機能を、最高速度未満に (最高速のインタフェースを必要としないよう) 構成したり、特定の動作モードに設定したりすることもできます。</p>
アプリケーションプログラミングインタフェース	<p>システムのマザーボードに装着されているすべての BCM57xx デバイス (BCM5703/5704、および x86 上の BCM5700/5701/5705/5782) にアクセスするには、クローニング文字特殊デバイスである /dev/bge を使用します。</p> <p>bge ドライバは使用される /kernel/misc/gld によって異なります。この /kernel/misc/gld は、LAN ドライバに必要な DLPI と STREAMS 機能を bge ドライバに提供する、ロード可能なカーネルモジュールです。サポートしている基本式についての詳細は、<a href="#">gld(7D)</a> を参照してください。</p> <p>開かれているストリームを特定のデバイス (PPA) に関連付けるには、DL_ATTACH_REQ メッセージを明示的に送信する必要があります。PPA ID は符号なし整数データ型として解釈され、対応するデバイスインスタンス (ユニット) 番号を示します。PPA フィールド値がシステムで有効なデバイスインスタンス番号と対応していない場合は、ドライバからエラー (DL_ERROR_ACK) が返されます。デバイスは、最初に接続されるときに初期化され、最後に切り離されるときに初期化解除 (停止) されます。</p> <p>ドライバが DL_INFO_REQ に応答して DL_INFO_ACK 基本式の中で返す値は、以下のような値です。</p> <ul style="list-style-type: none"> <li>■ 最大 SDU は 1500 (ETHERMTU - &lt;sys/ethernet.h&gt; で定義済み)。</li> <li>■ 最小 SDU は 0。</li> <li>■ DLSAP アドレス長は 8。</li> </ul>

- MACタイプはDL\_ETHER。
- SAPの長さの値は-2。これは、DLSAPアドレス内では、物理アドレス構成要素のすぐ後に2バイトのSAP構成要素が付くことを意味します。
- ブロードキャストアドレス値は、Ethernet/IEEEブロードキャストアドレス (FF:FF:FF:FF:FF:FF)。

DL\_ATTACHED状態になったときは、DL\_BIND\_REQを送信して、特定のサービスアクセスポイント(SAP)をストリームに関連付ける必要があります。

## 構成

デフォルトでは、bgeドライバで自動ネゴシエーションが実行されて接続速度とモードが選択されます。接続速度とモードは、IEEE803.2標準規格に従い、以下のいずれかの組み合わせのうちから選択されます。

- 1000 Mbps、全二重
- 1000 Mbps、半二重
- 100 Mbps、全二重
- 100 Mbps、半二重
- 10 Mbps、全二重
- 10 Mbps、半二重

自動ネゴシエーションプロトコルでは、双方の接続相手でサポートされる最適な共通モードとして、以下が自動的に選択されます。

- 速度(1000 Mbps、100 Mbps、または10 Mbps)
- 動作モード(全二重または半二重)

bge デバイスはすべての動作モードに対応しているので、結果的には相手側のデバイスでサポートされる最高のスループットモードが選択されることになります。

また、`ndd(1M)`を使用してbgeデバイスが通知する機能を設定することもできます。bgeドライバは、`adv_`で始まる名前が付いたパラメータをサポートします(下記を参照)。これらのパラメータには、デバイスでその動作モードを通知するかどうかを決定するプール型の値が含まれています。`adv_pause_cap`は、リンクパートナーに半二重または全二重ポーズを通知するかどうかを示します。そして、`adv_asym_pause_cap`を設定すると、bgeは、非同期ポーズが必要なリンクパートナーに非同期ポーズを通知できます。さらに、`adv_100T4_cap`を使用すると、bgeは、自分の100T4機能をリンクパートナーに通知できます。`adv_autoneg_cap`パラメータでは、自動ネゴシエーションを実行するかどうかを制御します。`adv_autoneg_cap`が0に設定されている場合には、ドライバは、以下に示す優先順位で、最初にゼロでないパラメータが指定されている動作モードを実行します。

(最高の優先順位/最大のスループット)

<code>adv_1000fdx_cap</code>	1000Mbps 全二重
<code>adv_1000hdx_cap</code>	1000Mbps 半二重
<code>adv_100fdx_cap</code>	100Mbps 全二重
<code>adv_100hdx_cap</code>	100Mbps 半二重
<code>adv_10fdx_cap</code>	10Mbps 全二重
<code>adv_10hdx_cap</code>	10Mbps 半二重

(最低の優先順位/最小のスループット)



たとえば、デバイス「bge2」がギガビット機能を通知しないようにするには、(スーパーユーザーとして)以下のように入力します。

```
# ndd -set /dev/bge2 adv_1000hdx_cap 0
# ndd -set /dev/bge2 adv_1000fdx_cap 0
```

デフォルトではすべての機能が使用可能になっています。いずれかの機能のパラメタを変更すると、接続相手では新たに変更された機能を使って接続速度と二重モードの再ネゴシエーションが行われるため、接続が切断されることに注意してください。

現在のパラメタ設定は、`ndd -get` を使用すると確認できます。また、ドライバは、`ndd` パラメタを使って現在の接続の状態、速度、および作業モードの設定をエクスポートします(これらの設定は読み取り専用で、変更することはできません)。たとえば、デバイス `bge0` の接続状態を調べるには、次のように入力します。

```
# ndd -get /dev/bge0 link_status
1
# ndd -get /dev/bge0 link_speed
100
# ndd -get /dev/bge0 link_duplex
2
# ndd -get /dev/bge0 link_rx_pause
1
# ndd -get /dev/bge0 link_tx_pause
1
```

上記の出力結果は、このリンクがその rx/tx 方向のポーズ機能を使用して、100Mbps 全二重で起動および動作していることを示します。さらに、このドライバは `loop_mode` でその作業モードをエクスポートしています。`loop_mode` を 0 に設定した場合、ループバックモードは無効になります。

## ファイル

```
/dev/bge*
  文字特殊デバイス

/kernel/drv/sparcv9/bge
  SPARC bge ドライババイナリ

/kernel/drv/amd64/bge
  64 ビット x86 プラットフォーム用のカーネルモジュール

/platform/i86pc/kernel/drv/bge
  32 ビット x86 bge ドライバのバイナリ

/platform/i86pc/kernel/drv/amd64/bge
  64 ビット x86 bge ドライバのバイナリ
```

## 属性

以下の属性については、`attributes(5)` を参照してください。

---

属性タイプ	属性値
アーキテクチャ	SPARC、x86

## 関連項目

[attributes\(5\)](#)、[gld\(7D\)](#)、[streamio\(7I\)](#)、[dlpi\(7P\)](#)

『Writing Device Drivers』

『STREAMS Programming Guide』

『Network Interfaces Programmer's Guide』

名前	cpr - 保存停止および復元再開モジュール
形式	/kernel/misc/cpr
機能説明	<p>cpr は、システム全体を保存停止したり復元再開したりするための読み込み可能なモジュールです。電力の節約や移動の目的で一時的に電源を切断するために、システムの保存停止が必要になる場合があります。ハードウェアの再構成や交換を行う場合は、cpr モジュールを使用せずに通常の停止を行ってください。復元再開操作が成功するには、ハードウェア構成が同じままであることが重要です。システムが保存停止されるときに、システム全体の状態は、復元再開が実行されるまで不揮発性の記憶装置に保存されます。</p> <p>タイプ5 キーボードの電源制御キーと Shift-電源制御キーの操作は、cpr モジュールを使用します。システムにインストールされていて、このモジュールを使用する可能性があるユーティリティーは、uadmin(1M) と uadmin(2) の2つです。</p> <p>cpr モジュールは、システムを保存停止するときに次の処理を実行します。まず、SIGFREEZE 信号がすべてのユーザースレッドに送られ、その後、それらのスレッドは停止されます。マルチプロセッサシステムの場合は、システムを単一プロセッサモードにします。次に、ユーザーのダーティーページは、バックアップ用の記憶装置にスワップアウトされ、すべてのファイルシステムは同期(sync) が取られます。すべての装置は活動停止状態になり、システム割り込みは無効になります。システムの保存停止を完了するために、カーネルのメモリーページと残りのユーザーページは、圧縮形式でルートファイルシステムに書き込まれます。</p> <p>再びシステムの電源が入れると、基本的に、保存停止の処理手順と逆の処理が実行されます。カーネルイメージは、起動プログラムの /cprboot によってルートファイルシステムから復元されます。割り込みや装置は、以前の状態に復元されます。最後に、ユーザースレッドは再スケジュールされ、システムの復元再開を待つプロセスに SIGTHAW を伝送して通知します。他のプロセッサがある場合は、復元され、オンラインになります。これで、システムは完全に保存停止前の状態に戻ります。</p> <p>いくつかの場合において、cpr モジュールが保存停止操作を実行することができない場合があります。出荷時の標準構成以外の追加装置がある場合には、追加装置が cpr に対応していない可能性があります。この場合は、保存停止が失敗し、エラーメッセージが出力されます。保存停止が成功するようにするには、装置を取り除くか、またはそのデバイスドライバを読み込み解除する必要があります。装置の製造元に連絡し、cpr に対応しているバージョンのデバイスドライバを取得してください。保存停止は、装置またはプロセスが、リアルタイム処理のような、余裕のない、または時間に関係する処理を行っている場合も失敗します。この場合は、システムは現在の実行状態のままとなります。失敗を通知するメッセージがコンソールに表示され、状態は呼び出し側に返されます。いったんシステムが正常に保存停止されると、ハードウェアの再構成など外部からの影響がないかぎり、復元再開操作は必ず成功します。</p> <p>一部のネットワークベースのアプリケーションは、保存停止・復元再開処理の際に失敗する場合があります。この問題は、主に背後のネットワークプロトコルと関係するアプリケーションに依存します。一般的に、再試行して自動的に接続を確立するアプリケー</p>

ションは、復元再開の際に何事もなかったかのように動作を継続します。再試行して自動的に接続を確立しないアプリケーションは、多くの場合失敗します。

保存停止と復元再開の処理速度は、システムの処理速度、メモリーの容量、負荷によって、15秒から数分の範囲になります。通常は、1分くらいです。

ファイル `/cprboot` cpr用の特別な起動プログラム  
`/.CPR` システム状態ファイル

属性 以下の属性については、`attributes(5)`を参照してください。

属性の種類	属性の値
使用条件	SUNWcpr

関連項目 `uadmin(1M)`, `uadmin(2)`, `attributes(5)`

注意事項 マルチプロセッサプラットフォームで保存停止または復元再開を動作させる場合には、すべてのCPUを制御する必要があります。すべてのCPUが停止できない場合は、保存停止が拒否されるため、保存停止が起動された時にMPテスト(`sundiag CPU`テストなど)を実行しないでください。

テープ、フロッピーディスクのような特定の装置操作は、取り外し可能な媒体であるために復元再開できません。これらの動作は、保存停止のときに検出して、保存停止が完全に正常終了する前に停止する必要があります。

使用上の留意点 `SIGFREEZE` および `SIGTHAW` 信号は、Solaris 2.4 リリースに適切に実装されていません。これらの信号は、将来のリリースで使用可能となります。これについては、保存停止または復元再開の際に追加のタスクを必要とするような、特別にカスタマイズされたアプリケーションのみについて問題となります。現在では、そのようなアプリケーションはありません。

非常にまれに、システムが復元再開の処理の初期の段階で失敗することがあります。この場合、システムが正常に復元再開も起動も行わないループに入っていることが論理上考えられます。このようなループに入った場合は、`L1+A` を使って `prom ok` プロンプトを表示し、以下のコマンドを入力してください。

```
<ok> set-default boot-file
```

このコマンドによってシステムはリセットされ、次の電源投入ではシステムが正常に起動します。

名前	cvc – 仮想コンソールドライバ
機能説明	<p>cvc 仮想コンソールドライバは、ネットワークコンソールを利用可能にする STREAMS ベースの擬似ドライバです。cvc へのドライバインタフェースは <code>console(7D)</code> です。</p> <p>cvc ドライバは、論理的に <code>console</code> ドライバの下位に位置するドライバです。cvc は、ネットワークコンソール接続が動作中の場合に、コンソール出力を <code>cvcredir(7D)</code> ドライバに切り替えます。ネットワークコンソール接続が動作していない場合は、cvc はコンソールの出力先を内部ハードウェアのインタフェースに変更します。</p> <p>cvc ドライバは、<code>cvcredir</code> と内部ハードウェアからのコンソール入力を受け取り、<code>/dev/console</code> に結び付けられているプロセスに渡します。</p>
注意事項	<p>cvc の機能は、SunOS の <code>wscons(7D)</code> の機能に取って代わるものです。この両方を、同時に使用しないでください。<code>wscons</code> ドライバは、直接に接続されたコンソール (フレームバッファやキーボード) を持つシステムで有用ですが、ローカルのキーボードやフレームバッファを持たない、cvc を使用するプラットフォームでは有用ではありません。</p>
属性	以下の属性については、 <code>attributes(5)</code> を参照してください。

属性タイプ	属性値
アーキテクチャ	Sun Enterprise 10000 サーバー、Sun Fire 15000 サーバー
使用条件	SUNWcvc.u

関連項目	<p><code>cvcd(1M)</code>, <code>attributes(5)</code>, <code>console(7D)</code>, <code>cvcredir(7D)</code>, <code>wscons(7D)</code></p> <p>『Sun Enterprise 10000 SSP リファレンスマニュアル』</p> <p>System Management Services (SMS) リファレンスマニュアル</p>
------	--

名前 cvcredir – 仮想コンソール入出力先変更ドライバ

機能説明 cvcredir 仮想コンソール入出力先変更ドライバは、一部のプラットフォームで提供されているネットワークコンソールをサポートする、STREAMS ベースの疑似ドライバです。cvcredir ドライバは、仮想コンソールドライバ [cvc\(7D\)](#) および仮想コンソールデーモン [cvcd\(1M\)](#) と連携して機能します。

cvcredir ドライバは、cvc からのコンソール出力を受け取って cvcd に渡します。また、cvcd からのコンソール入力を受け取って cvc に渡します。

属性 以下の属性については、[attributes\(5\)](#) を参照してください。

属性タイプ	属性値
アーキテクチャ	Sun Enterprise 10000 サーバー、Sun Fire 15000 サーバー
使用条件	SUNWcvc.u

関連項目 [cvcd\(1M\)](#), [attributes\(5\)](#), [console\(7D\)](#), [cvc\(7D\)](#)

『Sun Enterprise 10000 SSP リファレンスマニュアル』

System Management Services (SMS) リファレンスマニュアル

名前 dtrace – DTrace 動的トレース機能

機能説明 dtrace ドライバは、DTrace ソフトウェアに対して動的な計測機能とトレース機能を提供します。また、このドライバは組み込み dtrace プロバイダでもあります。dtrace ドライバは公開インタフェースではありません。このプロバイダが提供する計測機能にアクセスするには、dtrace(1M) などの DTrace ツールを使用します。DTrace 機能で利用可能な、ドキュメントが公開されたインタフェースや dtrace プロバイダのプロープについては、『Solaris 動的トレースガイド』を参照してください。

属性 属性についての詳細は、マニュアルページの attributes(5) を参照してください。

属性タイプ	属性値
使用条件	SUNWdtrp
インタフェースの安定性	非公開

関連項目 dtrace(1M), libdtrace(3LIB), attributes(5)

『Solaris 動的トレースガイド』

名前	ecpp – IEEE 1284 互換のバラレルポートドライバ
形式	<pre>#include &lt;sys/types.h&gt;  #include &lt;sys/ecppio.h&gt;  ecpp@unit-address</pre>
機能説明	<p>ecpp ドライバは、IEEE 1284 に準拠したデバイスに対して双方向のインタフェース、セントロニクスデバイスに対して単方向の順方向インタフェースを提供します。ecpp ドライバは、IEEE 1284 互換プロトコル、ニブルプロトコル、ECP プロトコル、およびセントロニクスプロトコルをサポートしています。ECPP_COMPAT_MODE および ECPP_CENTRONICS の動作モードは、論理的に同じであるハンドシェイク方式のプロトコルです。ただし、ECPP_COMPAT_MODE をサポートしているデバイスは、IEEE 1284 に準拠したデバイスです。IEEE 1284 に準拠したデバイスは、少なくとも ECPP_COMPAT_MODE および ECPP_NIBBLE_MODE をサポートしています。セントロニクスデバイスは、ECPP_CENTRONICS モードをサポートします。</p> <p>デフォルトでは、ECPP_COMPAT_MODE デバイスのハンドシェイク方式のストロープパルス幅は、500 ナノ秒です。このモードの場合、データの順方向転送は DMA によって行われます。デフォルトでは、ECPP_CENTRONICS デバイスのストロープパルス幅は、2 マイクロ秒です。これらのデバイスでは、順方向転送は PIO を介して行われます。ECPP_COMPAT_MODE および ECPP_CENTRONICS デバイスのデフォルトの動作は、ecpp.conf で定義された調整可能な変数によって変更可能です。</p> <p>ecpp ドライバは、「排他的に使用する」デバイスです。すなわち、デバイスがすでに開いている場合は、その後に関心しても EBUSY によって失敗します。</p>
デフォルトの動作	<p>ecpp デバイスが開かれるたびに、そのデバイスは EBUSY としてマークされ、設定変数はデフォルト値に設定されます。write_timeout は、90 秒に設定されます。</p> <p>ecpp ドライバは、次に説明するアルゴリズムに従って、モード変数を設定します。まず、ecpp ドライバは、open(2) が実行されている間に ECPP_ECP_MODE モードへの移行を試みます。これに失敗すると、ドライバは、モードを ECPP_NIBBLE_MODE モードにするようにデバイスとネゴシエーションを行います。このネゴシエーションに失敗すると、ドライバは ECPP_CENTRONICS モードで動作します。デバイスを開くのに成功すると、IEEE 1284 に準拠するデバイスは、ECPP_ECP_MODE または ECPP_NIBBLE_MODE のいずれかで逆方向アイドルフェーズのアイドル状態になります。その後 write(2) 呼び出しにより、ドライバは ECPP_ECP_MODE または ECPP_COMPAT_MODE のいずれかの順方向リンクフェーズに移動させます。この転送が終了するとリンクはアイドル状態に戻ります。</p> <p>アプリケーションは、ECPPIOC_SETPARMS ioctl(2) の呼び出しによって、デバイスを特定のモードにするようにネゴシエーションを行ったり write_timeout 値を設定したりすることができます。ネゴシエーションが成功するには、要求されたモードにホストワークステーションと周辺デバイスの両方が対応している必要があります。</p>



調整	<p>ecpp ドライバの特徴は、<code>/kernel/drv/ecpp.conf</code> に記述された変数で調整可能です。これらの変数は、システム起動中にドライバによって読み込まれます。変数を調整するには、<code>ecpp.conf</code> ファイルを編集し、<code>update_drv(1M)</code> を起動してカーネルにファイルを再度読み込ませます。</p> <p>セントロニクス周辺機器および特定の IEEE 1284 に準拠した周辺機器は、高速ハンドシェイキングモードで動作するパラレルポートと共には動作しません。印刷に関する問題が発生した場合は、<code>"fast-centronics"</code> および <code>"fast-1284-compatible"</code> を <code>"false"</code> に設定します。詳細は、<code>/kernel/drv/ecpp.conf</code> を参照してください。</p>
読み取り・書き込み操作	<p>ecpp ドライバは、全二重の STREAMS デバイスドライバです。アプリケーションが IEEE 1284 に準拠するデバイスに書き込んでいる間、別のスレッドはそのデバイスから読み取ることができます。</p>
書き込み操作	<p><code>write(2)</code> 操作は、ストリームヘッドに正常に書き込まれたバイト数を返します。セントロニクスデバイスがデータを転送している間に異常が発生した場合は、状態ビットの内容はエラーが発生した時のものが保存されます。アプリケーションプログラムは、この状態ビットを <code>BPPIOC_GETERR ioctl(2)</code> 呼び出しを使用して取り出すことができます。取り出された状態情報は、転送の試行または <code>BPPIOC_TESTIO ioctl(2)</code> が発生するたびに書き換えられます。</p>
読み取り操作	<p><code>read(2)</code> の実行中に失敗、またはエラー条件が発生した場合には、正常に読み取られたバイト数が戻されます(不十分な読み取り)。利用可能なデータがないポートを読み取ろうとすると、<code>read(2)</code> は、<code>O_NDELAY</code> が設定されている場合には、<code>0</code> を返します。<code>O_NONBLOCK</code> が設定されている場合には、<code>read(2)</code> は、<code>-1</code> を返し、<code>errno</code> に <code>EAGAIN</code> を設定します。<code>O_NDELAY</code> と <code>O_NONBLOCK</code> が設定されていない場合、<code>read(2)</code> は、データが用意されるまで待ちます。</p>
ioctls	<p>以下の <code>ioctl(2)</code> 呼び出しを使用することができます。ecpp がデータを転送する場合、ドライバは、<code>ioctl(2)</code> 呼び出しが処理される前にデータがデバイスに送られるまで待機します。</p> <p>ecpp ドライバでは、<code>prnio(7I)</code> インタフェースが使用できます。</p> <p>注 - <code>PRNIOC_RESET</code> コマンドは、<code>nInit</code> シグナルを 2 ミリ秒間切り替えた後、デフォルトのネゴシエーションを行います。</p> <p>次の <code>ioctl(2)</code> 呼び出しは、下位互換性を維持するためにのみ残されています。そのため、新たに作成するアプリケーションでは使用しないことをお勧めします。</p> <p><b>ECPPIOC_GETPARMS</b>                    現在の転送パラメタを獲得します。引数は、<code>struct ecpp_transfer_parms</code> を指すポインタです。この構造体の要素については、以下を参照してください。デバイスが開かれてから設定されたパラメタがない場合には、この構造体にはデフォルトの値が設定されます。上記の「<a href="#">デフォルトの動作</a>」を参照してください。</p> <p><b>ECPPIOC_SETPARMS</b>                    転送パラメタを設定します。引数は、<code>struct ecpp_transfer_parms</code> を指すポインタです。設定されたパラメタが範囲外の場合には、<code>EINVAL</code> が返されます。周辺</p>

デバイスまたはホストデバイスが要求されたモードに対応していない場合は、EPROTONOSUPPORT が返されます。ecpp\_transfer\_parms とその有効なパラメタについては、以下を参照してください。

転送パラメタ構造体は、<sys/ecppio.h> で定義されています。

```
struct ecpp_transfer_parms {
    int write_timeout;
    int mode;
};
```

write\_timeout フィールドは、ecpp.conf に指定されている ecpp-transfer-timeout の値に設定されます。

write\_timeout フィールドは、転送要求に対して周辺デバイスが応答するまでドライバが待つ時間を指定します。この値は、0 より大きく、ECPP\_MAX\_TIMEOUT より小さく指定します。これ以外の値は、範囲外の値となります。

mode フィールドは、現在のパラレルポートが設定されている IEEE 1284 モードを反映します。このモードの値は、以下の ECPP\_CENTRONICS、ECPP\_COMPAT\_MODE、ECPP\_NIBBLE\_MODE、ECPP\_ECP\_MODE のいずれかに設定することができます。他のすべての値は無効です。要求されたモードに対応していない場合には、ECPPIOC\_SETPARMS は EPROTONOSUPPORT. を返します。またこのモードは、ECPP\_CENTRONICS モードに設定されたあと、ECPPIOC\_SETPARMS は、アプリケーションを元のモードに戻します。

#### ECPPIOC\_GETDEVID

この ioctl コマンドは、指定されたモードで周辺デバイスから IEEE 1284 デバイスの ID を獲得します。現在、デバイスの ID は、ニブルモードでしか獲得できません。<sys/ecppsys.h> で定義された構造体へのポインタは、引数として渡される必要があります。

1284 デバイスの ID の構造体は以下ようになります。

```
struct ecpp_device_id {
    int mode; /* デバイスの ID の読み取りに使用されるモード */
    int len; /* バッファの長さ */
    int rlen; /* デバイスの ID の文字列の実際の長さ */
    char *addr; /* バッファのアドレス */
};
```

モードは IEEE 1284 のモードのことで、このモードでデバイスの ID 情報を取り出すためのネゴシエーションがポートに対して行われます。周辺デバイスまたはホストがこのモードをサポートしていない場合、EPROTONOSUPPORT が返されます。アプリケーションは、モードを ECPP\_NIBBLE\_MODE に設定します。len は、addr が指すバッファの長さです。rlen は、周辺デバイスから返されるデバイスの ID 文字列の実際の長さを表します。返された rlen が len よりも長い場合、アプリケーションは、再度 ECPP\_IOCTL\_GETDEVID を呼び出すことができます。その際、バッファの長さは rlen と等しいかそれより長くなります。IEEE 1284 デバイスの ID の 2 バイトの長さは、考慮にはいれていないため、ユーザーのバッファでは返されないことに注意してください。

ECPP\_IOCTL\_GETDEVID の実行に成功したら、ドライバは ECPP\_COMPAT\_MODE にリンクを返します。アプリケーションは、リンクが機能していた以前のモードを判定し、そのモードにリンクを戻します。

#### BPPIOC\_TESTIO

セントロニクスまたは互換モードでの周辺デバイスの操作について、順方向転送の準備ができていないかをテストします。

TESTIO は、オープンフラグとセントロニクスのステータス信号を検査することにより周辺デバイスがデータを受け取る準備ができていないかを判定します。デバイスの現在のモードが、ECPP\_NIBBLE\_MODE の場合、ドライバは ECPP\_COMPAT\_MODE モードへの移行を試みて、ステータス信号を検査してから、ECPP\_NIBBLE\_MODE モードに戻します。現在のモードが、ECPP\_CENTRONICS または ECPP\_COMPAT\_MODE の場合、TESTIO は現在のモードでセントロニクスステータス信号を検査します。データを受け取るには、デバイスは、nErr と Select 信号は立っている必要がありますが、PE と Busy 信号は立っていることは許されません。ecpp がデータを転送する場合、TESTIO は、TESTIO を実行する前にドライバに送られた以前のデータが配信されるまで待機しています。しかし、TESTIO が待機している間にエラーが起こった場合、TESTIO は即座にエラーを返します。TESTIO が、状況が良いと判定した場合は、0 が返されます。しかし、-1 が返された場合は、errno が、EIO に設定され、状態ピンの状況が捕らえられます。この状況は、BPPIOC\_GETERR ioctl(2) 呼び出しにより取り出すことができます。timeout\_occurred と bus\_error フィールドは、この ioctl(2) により設定されることはありません。

BPPIOC\_TESTIO と BPPIOC\_GETERR は、bpp(7D) に指定されている ioctl と互換性があります。

#### BPPIOC\_GETERR

最後のエラー状態を取り出します。引数は、`<sys/bpp_io.h>` ヘッダーファイルに定義されている `struct bpp_error_status` を指すポインタです。エラーステータスの構造体は以下のようになります。

```
struct bpp_error_status {
    char    timeout_occurred; /* 1= タイムアウト */
    char    bus_error;        /* 使用されない */
    uchar_t pin_status;      /* エラーを起こす可能性の
                             /* あるピンの状態 */
};
```

`pin_status` フィールドには、起こりうるエラーの条件を示します。`pin_status` の有効なビットは、`BPP_ERR_ERR`、`BPP_SLCT_ERR`、`BPP_PE_ERR`、`BPP_BUSY_ERR` です。設定されたビットは、関連するピンが何であるかを示します。

この構造体は、`write(2)` 呼び出し時の最新のエラー条件のすべての適切な状態ビット、または最新の `BPPIOC_TESTIO ioctl(2)` 呼び出し時のビットの状態を示します。

`pin_status` は、`ECPP_CENTRONICS` または `ECPP_COMPAT_MODE` で発生する可能性があるエラー条件を示します。これらのモードでは、状態ピンは、デバイスの状態を示します。たとえば、多くのセントロニクスプリンタでは、紙詰まりの発生時に `nErr` の信号レベルを下げます。状態ピンの動作は、デバイスによって異なります。追加状態情報は、逆方向チャンネルを通して返されます。

`timeout_occurred` の値は、`write(2)` の呼び出し中に時間切れが発生した時に設定されます。`bus_error` は、このインタフェースでは使用されません。

#### ioctls

以下の ioctl は、パラレルポートのステータスを直接読み込んで書き込み、信号を制御するために使用されます。デバイスの現在のモードが、`ECPP_ECP_MODE` または `ECPP_NIBBLE_MODE` の場合、ドライバは `ECPP_COMPAT_MODE` にリンクするようにネゴシエーションを行い、レジスタの値を取得または設定し、リンクを `ECPP_NIBBLE_MODE` に戻します。現在のモードが、`ECPP_CENTRONICS` または `ECPP_COMPAT_MODE` の場合、これらの ioctl は現在のモードでレジスタの値を獲得または設定します。

#### ECPPIOC\_GETREGS

レジスタ値を読み取ります。引数は、`struct ecpp_regs` 構造体へのポインタです。この構造体については、以下の記述を参照してください。

**ECPPIOC\_SETREGS**

ecpp レジスタ値を設定します。引数は、`struct ecpp_regs` へのポインタです。この構造体については、以下の記述を参照してください。パラメータが範囲外の場合は、`EINVAL` が返されます。

ポートレジスタ構造体は、`<sys/ecppio.h>` に定義されています。

```
struct ecpp_regs {
    uchar    dsr; /* status reg */
    u_char   dcr; /* control reg */
};
```

ステータスレジスタは、読み取り専用です。`ECPPIOC_SETREGS ioctl` は、このレジスタには影響を与えません。`dsr` の有効なビット値は、`ECPP_nER`、`ECPP_SLCT`、`ECPP_PE`、`ECPP_nACK`、`ECPP_nBUSY` です。ほかのすべてのビットは、予約されていて、常に 1 を返します。

制御レジスタは、読み取りと書き取りができます。`dcr` の有効なビット値は、`ECPP_STB`、`ECPP_AFX`、`ECPP_nINIT`、`ECPP_SLCTIN` です。ほかのすべてのビットは予約されています。予約されているビットを読み取ると、常に 1 を返します。これらのビットに 0 を書き込もうとすると、`EINVAL` が返されます。

**デバイスの特殊ファイル**

`/dev/lpN`

Solaris x86 のみ (以前の `lp(7D)` デバイスとの下位互換性のため)。

**ファイル**

<code>kernel/drv/ecpp</code>	32 ビット ELF カーネルモジュール
<code>kernel/drv/sparcv9/ecpp</code>	64 ビット SPARC ELF カーネルモジュール
<code>kernel/drv/amd64/ecpp</code>	64 ビット x86 ELF カーネルモジュール
<code>kernel/drv/ecpp.conf</code>	ドライバ構成ファイル
<code>kernel/drv/sparcv9/ecpp.conf</code>	64 ビット SPARC 用ドライバ構成ファイル
<code>kernel/drv/amd64/ecpp.conf</code>	64 ビット x86 用ドライバ構成ファイル

**エラー**

<b>EBADF</b>	書き込み専用で開いたデバイスに対して、読み取りが行われようとなりました。または、読み取り専用で開いたデバイスに対して、書き込みが行われようとなりました。
<b>EBUSY</b>	すでに開かれているデバイスが開かれようとなりました。デバイスの 1 つが開いているときに、ドライバを読み込み解除しようとした。
<b>EINVAL</b>	<code>ecpp_transfer_params</code> 構造体の中で <code>ECPPIOC_SETPARMS ioctl()</code> が範囲外の値で呼び出されました。 <code>ecpp_regs</code> 構造体の中で <code>ECPPIOC_SETREGS ioctl()</code> が無効な値で呼び出されました。 <code>modload(1M)</code> または <code>modunload(1M)</code> の実行中に、無効なコマンド引数を受け取りました。

- EIO**                    アクセス時にドライバがバスエラーを検出しました。読み取り処理または書き込み処理が、周辺デバイスのエラーまたは転送の時間切れのために、正常に終了しませんでした。
- ENXIO**                  ドライバが、接続ができないデバイスを開く要求を受け付けました。ドライバが、障害のある周辺デバイスに対する書き込み処理を受け取りました。

**属性**                    以下の属性については、`attributes(5)`を参照してください。

属性タイプ	属性値
アーキテクチャ	PCI ベースのシステムに限定
	ISA ベースのシステムに限定 (x86)
使用条件	SUNWpd (Sparc)
	SUNWpsdcr (x86)
インタフェースの安定性	開発中

**関連項目**              `modload(1M)`, `modunload(1M)`, `update_drv(1M)`, `ioctl(2)`, `open(2)`, `read(2)`, `write(2)`, `attributes(5)`, `bpp(7D)`, `usbprn(7D)`, `prnio(7I)`, `streamio(7I)`

IEEE Std 1284-1994

**診断**                    パラレルポートのコントローラが使用できない  
                              ドライバは、指定されたホストでパラレルポートのコントローラをサポートしていません。接続が失敗しました。

名前	fas – FAS SCSI ホストバスアダプタドライバ
形式	fas@sbus-slot,0x8800000
機能説明	<p>fas ホストバスアダプタドライバは、Qlogic FAS366 SCSI チップに対応した SCSI 準拠のネクサスドライバです。</p> <p>fas ドライバは、SCSA インタフェースが提供する標準機能に対応しています。ドライバは、タグ付きおよびタグなしの待ち行列化、Fast/Wide SCSI、ほとんど制限のない転送サイズ(移動 DVMA ウィンドウ法による)、自動要求検知に対応しています。ただし、リンクされたコマンドには対応していません。</p>
ドライバの設定	<p>fas ドライバは、広域的な SCSI 設定に優先する属性を fas.conf に定義することによって設定することができます。使用することができる属性は、scsi-options、target&lt;n&gt;-scsi-options、scsi-reset-delay、scsi-watchdog-tick、scsi-tag-age-limit、scsi-initiator-id です。</p> <p>target&lt;n&gt;-scsi-options は target&lt;n&gt; の scsi-options 属性に優先します。&lt;n&gt; には、10 進数の 0 ~ 15 の値を指定することができます。使用することができる scsi オプションは、SCSI_OPTIONS_DR、SCSI_OPTIONS_SYNC、SCSI_OPTIONS_TAG、SCSI_OPTIONS_FAST、SCSI_OPTIONS_WIDE です。</p> <p>周期的な時間間隔である scsi-watchdog-tick が経過する度に、fas ドライバは、現在実行中および切り離されたすべてのコマンドを調べ、時間切れかどうかを確認します。</p> <p>scsi-tag-age-limit は、fas ドライバがすべてのタグ ID を一巡して調べた後に、現在使用されている特定のタグ ID の割り当てを試みる回数です。同一のタグ ID が scsi-tag-age-limit の回数だけ使用されると、すべての未処理のコマンドが完了するか、時間切れになるまで、このターゲットにコマンドは発行されません。</p> <p>詳細については、scsi_hba_attach(9F) を参照してください。</p>
使用例	<p>例 1 fas 設定ファイルの例</p> <p>ファイル /kernel/drv/fas.conf を作成して、以下の行を追加してください。</p> <pre>scsi-options=0x78;</pre> <p>これは、すべての fas インスタンスに対して、タグ付き待ち行列化、Fast SCSI、Wide モードを無効にします。以下の例は、1 つの特定の fas のオプションを無効にします。詳細については、(driver.conf(4) を参照してください。</p> <pre>name="fas" parent="/iommu@f,e0000000/sbus@f,e001000"     reg=3,0x8800000,0x10,3,0x8810000,0x40     target1-scsi-options=0x58     scsi-options=0x178 scsi-initiator-id=6;</pre> <p>OBP のデフォルトのイニシエータ ID は 7 ですが、接続時に 6 に変更されます。OBP のイニシエータ ID を変更してください。</p>

上記の例では、ターゲット1の `scsi-options` を `0x58` に設定し、このSCSIバス上にある他のすべてのターゲットは `0x178` に設定しています。

親の物理バス名は、`/devices` ツリーを使用するか、論理デバイス名のリンクをたどることによって判定することができます。

```
# ls -l /dev/rdisk/c1t3d0s0
lrwxrwxrwx 1 root  other  78 Aug 28 16:05 /dev/rdisk/c1t3d0s0 ->
```

```
.. /.. /devices/iommu@f,e0000000/sbus@f,e0001000/SUNW,fas@3,88000000/sd@3,0:a,raw
```

レジスタの属性値は、`prtconf(1M)` の出力 (`-v` オプションを使用) によって判定されます。

```
SUNW,fas, instance #0
```

```
.....
```

```
Register Specifications:
```

```
Bus Type=0x3, Address=0x88000000, Size=10
```

```
Bus Type=0x3, Address=0x88100000, Size=40
```

また、デバイス照会文字列を使用することによって、`scsi-options` をデバイスごとに指定することができます。同一の照会文字列を持つすべてのデバイスには、同一の `scsi-options` が設定されます。これを利用して、同種のすべてのデバイスに対して、いくつかの `scsi-options` を無効にすることができます。

```
device-type-scsi-options-list=
    "TOSHIBA    XM5701TASUN12XCD", "cd-scsi-options";
cd-scsi-options = 0x0;
```

`/kernel/drv/fas.conf` の中の上記のエントリは、照会文字列が `TOSHIBA XM5701TASUN12XCD` のすべてのデバイスの `scsi-options` を `cd-scsi-options` に設定します。照会文字列を調べるには、システムを起動する前に `ok` プロンプトに対して `probe-scsi` または `probe-scsi-all` コマンドを実行します。

`scsi-options` をターゲットごとに設定するには、以下のように入力します。

```
target1-scsi-options=0x78;
device-type-scsi-options-list =
    "SEAGATE ST32550W", "seagate-scsi-options" ;
seagate-scsi-options = 0x58;
scsi-options=0x3f8;
```

この例は、ターゲット1の `scsi-options` を `0x78` に設定し、ある特定の種類のディスクの `scsi-options` を `0x58` に設定し、このSCSIバス上にある他のすべてのターゲットの `scsi-options` を `0x3f8` に設定します。



ターゲットIDごとに指定された `scsi-options` は、最も優先順位が高くなります。その次に優先順位が高いのは、デバイスの種類ごとに指定された `scsi-options` です。バスごとに指定される広域的な `fas scsi-options` (すべてのインスタンスに影響します) は、最も優先順位が低くなります。

指定された `scsi-options` が有効になるには、システムを再起動する必要があります。

#### ドライバの機能

ターゲットドライバがドライバの機能を有効にするには、`fas` ドライバの機能を設定する必要があります。ターゲットドライバは、これらの機能の照会および変更ができます。照会および変更ができる機能には、`synchronous`、`tagged-qing`、`wide-xfer`、`auto-rqsense`、`qfull-retries`、`qfull-retry-interval` があります。他のすべての機能は、照会のみ行うことができます。

デフォルトでは、`tagged-qing`、`auto-rqsense`、`wide-xfer` 機能は無効で、`disconnect`、`synchronous`、`untagged-qing` 機能は有効です。これらの機能には、2進数の値 (0 または 1) のみを設定することができます。 `qfull-retries` のデフォルト値は 10 で、`qfull-retry-interval` のデフォルト値は 100 です。 `qfull-retries` 機能は、`uchar_t` 型 (0 ~ 255) で、`qfull-retry-interval` は、`u_short` 型 (0 ~ 65535) です。

ターゲットドライバは、`tagged-qing` および `wide-xfer` を明示的に有効にする必要があります。 `tagged-qing` が無効になっていても、`fas` がコマンドを待ち行列に入れることができるため、`untagged-qing` 機能は常に有効で、その値を変更することはできません。

`scsi-options` の値と機能が矛盾している場合は、`scsi-options` に設定されている値が常に優先されます。 `scsi_ifsetcap(9F)` 呼び出しでは、`whom != 0` のみを使用することができます。

詳細については、`scsi_ifsetcap(9F)` および `scsi_ifgetcap(9F)` を参照してください。

#### ファイル

`/kernel/drv/fas`                    ELF カーネルモジュール  
`/kernel/drv/fas.conf`            省略可能な設定ファイル

#### 属性

以下の属性については、`attributes(5)` を参照してください。

属性の種類	属性の値
Architecture	FAS366 ベースの SCSI ポートおよび SunSWIFT SBus SCSI ホストアダプタ/Fast Ethernet を持つ SPARC SBus ベースのシステムに限定

#### 関連項目

`prtconf(1M)`, `driver.conf(4)`, `attributes(5)`, `scsi_abort(9F)`, `scsi_hba_attach(9F)`, `scsi_ifgetcap(9F)`, `scsi_ifsetcap(9F)`, `scsi_reset(9F)`, `scsi_sync_pkt(9F)`, `scsi_transport(9F)`, `scsi_device(9S)`, `scsi_extended_sense(9S)`, `scsi_inquiry(9S)`, `scsi_pkt(9S)`

『Writing Device Drivers』

『ANSI Small Computer System Interface-2 (SCSI-2)』

『FAS366 Technical Manuals』 (QLogic Corporation 刊)

## 診断

以下で説明するメッセージは、コンソールに表示されると同時にログファイルに記録されます。

最初の5つは、fas ドライバの接続処理中に表示されるメッセージです。これらの5つのメッセージはすべて、fas ドライバが接続処理に失敗したことを意味します。これらのメッセージの前に、「fas%d」が表示されます。「%d」は、fas コントローラのインスタンス番号です。

### Device in slave-only slot, unused

SBus デバイスがスレーブ専用のスロットに配置されたため、アクセスすることができません。SBus デバイスをスレーブ以外のスロットに移動してください。

### Device is using a hilevel intr

fas ドライバが使用することができない割り込みレベルでデバイスが設定されています。SBus デバイスを調べてください。

### Cannot alloc dma handle

ドライバが DMA コントローラに対して記憶域を割り当てることができません。

### Cannot alloc cmd area

ドライバがコマンドアドレスに対して記憶域を割り当てることができません。

### Cannot create kmem\_cache

ドライバが内部データ構造に対して記憶域を割り当てることができません。

### Unable to map FAS366 registers

ドライバがデバイスレジスタを割り当てることができません。問題のあるハードウェアを調べてください。ドライバがデバイスに接続することができないために、SCSI デバイスにアクセスすることができません。

### Cannot add intr

ドライバが割り込みサービ斯拉ーチンをカーネルに追加することができません。

### Cannot map dma

ドライバは、DMA コントローラを検出することができません。自動構成エラーです。

### Cannot bind cmdarea

ドライバがアドレスに DMA ハンドルを割り当てることができません。

### Cannot create devctl minor node

ドライバがコントローラのマイナーノードを作成することができません。

### Cannot attach

ドライバが接続処理に失敗しました。通常、接続に失敗した原因を示す警告メッセージが後に続きます。

**Disabled TQ since disconnects are disabled**

scsi-options で切り離しが無効にされているため、タグ付きの待ち行列化が無効となりました。

**Bad clock frequency**

問題のあるハードウェアを調べてください。

**Sync of pkt (<address>) failed**

scsi パケットの同期処理が失敗しました。scsi\_sync\_pkt(9F) を参照してください。

**All tags in use!**

ドライバが別のタブ番号を割り当てられませんでした。ターゲットデバイスが、タグ付き待ち行列化に正しく対応していません。

**Gross error in FAS366 status**

ドライバが、重大な SCSI バス障害を検知しました。ケーブルとターミネータを調べてください。

**Spurious interrupt**

ドライバが、ハードウェアの割り込みが発生していないのに割り込みを受信しました。

**Lost state in phasemanage**

ドライバが、SCSI バスの状態を正しく判断することができませんでした。

**Unrecoverable DMA error duringselection**

DMA がホスト SBus の障害を検知しました。問題のあるハードウェアを調べてください。

**Bad sequence step (<step number>) in selection**

FAS366 ハードウェアが問題のある処理手順を報告しました。問題のあるハードウェアを調べてください。

**Undetermined selection failure**

ターゲットの選択が予想されたとおりになりません。問題のあるハードウェアを調べてください。

**Target <n>:failed reselection (bad reselect bytes)**

再接続処理が失敗しました。ターゲットが間違った長さのメッセージバイトを送信しました。問題のあるハードウェアを調べてください。

**Target <n>:failed reselection (bad identify message)**

再接続処理が失敗しました。ターゲットが識別メッセージを送信しなかったか、メッセージが壊れています。

**Target <n>:failed reselection (not in msgin phase)**

再接続処理後の不正な SCSI バスフェーズ。問題のあるハードウェアを調べてください。

**Target <n>:failed reselection (unexpected bus free)**

再接続処理後の不正な SCSI バスフェーズ。問題のあるハードウェアを調べてください。

**Target <n>: failed reselection (timeout on receivingtag msg)**

再接続処理が失敗しました。ターゲットは、タグバイトの送信に失敗しました。問題のあるハードウェアを調べてください。

**Target <n>: failed reselection (botched tag)**

再接続処理が失敗しました。ターゲットは、タグバイトの送信に失敗しました。問題のあるハードウェアを調べてください。

**Target <n>: failed reselection (invalid tag)**

再接続処理が失敗しました。ターゲットは、間違ったタグバイトを送信しました。問題のあるハードウェアを調べてください。

**Target <n>: failed reselection (Party error in reconnectmsg's)**

再接続処理が失敗しました。パリティエラーを検出しました。問題のあるハードウェアを調べてください。

**Target <n>: failed reselection (no command)**

再接続処理が失敗しました。ターゲットは、中止またはリセットを受け付けましたが、再接続を試みます。問題のあるハードウェアを調べてください。

**Unexpected bus free**

通知なしに、ターゲットがバスから切り離されました。問題のあるハードウェアを調べてください。

**Target <n> didn't disconnect after sending <message>**

ターゲットが、<message> を送信した後、予想されたとおりに切り離されませんでした。

**Bad sequence step (0x?) in selection**

処理手順レジスタが不適切な値を示しています。ターゲットの動作が正しくない可能性があります。

**Illegal dma boundary?**

ドライバが、処理できない境界にアクセスしようとしてしました。

**Unwanted data xfer direction for Target <n>**

ターゲットが、予想されていないフェーズに入りました。

**Unrecoverable DMA error on dma <send/receive>**

データを送信・受信している際に DMA エラーがあります。ホスト DMA コントローラが問題をいくつか検知しました。

**SCSI bus DATA IN phase parity error**

ドライバが、SCSI バスでパリティエラーを検出しました。

**SCSI bus MESSAGE IN phase parity error**

ドライバが、SCSI バスでパリティエラーを検出しました。

**SCSI bus STATUS phase parity error**

ドライバが、SCSI バスでパリティエラーを検出しました。

**Premature end of extended message**

拡張 SCSI バスメッセージが終了しませんでした。ターゲットのファームウェアを調べてください。

**Premature end of input message**

複数バイトの入力メッセージが切り捨てられました。ターゲットのファームウェアを調べてください。

**Input message botch**

ドライバが、ターゲットからのメッセージを判断することができませんでした。

**Extended message <n> is too long**

ターゲットが送信した拡張メッセージが予想されていたものより長いものでした。

**<name> message <n> from Target <m> garbled**

ターゲット <m> が、ドライバが理解することができない値 <n> のメッセージ <name> を送信しました。

**Target <n> rejects our message <name>**

ターゲット <n> が、ドライバが送信したメッセージを拒否しました。

**Rejecting message <name> from Target <n>**

ドライバが、ターゲットから受信したメッセージを拒否しました。

**Cmd transmission error**

ドライバは、コマンドバイトを送信することができませんでした。

**Target <n> refused message resend**

ターゲットは、再送されたメッセージを受け付けませんでした。

**MESSAGE OUT phase parity error**

ドライバが SCSI バス上でパリティエラーを検出しました。

**Two bytes message <name> <value> rejected**

ドライバは、この 2 バイトメッセージを受け付けませんでした。

**Gross error in fas status <stat>**

fas チップが FIFO オーバーフローなどの重大なエラーを示しました。

**Polled cmd failed (target busy)**

ポーリングされたコマンドが失敗しました。これは、ターゲットが未処理のコマンドを妥当な時間内に終了することができなかったためです。

**Polled cmd failed**

ポーリングされたコマンドが時間切れまたはバスエラーのために失敗しました。

**Auto request sense failed**

ドライバがターゲットからの要求検知を取得することができません。

**Disconnect command timeout for Target <id>.<lun>**

ターゲット *id/lun* が切り離されている間に時間切れが発生しました。これは、通常、ターゲットのファームウェアの問題によるものです。タグ付きの待ち行列化のターゲットでは、<*n*> コマンドは時間切れが検出された時点でまだ処理が終了していませんでした。

**Disconnected tagged cmds (<n>) timeout for Target <id>.<lun>**

ターゲットまたは、*lun* が切り離されている間に時間切れが発生しました。これは、通常、ターゲットのファームウェアの問題によるものです。タグ付きの待ち行列化のターゲットでは、<*n*> コマンドは時間切れが検出された時点でまだ処理が終了していませんでした。

**Connected command timeout for Target <id>.<lun>**

これは、通常、SCSI バスの問題です。ケーブルと終端を調べてください。

**Target <id>.<lun> reverting to async. mode**

データ転送の停滞が検出されました。ドライバは、データ転送速度を落としてこの問題を回避します。

**Target <id>.<lun> reducing sync. transfer rate**

データ転送の停滞が検出されました。ドライバは、データ転送速度を落としてこの問題を回避します。

**Reverting to slow SCSI cable mode**

データ転送の停滞が検出されました。ドライバは、データ転送速度を落としてこの問題を回避します。

**Target <id> reducing sync. transfer rate**

データ転送の停滞が検出されました。ドライバは、データ転送速度を落としてこの問題を回避します。

**Target <id> reverting to async. mode**

データ転送の停滞が検出されました。ドライバは、データ転送速度を落としてこの問題を回避します。

**Target <id> disabled wide SCSI mode**

SCSI バス上での問題がありました。これ以上の問題を避けるために、ドライバはより保守的な動作モードに入ります。

**Reset SCSI bus failed**

SCSIバスの再設定が失敗しました。

**External SCSI bus reset**

別のインシエータが SCSI バスを再設定しました。

**警告**

*fasfas* ハードウェア (FAS366) は、Fast/Wide SCSI モードに対応していますが、Fast20 には対応していません。最大 SCSI 帯域幅は、20 MB/秒です。インシエータモードブロック処理 (IBS) には対応していません。

**注意事項**

*fas* ドライバは、ネゴシエートされた転送速度 (*target<n>-sync-speed*)、Wide バスに対応しているかどうか (*target<n>-wide*)、特定のターゲットの *scsi-options* (*target<n>-scsi-options*)、タグ付き待ち行列化が有効であるかどうか (*target<n>-TQ*)

をターゲットごとに示す属性をエクスポートします。sync-speedの属性値は、KB/秒単位のデータ転送速度です。target<n>-TQおよびtarget<n>-wide属性は、ターゲットに対して対応する機能が有効であれば1を、無効であれば0の値を取ります。fasの属性を表示する場合は、prtconf(1M)(表示オプション)を参照してください。

SUNW,fas, instance #1

Driver software properties:

```

name <target3-TQ> length <4>
  value <0x00000001>.
name <target3-wide> length <4>
  value <0x00000000>.
name <target3-sync-speed> length <4>
  value <0x00002710>.
name <target3-scsi-options> length <4>
  value <0x000003f8>.
name <target0-TQ> length <4>
  value <0x00000001>.
name <pm_norm_pwr> length <4>
  value <0x00000001>.
name <pm_timestamp> length <4>
  value <0x30040346>.
name <scsi-options> length <4>
  value <0x000003f8>.
name <scsi-watchdog-tick> length <4>
  value <0x0000000a>.
name <scsi-tag-age-limit> length <4>
  value <0x00000002>.
name <scsi-reset-delay> length <4>
  value <0x00000bb8>.

```

Register Specifications:

```

Bus Type=0x3, Address=0x8800000, Size=10
Bus Type=0x3, Address=0x8810000, Size=40

```

Interrupt Specifications:

```

Interrupt Priority=0x35 (ipl 5)

```

---

名前	ffb - 24 ビット UPA カラーフレームバッファとグラフィックスアクセラレータ
機能説明	<p>ffb は 24 ビット UPA ベースのカラーフレームバッファとグラフィックスアクセラレータです。ffb は、シングルバッファのフレームとダブルバッファのフレームの 2 つの構成で提供されます。</p> <p>シングルバッファのフレーム                      バッファ 32 の 1280 × 1024 ピクセルの表示メモリー平面によって構成されます。これには、24 ビットシングルバッファリングと 8 ビット X 平面が含まれます。</p> <p>ダブルバッファのフレームバッファ              32 の 1280 × 1024 ピクセルの表示メモリー平面によって構成されます。これには、24 ビットダブルバッファリング、8 ビット X 平面、28 ビット Z バッファ平面、4 ビット Y 平面が含まれます。</p> <p>ffb ドライバは、fbio(7I) で定義されている以下のフレームバッファ ioctl に対応しています。</p> <p>FBIOPUTCMAP、FBIOGETCMAP、FBIOSVIDEO、FBIOGVIDEO、FBIOVERTICAL、FBIOSCURLOR、FBIOGCURLOR、FBIOSCURPOS、FBIOGCURPOS、FBIOGCURMAX、FBIOWID_PUT、FBIOWID_GET</p>
ファイル	/dev/fbs/ffb0              デバイス特殊ファイル
関連項目	ffbconfig(1M), mmap(2), fbio(7I)



名前	ge - GEM GigabitEthernet デバイスドライバ
形式	/dev/ge
機能説明	<p>GigabitEthernet ドライバ (ge ドライバ) は、マルチスレッド化され、読み込みや複製が可能な STREAMS ハードウェアドライバです。GEM、SBus、および PCI GigabitEthernet アドイン型アダプタに対してコネクションレス型のデータリンクプロバイダインタフェース <code>dlpi(7P)</code> をサポートします。ge ドライバは、システムに組み込まれた、複数の GEM アダプタをサポートするものです。、ge ドライバは、おもに GEM 対応の Ethernet ハードウェアをサポートし、SUNW, sbus-gem (SBus 対応 GEM) や pci108e, 2bad (PCI 対応 GEM) などのデバイスを制御します。その機能は、チップの初期化、フレームの転送と受信、マルチキャストとプロミスキュアスのサポート、エラーの復旧処理および報告です。</p> <p>GEM デバイスは、外部 SERDES および光ファイバー通信トランシーバを制御する GEM ASIC (専用チップ) を使用して、100BASE-SX のネットワークインタフェースを提供します。GEM ASIC は、適切なバスインタフェースに対し、MAC 機能や PCS (Physical Code Sub-layer: 物理コードサブレイヤー) 機能を提供します。外部 SERDES は、光ファイバートランシーバに接続され、物理的な接続を提供します。</p> <p>100BASE-SX 規格には、自動的に動作モードを選択する自動ネゴシエーション機能が規定されています。さらに全二重モードのために、GEM ASIC は、フロー制御に準拠した自動ネゴシエーション機能を実現しています。GEM PCS 機能は、遠隔リンク先 (リンクパートナー) に対して、自動ネゴシエーションを実行したり、遠隔リンク先からのメッセージを受信したりします。また、優先度に応じてもっとも優先順位の高いモードを選択します。ドライバが対応しているシステムでは、強制モードもサポートしません。</p>
アプリケーションプログラミングインタフェース	<p>文字複製された特殊デバイスである /dev/ge は、システムに組み込まれている、すべての ge コントローラへのアクセスに使用されます。</p> <p>ge ドライバは、スタイル 2 型のデータリンクサービスを提供するものです。M_PROTO 型、および M_PCPROTO 型のメッセージは、DLPI プリミティブとして解釈されます。DLPI プリミティブについては、<code>&lt;sys/dlpi.h&gt;</code> ファイルで定義されています。詳細については、<code>dlpi(7P)</code> のマニュアルページを参照してください。</p> <p>ユーザーからの <code>DL_ATTACH_REQ</code> メッセージは、特殊デバイス /ppa に対して開かれたストリームに関連付けられなければなりません。ppa ID は、<code>unsigned long</code> 型のデータとして解釈され、対応するデバイスインスタンス (ユニット) 番号を指し示します。ppa フィールドの値が、そのシステムの有効なデバイスインスタンス番号と一致しない場合、ドライバは、エラー値とし <code>DL_ERROR_ACK</code> を返します。デバイスは、最初の接続時に初期化され、最後の接続時に終了処理 (停止) されます。</p> <p><code>DL_INFO_REQ</code> に対してドライバから換えされる <code>DL_INFO_ACK</code> の値は、以下のようになります。</p> <ul style="list-style-type: none"> <li>■ SDU の最大値は 1500 です (<code>ETHERMTU -&lt;sys/ethernet.h&gt;</code> で定義されている)。</li> </ul>

- SDU の最小値は 0 です。
- dlsap のアドレス長は 8 です。
- MAC 型は DL\_ETHER です。
- sap 長の値は -2 です。これは、物理アドレスコンポーネントの直後に、DLSAP アドレス内部にある、2 バイトの sap コンポーネントが続くことを意味しています。
- サービスモードは DL\_CLDLS です。
- オプションなしのサービス品質 (QOS) サポートしない場合、QOS フィールドの値は 0 になります。
- サービス提供の形式は DL\_STYLE2 です。
- バージョンは DL\_VERSION\_2 です。
- ブロードキャストアドレスの値は、Ethernet/IEEE ブロードキャストアドレス (0xFFFFF) です。

DL\_ATTACHED 状態において特殊な Service Access Pointer (SAP) をストリームに関連付けるために、ユーザーは DL\_BIND\_REQ を送信する必要があります。フィールドの値が [0-0xFFFF] の範囲内にある限り、ドライバは、DL\_BIND\_REQ 中の sap フィールドを Ethernet の type (種類) として解釈します。いかなる場合も、ただ 1 つの種類の Ethernet をストリームに結合することができます。

ユーザーが sap の値として 0 を指定した場合、受け取り側は、802.3 モードになります。type フィールドの値が [0-1500] の範囲内にあるメディアを受信したすべてのフレームは、802.3 準拠とみなされます。そして、fsap の値が 0 に結合されている、開いているすべてのストリームに経路付けられます。1 つ以上のストリームがであるなら、DL\_UNITDATA\_IND メッセージとして、複数のストリームに経路付けられます。

データ伝送中、ge ドライバは、DL\_BIND\_REQ 中の sap 値を検査して、sap が 0 かどうか、および、destination type フィールドの値が [0-1500] の範囲内にあるかどうかを判断します。もし、いずれの条件も真の場合、ドライバは、すべての DL\_UNITDATA\_REQ メッセージの直後に続く最初の M\_PROTO mblk (メッセージブロック) を含めずに、メッセージの長さを算出します。そして、この値を MAC フレームヘッダ長フィールドに持つ 802.3 フレームを送信します。

ge ドライバの DLSAP アドレスの書式は、8 バイトの DLSAP アドレスを生成する、2 バイトの sap (型) コンポーネントと、それに続く (Ethernet の) 6 バイトの物理アドレスコンポーネントから成り立っています。アプリケーションは、この特殊な実装を指定する DLSAP アドレスの書式をハードコードすべきではありません。アドレスの構成と分解を行う DL\_INFO\_ACK プリミティブに返される情報を使用すべきです。sap のデータ長、DLSAP の全データ長、sap または物理的な指示は、DL\_INFO\_ACK の中に含まれています。物理アドレス長は、全 DLSAP アドレス長から sap の値を弾くか、ストリームに関連付けられた現在の物理アドレスを得る、DL\_PHYS\_ADDR\_REQ を発行することにより、算出できます。

DL\_BOUND ステートでは、sap ドライバに DL\_UNITDATA\_REQ メッセージを送ることで、ユーザーは Ethernet 上にフレームを送信することができます。ge ドライバは、

DL\_UNITDATA\_IND メッセージに Ethernet タイプが一致している sap を持つ、開かれた、もしくは結合されたストリームに、受信した Ethernet フレームを経路付けることとなります。必要に応じて、受信された Ethernet フレームは複製され、複数の開かれたストリームに経路付けされます。DL\_UNITDATA\_REQ および DL\_UNITDATA\_IND メッセージの中に含まれる DLSAP アドレスは、sap (型) と物理 (Ethernet) コンポーネントから構成されています。

強制コネクションレス DLPI メッセージセットに加えて、ge ドライバは、以下のプリミティブをサポートしています。

#### ge プリミティブ

DL\_ENABMULTI\_REQ と DL\_DISABMULTI\_REQ プリミティブは、個々のマルチキャストグループアドレスの受け取りを許可・不許可します。この2つのプリミティブを使うことによって、マルチキャストアドレスの組み合わせが各ストリーム上で何度も生成されたり、変更されたりします。ドライバは、DL\_ATTACHED に続くいかなる状態においても、DL\_ENABMULTI\_REQ と DL\_DISABMULTI\_REQ のプリミティブを受け入れます。

dL\_level フィールドの DL\_PROMISC\_PHYS フラグセットによって、DL\_PROMISCON\_REQ と DL\_PROMISCOFF\_REQ プリミティブは、ローカルホストの生成したフレームを含むメディアにある、すべてのプロミスキュアモードフレームの受け入れを許可・不許可します。DL\_PROMISC\_SAP フラグセットとともに使用されるとき、この機能は、すべての sap (Ethernet の種類を表す) の値の受け入れを許可・不許可します。DL\_PROMISC\_MULTI フラグセットとともに使用されるとき、この機能は、すべてのマルチキャストグループアドレスの受け入れを許可・不許可します。それぞれの効果は、常にそれぞれのストリームの土台となり、該当するストリーム、または他のストリームの、他の sap および物理レベルの設定に依存します。

DL\_PHYS\_ADDR\_REQ プリミティブは、DL\_PHYS\_ADDR\_ACK プリミティブ中のストリームにその時点で関連 (結合) 付けられた 6 オクテットの Ethernet アドレスを返します。このプリミティブは、成功した DL\_ATTACH\_REQ の後に続く状態であるときのみ有効です。

DL\_SET\_PHYS\_ADDR\_REQ プリミティブは、このストリームにその時点で関連 (結合) 付けられた 6 オクテットの Ethernet アドレスを変更します。最初にこのストリームを開いたプロセスの資格 (credentials) は、スーパーユーザでなければなりません。そうでない場合、DL\_ERROR\_ACK 内に EPERM が換えされます。DL\_SET\_PHYS\_ADDR\_REQ プリミティブは、このデバイスに結合された、現在および未来の他のすべてのストリームに影響を及ぼす点で、破壊的なものです。DL\_SET\_PHYS\_ADDR\_REQ DL\_SET\_PHYS\_ADDR\_REQ プリミティブがこのストリーム上で成功したとき、このデバイスに結合された他のすべてのストリームに M\_ERROR が送られます。いったん変更があると、その後に関閉、このデバイスに結合されたすべてのストリームは、この新しい物理アドレスを取得するはずですが、変更があったとき、この物理アドレスは、DL\_SET\_PHYS\_ADDR\_REQ によって物理アドレスが再度変更されるか、システムが再起動されるまで保持されます。

#### ge ドライバ

デフォルトでは、ge ドライバは自動ネゴシエーションを実行して、リンクのモードとフロー制御モードを選択します。リンクは、以下に示す 4 つのモードのいずれかになります。

- 1000 Mbps、全二重接続

- 1000 Mbps、半二重接続
- 対称的中断 (Symmetric Pause)
- 非対称的中断 (Asymmetric Pause)

通信速度とモードは、1000BASE-TX 規格に規定されています。

自動ネゴシエーションプロトコルは、以下の項目を自動的に選択します。

- 動作モード (半二重または全二重)
- フロー制御機能 (対称または非対称、あるいはその両方)

自動ネゴシエーションプロトコルは、以下の処理を行います。

- リンク先パートナーのサポートする全動作モードの取得
- リンク先パートナーに対する、各機能を通知
- 優先度に基づいた、最も優先度の高いモードの選択

GEM ハードウェアは、上記の全動作モードを実現しています。デフォルト時は、リンク先との接続のために自動ネゴシエーションを使用し、次に、リンク先パートナーと同じ動作モードを選択します。PCS は、強制モードもサポートしています。これは、nnd ユーティリティを使用して、ドライバが動作モードとフロー制御機能を選択できるモードです。

GEM デバイスは、プログラム可能な Inter-Packet Gap (IPG) パラメータの `ipg1` および `ipg2` をサポートしています。デフォルトでは、ドライバは、`ipg1` を 8 バイトの倍数に、`ipg2` を 4 バイトの倍数に設定します (これらは標準的な数値です)。ユーザーは、これらの数値を標準的な 1000 Mbps IPG セットから 0.096 マイクロ秒へ変更してもかまいません。

#### ge のパラメータリスト

ge ドライバを使用して、デバイスの各種パラメータを設定したり、取得することができます。パラメータのリストには、現在のトランシーバの状態、現在の接続の状態、パケット間隔、PCS の各機能、およびリンク先パートナーの各機能が含まれています。

PCS には、機能別に 2 つのセットがあります。1 つは、ハードウェアの各機能を反映したものであり、読み取り専用です。2 つめのセットは読み取り/書き込み可能で、高速選択 (speed selection) として使用され、ユーザーが選択した数値を反映したものです。ブート時には、これらの 2 つの機能セットは同じになります。リンク先パートナーの各機能は読み取り専用で、変更できません。

#### ファイル

`/dev/ge`

ge 文字型特殊デバイス

`/kernel/drv/ge.conf`

システム全般のデフォルトデバイスドライバ属性

#### 関連項目

`nnd(1M)`, `netstat(1M)`, `driver.conf(4)`, `dlpi(7P)`, `hme(7D)`, `qfe(7D)`

名前	glm – GLM SCSI ホストバスアダプタドライバ
形式	scsi@unit-address
機能説明	<p>glm ホストバスアダプタドライバは、LSI 53c810、LSI 53c875、LSI 53c876、LSI 53C896、および LSI 53C1010 SCSI チップに対応した SCSI 準拠のネクサスドライバです。</p> <p>glm ドライバは、SCSA インタフェースが提供する標準機能に対応しています。ドライバは、タグ付きおよびタグなしの待ち行列化、Narrow/Wide/Fast/Ultra SCSI/Ultra SCSI 2/Ultra SCSI 3、自動要求検知に対応しています。ただし、リンクされたコマンドには対応していません。</p>
ドライバの設定	<p>glm ドライバは、広域的な SCSI 設定に優先する属性を glm.conf に定義することによって設定することができます。glm で使用・変更することができる属性は、scsi-options、target&lt;n&gt;-scsi-options、scsi-reset-delay、scsi-tag-age-limit、scsi-watchdog-tick、および scsi-initiator-id です。</p> <p>target&lt;n&gt;-scsi-options は target&lt;n&gt; の scsi-options 属性に優先します。&lt;n&gt; には、10 進数の 0 ~ 15 の値を指定することができます。使用することができる scsi オプションは、SCSI_OPTIONS_DR、SCSI_OPTIONS_SYNC、SCSI_OPTIONS_TAG、SCSI_OPTIONS_FAST、SCSI_OPTIONS_WIDE、SCSI_OPTIONS_FAST20、SCSI_OPTIONS_FAST40、および SCSI_OPTIONS_FAST80 です。</p> <p>周期的な時間間隔である scsi-watchdog-tick が経過する度に、glm ドライバは、現在実行中および切り離されたすべてのコマンドを調べ、時間切れかどうかを確認します。</p> <p>scsi-tag-age-limit は、glm ドライバがすべてのタグ ID を一巡して調べた後に、現在使用されている特定のタグ ID の割り当てを試みる回数です。同一のタグ ID が scsi-tag-age-limit の回数だけ使用されると、すべての未処理のコマンドが完了するか、時間切れになるまで、このターゲットにコマンドは発行されません。</p> <p>詳細については、scsi_hba_attach(9F) を参照してください。</p>
使用例	<p>例 1 glm 設定ファイルの使用</p> <p>ファイル /kernel/drv/glm.conf を作成して、以下の行を追加してください。</p> <pre>scsi-options=0x78;</pre> <p>これは、すべての glm インスタンスに対して、タグ付き待ち行列化、Fast SCSI、Fast/Ultra SCSI、Wide モードを無効にします。</p> <p>以下の例は、1 つの特定の glm のオプションを無効にします。詳細については、driver.conf(4) と pci(4) を参照してください。</p> <pre>name="glm" parent="/pci@1f,4000"   unit-address="3"   target1-scsi-options=0x58   scsi-options=0x178 scsi-initiator-id=6;</pre>

## 例 1 glm 設定ファイルの使用 (続き)

OBP のデフォルトのイニシエータ ID は 7 ですが、接続時に 6 に変更されます。OBP のイニシエータ ID を変更してください。

上記の例では、ターゲット 1 の `scsi-options` を `0x58` に設定し、この SCSI バス上にある他のすべてのターゲットは `0x178` に設定しています。

親の物理パス名は、`/devices` ツリーを使用するか、論理デバイス名のリンクをたどることによって判定することができます。

```
# ls -l /dev/rdisk/c0t0d0s0
lrwxrwxrwx 1 root root 45 May 16 10:08 /dev/rdisk/c0t0d0s0 ->
  . . . devices/pci@1f,4000/scsi@3/sd@0,0:a,raw
```

この場合、上記の例と同じように、親は `/pci@1f,4000` で、`unit-address` は、`scsi@3` ノードに結合されている番号です。

`scsi-options` をターゲットごとに設定するには、以下のように入力します。

```
target1-scsi-options=0x78;
device-type-scsi-options-list =
  "SEAGATE ST32550W", "seagate-scsi-options" ;
seagate-scsi-options = 0x58;
scsi-options=0x3f8;
```

この例は、ターゲット 1 の `scsi-options` を `0x78` に設定し、ある特定の種類のディスクの `scsi-options` を `0x58` に設定し、この SCSI バス上にある他のすべてのターゲットの `scsi-options` を `0x3f8` に設定します。

ターゲット ID ごとに指定された `scsi-options` は、最も優先順位が高くなります。その次に優先順位が高いのは、デバイスの種類ごとに指定された `scsi-options` です。バスごとに指定される広域的な `scsi-options` (すべてのインスタンスに影響します) は、最も優先順位が低くなります。

指定された `scsi-options` が有効になるには、システムを再起動する必要があります。

## ドライバの機能

ターゲットドライバがドライバの機能を有効にするには、glm ドライバの機能を設定する必要があります。ターゲットドライバは、これらの機能の照会および変更ができます。照会および変更ができる機能には、`synchronous`、`tagged-qing`、`wide-xfer`、`auto-rqsense`、`qfull-retries`、`qfull-retry-interval` があります。他のすべての機能は、照会のみ行うことができます。

デフォルトでは、`tagged-qing`、`auto-rqsense`、`wide-xfer` 機能は無効で、`disconnect`、`synchronous`、`untagged-qing` 機能は有効です。これらの機能には、2 進数の値 (0 または 1) のみを設定することができます。`qfull-retries` のデフォルト値は 10 で、`qfull-retry-interval` のデフォルト値は 100 です。`qfull-retries` 機能は、`u_char` 型 (0 ~ 255) で、`qfull-retry-interval` は、`u_short` 型 (0 ~ 65535) です。

ターゲットドライバは、`tagged-qing` および `wide-xfer` を明示的に有効にする必要があります。`untagged-qing` 機能は常に有効で、その値を変更することはできません。

`scsi-options` の値と機能が矛盾している場合は、`scsi-options` に設定されている値が常に優先されます。`scsi_ifsetcap(9F)` 呼び出しでは、`whom != 0` のみを使用することができます。

詳細については、`scsi_ifsetcap(9F)` および `scsi_ifgetcap(9F)` を参照してください。

## ファイル

`/kernel/drv/glm` ELF ELF カーネルモジュール  
`/kernel/drv/glm.conf` 省略可能な設定ファイル

## 属性

以下の属性については、`attributes(5)` を参照してください。

属性の種類	属性の値
アーキテクチャ	LSI 53c810、LSI 53c875、LSI 53c876、LSI 53C896、および LSI 53C1010 SCSI、入出力プロセッサを持つ PCI ベースのシステムに限定

## 関連項目

`prtconf(1M)`, `driver.conf(4)`, `pci(4)`, `attributes(5)`, `scsi_abort(9F)`, `scsi_hba_attach(9F)`, `scsi_ifgetcap(9F)`, `scsi_ifsetcap(9F)`, `scsi_reset(9F)`, `scsi_sync_pkt(9F)`, `scsi_transport(9F)`, `scsi_device(9S)`, `scsi_extended_sense(9S)`, `scsi_inquiry(9S)`, `scsi_pkt(9S)`

## Writing Device Drivers

『ANSI Small Computer System Interface-2 (SCSI-2)』

LSI Logi 社 (旧 Symbios Logic 社.):

- SYM53c810 PCI-SCSI 入出力プロセッサ - Narrow モードをサポートする
- SYM53c875 PCI-SCSI 入出力プロセッサ - Fast-20 をサポートする
- SYM53c876 PCI-SCSI 入出力プロセッサ - デュアルチャネル Fast-20 をサポートする
- SYM53c896 PCI-SCSI 入出力プロセッサ - デュアルチャネル Fast-40 をサポートする
- SYM53c1010 PCI-SCSI 入出力プロセッサ - デュアルチャネル Fast-80 をサポートする

## 診断

以下で説明するメッセージは、コンソールに表示されると同時にログファイルに記録されます。

Device is using a hilevel intr

`glm` ドライバが使用することができない割り込みレベルでデバイスが設定されています。PCI デバイスを調べてください。

map setup failed

ドライバがデバイスレジスタを割り当てることができません。問題のあるハードウェアを調べてください。ドライバがデバイスに接続することができないために、SCSI デバイスにアクセスすることができません。

**glm\_script\_alloc failed**

ドライバが SCSI プロセッサの SCRIPTS を読み込むことができません。問題のあるハードウェアを調べてください。ドライバがデバイスに接続することができないために、SCSI デバイスにアクセスすることができません。

**cannot map configuration space.**

ドライバが設定レジスタを割り当てることができません。問題のあるハードウェアを調べてください。ドライバがデバイスに接続することができないために、SCSI デバイスにアクセスすることができません。

**attach failed**

ドライバが接続することができません。通常、接続が失敗した理由を示す警告が出力されます。これらは、ハードウェア障害とみなすことができます。

**SCSI bus DATA IN phase parity error**

ドライバが、SCSI バスでパリティエラーを検出しました。

**SCSI bus MESSAGE IN phase parity error**

ドライバが、SCSI バスでパリティエラーを検出しました。

**SCSI bus STATUS phase parity error**

ドライバが、SCSI バスでパリティエラーを検出しました。

**Unexpected bus free**

通知なしに、ターゲットがバスから切り離されました。問題のあるハードウェアを調べてください。

**Disconnected command timeout for Target <id>.<lun>**

ターゲット *id/lun* が切り離されている間に時間切れが起きました。これは、通常、ターミナルのファームウェアの問題です。タグ付き待ち行列化を行うターゲットについては、時間切れが検出された際に *<n>* コマンドが未処理になります。

**Disconnected tagged cmds (<n>) timeout for Target<id>.<lun>**

ターゲットまたは、*lun* が切り離されている間に時間切れが発生しました。これは、通常、ターゲットのファームウェアの問題によるものです。タグ付きの待ち行列化のターゲットでは、*<n>* コマンドは時間切れが検出された時点でまだ処理が終了していませんでした。

**Connected command timeout for Target <id>.<lun>**

これは、通常、SCSI バスの問題です。ケーブルと終端を調べてください。

**Target <id> reducing sync. transfer rate**

データ転送の停滞または DATA-IN フェーズパリティエラーが検出されました。ドライバは、データの転送レートを下げることによって、この問題の解決を試みません。

**Target <id> reverting to async. mode**

このターゲットの2つ目のデータ転送の停滞が検出されました。ドライバは、データの転送レートを下げることによって、この問題の解決を試みます。



**Target <id> disabled wide SCSI mode**

このターゲットの2つ目のデータフェーズの停滞が検出されました。ドライバは、Wide SCSI モードを無効にすることによって、この問題の解決を試みます。

**auto request sense failed**

自動要求パケットの開始が失敗しました。別の自動要求パケットがすでに転送されている可能性があります。

**invalid reselection (<id>.<lun>)**

再選択が失敗しました。ターゲットは、**abort** または **reset** を受け付けましたが、再接続を試みます。問題のあるハードウェアを調べてください。

**invalid intcode**

SCRIPTS プロセッサが無効の SCRIPTS 割り込みを生成しました。問題のあるハードウェアを調べてください。

**注意事項**

glm ドライバは次の LSI チップをサポートします。

- LSI 53C810 - Narrow, Fast SCSI モードをサポートします。最大 SCSI 帯域幅は 10M バイト/秒です。
- LSI 53C875 - Wide, Fast, および Ultra SCSI モードをサポートします。最大 SCSI 帯域幅は 40M バイト/秒です。
- LSI 53C896 - Wide, Fast, および Ultra SCSI 2 モードをサポートします。最大 LVD SCSI 帯域幅は 80M バイト/秒です。
- LSI 53C1010 - Wide, Fast, および Ultra SCSI 3 モードをサポートします。最大 LVD SCSI 帯域幅は 160M バイト/秒です。

glm ドライバは、ネゴシエートされた転送速度 (`target<n>-sync-speed`)、Wide バスに対応しているかどうか (`target<n>-wide`)、特定のターゲットの `scsi-options` (`target<n>-scsi-options`)、タグ付き待ち行列化が有効であるかどうか (`target<n>-TQ`) をターゲットごとに示す属性をエクスポートします。`sync-speed` の属性値は、KB/秒単位のデータ転送速度です。`target<n>-TQ` および `target<n>-wide` 属性は、ターゲットに対して対応する機能が有効であれば 1 を、無効であれば 0 の値を取ります。glm の属性を表示する場合は、`prtconf(1M)` (表示オプション) を参照してください。

**scsi, instance #0****Driver properties:**

```
name <target6-TQ> length <4>
value <0x00000000>.
name <target6-wide> length <4>
value <0x00000000>.
name <target6-sync-speed> length <4>
value <0x00002710>.
name <target1-TQ> length <4>
value <0x00000001>.
name <target1-wide> length <4>
value <0x00000000>.
name <target1-sync-speed> length <4>
value <0x00002710>.
```

```
name <target0-TQ> length <4>
  value <0x00000001>.
name <target0-wide> length <4>
  value <0x00000001>.
name <target0-sync-speed> length <4>
  value <0x00009c40>.
name <scsi-options> length <4>
  value <0x000007f8>.
name <scsi-watchdog-tick> length <4>
  value <0x0000000a>.
name <scsi-tag-age-limit> length <4>
  value <0x00000002>.
name <scsi-reset-delay> length <4>
  value <0x00000bb8>.
name <latency-timer> length <4>
  value <0x00000088>.
name <cache-line-size> length <4>
  value <0x00000010>.
```

名前	hme – SUNW,hme Fast-Ethernet デバイスドライバ
形式	/dev/hme
機能説明	SUNW,hme Fast-Ethernet ドライバは、マルチスレッド化された、読み込みや複製が可能な STREAMS ハードウェアドライバです。このドライバでは、SUNW,hme Fast-Ethernet コントローラにおいてコネクションレス型のデータリンクプロバイダインタフェース(dlpi(7P))を使用することができます。マザーボードと数種類の追加型 SBus SUNW,hme コントローラを使用することができます。このドライバでは、システムにインストールされた複数の SUNW,hme コントローラを使用することができます。

hme ドライバは、SUNW,hme ハードウェアの基本的な機能を提供します。このドライバは、SUNW,hme デバイスを扱うために使用します。機能には、チップの初期化、フレームの転送と受信、マルチキャストとプロミスキュアスへの対応、エラーからの回復および報告が含まれます。SUNW,hme デバイスは、サンの FEPS ASIC と内部トランシーバを使用して 100Base-TX ネットワークインタフェースを提供します。FEPS ASIC は、SBus インタフェースと MAC 機能を提供します。を使用します。物理層機能は、RJ45 コネクタに接続する内部トランシーバによって提供されます。PCI 対応の SUNW,hme デバイスは、サンの PFEX ASIC を使用して PCI インタフェースと MAC 機能を提供します。両方のデバイスとも 100Base-TX オンボードトランシーバに接続します。100Base-TX オンボードトランシーバは、RJ-45 コネクタに接続し、物理層の機能と外部との接続を提供します。RJ-45 コネクタに加えて、MII (Media Independent Inter-face) コネクタも SunSwitch SBus アダプタボードを除くすべての SUNW,hme デバイスで提供されます。MII インタフェースは、100Base-TX 規格で指定された物理媒体 (銅またはファイバ) を使用する外部トランシーバに接続するために使用されます。外部トランシーバが MII に接続されている場合、hme ドライバは外部トランシーバを選択し、内部トランシーバを無効にします。

100Base-TX 規格によって、操作のモードと速度を自動的に選択するための「自動ネゴシエーション」プロトコルを指定されます。内部トランシーバは、接続の遠隔地側 (接続相手) と「自動ネゴシエーション」を行うことができ、遠隔地側の機能に関する情報を受け取ります。内部トランシーバは、属性に基づいて、「最も優先順位の高い共通のモード」を選択します。また、ドライバが操作のモードを選択することができる場合は、強制モードを使用することができます。

アプリケーション  
プログラミング  
インタ  
hme と DLPI

複製された文字型特殊デバイスの /dev/hme は、システムにインストールされたすべての SUNW,hme コントローラに接続するために使用します。

hme ドライバは、「スタイル 2」のデータリンクサービスプロバイダです。すべての M\_PROTO 型および M\_PCPROTO 型のメッセージは、DLPI 基本式として解釈されます。有効な DLPI 基本式は <sys/dlpi.h> の中で定義されています。詳細は、dlpi(7P) を参照してください。開いているストリームに特殊デバイス (ppa) を関連付けるには、ユーザーからの明示的な DL\_ATTACH\_REQ メッセージが必要となります。この ppa ID は、符号なしの長整数として伝達され、対応するデバイスインスタンス (ユニット) 番号を示します。ppa フィールドの値がシステムにおける有効なデバイスインスタンス番号に対応していない場合は、エラー (DL\_ERROR\_ACK) が hme ドライバによって返されます。デバイスは最初の接続時に初期化され、最後の切り離し時に終了 (停止) します。

ユーザーからの DL\_INFO\_REQ に応答して hme ドライバが返す DL\_INFO\_ACK 基本式の値を以下に示します。

- SDU の最大値は 1500 になります (<sys/ethernet.h> で定義されている ETHERMTU)。
- SDU の最小値は 0 です。
- dlsapdlsap アドレス長は 8 です。
- MAC 形式は DL\_ETHER です。
- sap 長の値は -2 で、これは、DLSAP アドレスの中で物理アドレス構成要素のすぐ後に 2 バイト長の sap 構成要素が位置することを意味します。
- サービスモードは DL\_CLDLS です。
- サービスの品質 (QOS: quality of service) オプションは現在のところ含まれていないため、QOS フィールドは 0 です。
- プロバイダスタイルは DL\_STYLE2 です。
- バージョンは DL\_VERSION\_2 です。
- ブロードキャストアドレスの値は Ethernet/IEEE のブロードキャストアドレス (0xFFFFF) です。

DL\_ATTACHED 状態になると、特定の sap (Service Access Pointer) をストリームと関連付けるために、ユーザーは DL\_BIND\_REQ を送る必要があります。hme ドライバは、DL\_BIND\_REQ 内の SAP フィールドを Ethernet の「タイプ」として解釈します。したがって、sap フィールドの有効な値は [0-0xFFFF] の範囲になります。一度にストリームに接続できるのは 1 つのタイプの Ethernet だけです。

ユーザーが sap の値として 0 を選択すると、受信側は 802.3 モードになります。タイプフィールドが [0-1500] の範囲内にある媒体から受信したすべてのフレームは 802.3 フレームとみなされ、sap 値が 0 のすべての開いているストリームに配信されます。複数のストリームが 802.3 モードにある場合は、フレームは複製され、DL\_UNITDATA\_IND メッセージとして複数のストリームに配信されます。

hme ドライバは、送信の際に DL\_BIND\_REQ の sap フィールドの値が 0 であるかどうか、宛先タイプフィールドが [0-1500] の範囲内であるかどうかを確認します。どちらかの条件が満たされている場合、ドライバは最初の M\_PROTO mblk (メッセージブロック) を除いたすべての後続の DL\_UNITDATA\_REQ メッセージの長さを計算し、MAC フレームヘッダー長フィールドが計算結果である 802.3 フレームを送信します。

hme ドライバの DLSAP アドレス形式は、6 バイトの物理 (Ethernet) アドレス部分と、それに続く 2 バイトの sap (タイプ) 部分とで構成されます。アプリケーションでは、DLSAP アドレスの作成や削除を行う際に、実装に依存した DLSAP アドレス形式を明示的に使用せずに、DL\_INFO\_ACK 基本式で返される情報を使用してください。sap 長、完全な DLSAP 長、sap/物理の順序は DL\_INFO\_ACK に含まれます。物理アドレス長は、完全な DLSAP アドレス長から sap 長を引くか、DL\_PHYS\_ADDR\_REQ を発行して、ストリームに対応する現在の物理アドレスを取得することによって計算することができます。

DL\_BOUND 状態になると、hme ドライバに DL\_UNITDATA\_REQ メッセージを送信することによって、ユーザーは Ethernet 上でフレームを送信することができます。hme ドライバ

は、DL\_UNITDATA\_INDメッセージのEthernetのタイプと一致するsap値を持ち、開いている結合されているストリームすべてに、受信されたEthernetフレームを配信します。受信されたEthernetフレームは複製され、必要に応じて複数の開いているストリームに配信されます。DL\_UNITDATA\_REQおよびDL\_UNITDATA\_INDメッセージに含まれるDLSAPアドレスは、sap(タイプ)と物理(Ethernet)部分で構成されます。

hmeドライバは、強制的なコネクションレスDLPIメッセージセットに加えて、以下の基本式に対応しています。

#### hmeの基本式

DL\_ENABMULTI\_REQおよびDL\_DISABMULTI\_REQ基本式は、個々のマルチキャストグループアドレスの受信を有効または無効にします。これらの基本式を使用して、一群のマルチキャストアドレスをストリームごとに繰り返し作成および変更することができます。これらの基本式は、DL\_ATTACHED後の状態にある場合に、hmeドライバによって受け付けられます。

dl\_levelフィールド内にDL\_PROMISC\_PHYSフラグが設定された状態でのDL\_PROMISCOFF\_REQおよびDL\_PROMISCON\_REQの基本式は、ローカルホストによって生成されたフレームを含む、媒体上のすべての(「プロミスキュアモード」の)フレームの受信を有効または無効にします。DL\_PROMISC\_SAPフラグが設定されている状態で使用すると、sap値(Ethernetのタイプ)の受信を有効または無効にします。DL\_PROMISC\_MULTIフラグが設定されている状態で使用すると、すべてのマルチキャストグループアドレスの受信を有効または無効にします。どちら効果も、常にストリームごとに適用され、当該ストリームやその他のストリームのsapおよび物理レベルの設定の影響は受けません。

DL\_PHYS\_ADDR\_REQ基本式は、DL\_PHYS\_ADDR\_ACK基本式の中のストリームに対応する(接続されている)Ethernetアドレス(6桁の数)を返します。この基本式は、DL\_ATTACH\_REQの成功後の状態においてのみ有効です。

DL\_SET\_PHYS\_ADDR\_REQ基本式は、このストリームに対応する(接続されている)Ethernetアドレスを変更します。このストリームを最初に開く処理は、スーパーユーザーのみが行うことができます。スーパーユーザーでない場合は、DL\_ERROR\_ACK.でEPERMが返されます。この基本式は、このデバイスに接続された現在および未来のすべてのストリームに影響する意味で破壊的です。この基本式がこのストリームに対して成功すると、このデバイスに接続された他のすべてのストリームにM\_ERRORが送信されます。物理アドレスが一度変更されると、その後開かれ、このデバイスに接続されたストリームは、この新しい物理アドレスを取得します。物理アドレスが一度変更されると、物理アドレスを再び変更するためにこの基本式が使用されるか、システムが再起動されるまでこの物理アドレスは存続します。

#### hmeドライバ

デフォルトでは、内部トランシーバが使用されている場合に、hmeドライバは接続のモードおよび速度を選択するために「自動ネゴシエーション」を実行します。

外部トランシーバがMIIインタフェースに接続されている場合、hmeドライバは、ネットワーク操作に外部トランシーバを選択します。外部トランシーバが「自動ネゴシエーション」に対応している場合、hmeドライバは、自動ネゴシエーションの手続きを使用して接続のモードと速度を選択します。外部トランシーバが自動ネゴシエーションに対応していない場合、hmeドライバはそのトランシーバに対応している最も優先度の高いモードを選択します。

接続は、以下の4つのモードのいずれかになります。

- 100 Mbps、全二重
- 100 Mbps、半二重
- 10 Mbps、全二重
- 10 Mbps、半二重

これらの速度とモードについては、100Base-TX 規格に記述されています。

自動ネゴシエーションプロトコルは以下を自動的に選択します。

- 動作モード (全二重または半二重)
- 速度 (100 Mbps または 10 Mbps)

自動ネゴシエーションプロトコルは以下を行います。

- 接続相手に対応しているすべての動作モードを取得します。
- 接続相手に対して自身の機能を通知します。
- 優先順位に基づいて最も優先順位の高い共通のモードを選択します。

内部トランシーバは、上記のすべての動作モードおよび速度に対応しています。デフォルトでは、内部トランシーバが使用されている場合に、接続のモードおよび速度と接続相手との共通の動作モードを選択するために自動ネゴシエーションが使用されます。

外部トランシーバが MII インタフェースに接続されている場合、hme ドライバは、ネットワーク操作用に外部トランシーバを選択します。外部トランシーバが自動ネゴシエーションに対応している場合は、以下のようになります。

- hme ドライバは、接続の速度とモードを選択するために自動ネゴシエーションを使用します。

外部トランシーバが自動ネゴシエーションに対応していない場合は、以下のようになります。

- hme ドライバは、トランシーバに対応している最も優先順位の高いモードを選択します。

ユーザーが速度とモードを指定することもできます。SUNW,hme デバイスは、プログラム可能なパケット間隔 (“IPG”: Inter-Packet Gap) パラメタの ipg1 および ipg2 に対応しています。デフォルトでは、ドライバは ipg1 を 8 バイト時間に設定し、ipg2 を 4 バイト時間に設定します。これらは標準的な値です。ユーザーは、ドライバが 10 Mbps と 100 Mbps のどちらに対応しているかによって IPG パラメタの値を変更する場合があります。IPG は 9.6 または 0.96 マイクロ秒に設定されます。

#### hme のパラメタ

hme ドライバでは、SUNW,hme デバイスに関するさまざまなパラメタを設定および取得することができます。これらのパラメタには、次の内容が含まれます。

現在のトランシーバの状態  
現在の接続の状態

パケット間隔  
ローカルランシーバの機能  
接続相手の機能

ローカルランシーバには2組の機能があります。1つ目の組はハードウェアの性能に関するもので、読み取り専用 (RO) のパラメタです。もう1つの組は、ユーザーが選択する値に関するもので、「速度の選択」で使用されます。読み取り・書き込み (RW) 機能もあります。起動時には、これら2組の機能は同じです。接続相手の機能も読み取り専用のパラメタです。これは、これらのパラメタの現在のデフォルト値が読み取り専用で、変更することができないためです。

ファイル	/dev/hme	hme 文字型特殊デバイス
	/kernel/drv/hme.conf	システム共通のデフォルトデバイスドライバ属性
関連項目	nnd(1M), netstat(1M), driver.conf(4), dlpi(7P)	

名前	m64 - PGX、PGX24、PGX64 フレームバッファードバイスドライバ
機能説明	m64 ドライバは、Sun PGX グラフィックアクセラレータデバイスドライバです。
ファイル	/dev/fbs/m64 <i>n</i> デバイス特殊ファイル
関連項目	m64config(1M)



名前	md - ユーザーが構成可能な疑似デバイスドライバ
機能説明	<p>md は、ユーザーが構成することができる疑似デバイスドライバであり、ディスクの連結、ストライプ、ミラー、RAID5 メタデバイス、トランスメタデバイス、ホットスベアの各ユーティリティを提供します。トランスデバイスはすでにサポートが中止され、UFS ロギングに置き換えられました。mount_ufs(1M) を参照してください。</p> <p>ブロックデバイスは、システムの通常のバッファ機構によってディスクにアクセスし、物理ディスクレコードとは無関係に、読み取りや書き込みが行われます。また、ユーザによる読み取りや書き込み処理を直接反映するための raw デバイスもあります。通常、1 回の読み取りまたは書き込みごとに 1 回の入出力処理が行われるので、大量のデータを転送する場合は、raw デバイスによる入出力の方が効率的です。ブロックデバイスは /dev/md/dsk にあり、raw デバイスは /dev/md/rdsk にあります。メタデバイスにはディスク全体の内容が含まれているので、スライス (パーティション) は存在しません。</p> <p>メタデバイスに対する入出力要求 (lseek(2) など) のオフセットは、512 バイト (DEV_BSIZE) の倍数にしてください。これ以外の数値にすると、EINVAL エラーとなります。入出力を行うデータが 512 バイトの倍数でない場合は、ドライバによってデータ長が 512 バイトの倍数に切り上げられます。</p> <p>md 疑似デバイスドライバは、Solaris 2.4 以降のオペレーティングシステムで、すべてのディスクデバイスをサポートします。</p>
ioctl	<p>ここでは、メタディスクドライバでサポートされる ioctl の一覧を示します。</p> <p>/dev/md/rdsk/d0 などの raw メタデバイスに対して有効な ioctl は、以下のとおりです。詳細は、dkio(7i) を参照してください。</p> <p><b>DKIOCGGEOM</b>      ディスクのジオメトリを把握するために使用されます。メタディスクドライバは、メタデバイスの最初のコンポーネントから dkg_nhead、dkg_nsect、dkg_rpm、dkg_write_reinstruct、dkg_read_reinstruct を調べます (metainit 実行時)。dkg_ncyl は、メタデバイスのサイズ (metastat コマンドによって確認できる) を (dkg_nhead * dkg_nsect) で割ることによって算出されます。合計サイズは必ず (dkg_nhead * dkg_nsect) の倍数になります。メタデバイスの最初のコンポーネントがシリンダ番号 0 から始まらない場合は、dkg_ncyl が 1 シリンダだけ増加します。これは、DKIOCGVTOC がメタデバイスをシリンダ 1 から始まるとして通知するからです。この場合、シリンダ 0 が使用されていないように見えるという副作用がありますが、算術演算は正しく行われます。メタデバイスが設定されていないと、ENXIO エラーとなります。</p> <p><b>DKIOCINFO</b>      管理デバイスまたはメタデバイスに対して発行されると、dki_unit にメタデバイスのユニット番号を設定し、dki_ctype に DKC_MD の値を設定し、dki_partition を 0 に設定します (スライスがないため)。</p>

DKIOCGVTOC	現在の vtoc を返します。vtoc が書き込まれていない場合は、デフォルトの vtoc を返します。v_nparts は、常に 1 です。メタデバイスの最初のコンポーネントがシリンダ 0 から始まる場合は、v_part[0].p_start が 0 となります。最初のコンポーネントがシリンダ 0 以外から始まっている場合は、p_start フィールドがシリンダ 1 の開始セクタとなります。v_part[0].p_size は、metastat によって通知される合計サイズと同じです。
DKIOCSVTOC	メタデバイスの状態データベースに vtoc を格納します。その結果、vtoc がリブート後も保存されます。

## 診断

### 通知ログメッセージ

以下の通知ログメッセージがあります。

```
md: dnum: Hotspared device dev with dev
```

最初に表示されている名前のデバイスが、2 番目のデバイス (ホットスペア) に交換されました。

```
md: dnum: Hotspared device dev(num,num) with dev(num,num)
```

最初に表示されている番号のデバイスが、2 番目のデバイス (ホットスペア) に交換されました。

```
md: Could not load misc /dev
```

指定した misc モジュールをロードできません。このモジュールが見つからないか、このモジュールに何かが入力されました。

```
md: dnum: no mem for property dev
```

prop\_op エントリーポイントにメモリを割り当てることができません。

```
md: db: Parsing error on 'dev'
```

mddb.bootlist <number> に対して実行した /kernel/drv/md.conf の設定コマンドの形式が正しくありません。metadb -p を実行すると、/kernel/drv/md.conf ファイルに正しい設定コマンドが挿入されます。

```
md: dnum: dev(num,num) needs maintenance
```

```
md: dnum: dev needs maintenance
```

ミラー内のデバイス上で入出力エラーまたはオープンエラーが発生し、ミラー内のコンポーネントが保守 (Maintenance) 状態になりました。

```
md: dnum: dev(num,num) last erred md: dnum: dev last erred
```

ミラー内のデバイス上で入出力エラーまたはオープンエラーが発生し、データがミラー内のほかの場所に複製されないので、ミラー内のコンポーネントが最後にエラー (Last Erred) 状態になっています。

#### 警告ログメッセージ

以下の警告ログメッセージがあります。

md: State database is stale

使用可能な複製が不足しているため、状態データベースがデータベース内のレコードを更新できません。このような状態では、メタデバイスドライバに対するアクセスはすべて失敗します。この問題を解決するには、複製を追加したり、アクセスできない複製を削除してください。

md: dnum: read error on devmd: dnum: write error on dev

指定したデバイス名の指定されたサブミラーで、読み取りまたは書き込みのエラーが発生した

md: dnum: read error on dev(num,num)md: dnum: write error on dev(num,num)

指定されたデバイス番号の指定されたサブミラーで、読み取りまたは書き込みのエラーが発生した

md: State database commit failed

md: State database delete failed

状態データベースの複製が含まれているコンポーネントで、デバイスエラーが発生したこれらのエラーメッセージは、過半数の複製にエラーが発生したデバイスがある場合 (たとえば、状態データベースの複製が含まれているコンポーネントが3つあり、そのうちの2つにエラーが発生した場合など) にだけ発行されます。この場合、状態データベースの確定と削除が定期的に再試行されます。複製を追加すると、確定または削除が終了し、システムが正常に戻ります。複製を追加しないままにしておくタイムアウトとなり、システム障害が発生します。

md: dnum: Cannot load dev driver

指定された下位のドライバモジュール (たとえば、sd、id、xy、または他社製のドライバ) をロードできません。このドライバモジュールは、削除されている可能性があります。

md: Open error of hotspare devmd: Open error of hotspare dev(num,num)

指定されたホットスペアをオープンできないか、下位のドライバをロードできません。

#### パニックログメッセージ

パニックログメッセージとして、以下のメッセージがあります。

md: dnum: Unknown close typemd: dnum: Unknown open type

未知のオープン型 (OTYP) でメタデバイスがオープンまたはクローズされています。

md: State database problem

失敗したメタデバイス状態データベースの確定または削除が、100回(デフォルト値)再試行されました。

ファイル	/dev/md/dsk/dn	ブロックデバイス ( <i>n</i> はデバイス番号)
	/dev/md/rdisk/dn	raw デバイス ( <i>n</i> はデバイス番号)
	/dev/md/setname/dsk/dn	ブロックデバイス ( <i>setname</i> はディスクセット名、 <i>n</i> はデバイス番号)
	/dev/md/setname/rdisk/dn	raw デバイス ( <i>setname</i> はディスクセット名、 <i>n</i> はデバイス番号)
	/dev/md/admin	管理デバイス
	/kernel/drv/md	ドライバモジュール
	/kernel/drv/md.conf	ドライバ構成ファイル
	/kernel/misc/md_stripe	ストライプドライバ misc モジュール
	/kernel/misc/md_mirror	ミラードライバ misc モジュール
	/kernel/misc/md_hotspares	ホットスペアドライバ misc モジュール
	/kernel/misc/md_trans	UFS ロギング用メタトランスドライバ
	/kernel/misc/md_raid	RAID5 ドライバ misc モジュール
関連項目	mdmonitord(1M), metaclear(1M), metadb(1M), metadetach(1M), metahs(1M), metainit(1M), metaoffline(1M), metaonline(1M), metaparam(1M), metarecover(1M), metarename(1M), metareplace(1M), metaroot(1M), metassist(1M), metaset(1M), metastat(1M), metasync(1M), metattach(1M), md.cf(4), md.tab(4), attributes(5)	
	Solaris ボリュームマネージャの管理	
注意事項	トランスメタデバイスは UFS ロギングに置き換えられました。既存のトランスデバイスはロギングを行わず、その基となるデバイスにデータを直接渡します。UFS ロギングの詳細については、mount_ufs(1M) を参照してください。	

名前	mpt-SCSI ホストバスアダプタドライバ
形式	scsi@unit-address
機能説明	<p>mpt ホストバスアダプタドライバは、LSI 53C1030 SCSI と SAS1064 チップをサポートする SCSSA に準拠したネクサスドライバーです。</p> <p>mpt ドライバは、SCSA インタフェースで提供される標準機能に対応しています。その機能には、タグ付きおよびタグなしの待ち行列化、Narrow/Wide/Fast/Ultra SCSI/Ultra SCSI 2/Ultra SCSI 3/Ultra SCSI 4、自動要求検知が含まれます。mpt ドライバは、リンクされたコマンドには対応していません。</p>
ドライバの設定	<p>mpt.conf に属性を定義することで、mpt ドライバを設定します。これらの属性は、グローバルな SCSI 設定より優先されます。mpt ドライバは、以下の変更可能な属性に対応しています。scsi-options、target&lt;n&gt;-scsi-options、scsi-reset-delay、scsi-tag-age-limit、scsi-watchdog-tick、および scsi-initiator-id。</p> <p>属性 target&lt;n&gt;-scsi-options は、target&lt;n&gt; の scsi-options 属性値より優先されます。ここで、&lt;n&gt; は、10進数の 0 から 15 までの可変値です。mpt ドライバは、次の SCSI オプションに対応しています。SCSI_OPTIONS_DR, SCSI_OPTIONS_SYNC, SCSI_OPTIONS_TAG, SCSI_OPTIONS_FAST, SCSI_OPTIONS_WIDE, SCSI_OPTIONS_FAST20, SCSI_OPTIONS_FAST40, SCSI_OPTIONS_FAST80, SCSI_OPTIONS_FAST160、および SCSI_OPTIONS_QAS。</p> <p>一定の間隔で scsi-watchdog-tick が経過するたびに、mpt ドライバは、現在実行中および切り離されたすべてのコマンドを時間切れかどうか調べます。</p> <p>scsi-tag-age-limit 属性は、mpt ドライバがすべてのタグ ID を一巡して調べた後に、現在使用中のタグ ID の割り当てを試みる回数を示します。同一のタグ ID が scsi-tag-age-limit の回数だけ使用されると、すべての未処理のコマンドが完了するか、時間切れになるまで、このターゲットにコマンドは発行されません。</p> <p>詳細については、scsi_hba_attach(9F) を参照してください。</p>
使用例	<p>例 1 mpt 設定ファイルの使用</p> <p>ファイル /kernel/drv/mpt.conf を作成して、以下の行を追加してください。</p> <pre>scsi-options=0x78;</pre> <p>上の例は、すべての mpt インスタンスに対して、タグ付き待ち行列化、Fast/Ultra SCSI、および Wide モードを無効にします。</p> <p>以下の例は、1つの特定の mpt デバイスのオプションを無効にします。詳細については、driver.conf(4) と pci(4) を参照してください。</p> <pre>name="mpt" parent="/pci@1f,4000" unit-address="3" target1-scsi-options=0x58 scsi-options=0x178 scsi-initiator-id=6;</pre>

## 例 1 mpt 設定ファイルの使用 (続き)

OBP のデフォルトのイニシエータ ID は 7 ですが、接続時に ID 6 に変更されます。OBP のイニシエータ ID を変更するようにしてください。

上記の例では、ターゲット 1 の `scsi-options` を `0x58` に設定し、この SCSI バス上にある他のすべてのターゲットは `0x178` に設定しています。

親の物理パス名は、`/devices` ツリーを使用するか、論理デバイス名のリンクをたどることによって判定することができます。

```
# ls -l /dev/rdisk/c0t0d0s0
lrwxrwxrwx 1 root root      45 May 16 10:08 /dev/rdisk/c0t0d0s0 ->
. . / . . /devices/pci@1f,4000/scsi@3/sd@0,0:a,raw
```

これまでの例と同様、親は `/pci@1f,4000`、`unit-address` は `scsi@3` ノードへの結合番号です。

`scsi-options` をターゲットごとに設定するには、以下のように入力します。

```
target1-scsi-options=0x78;
device-type-scsi-options-list =
    "SEAGATE ST32550W", "seagate-scsi-options" ;
seagate-scsi-options = 0x58;
scsi-options=0x3f8;
```

これは、ターゲット 1 の `scsi-options` を `0x78` に設定します。SCSI バスのその他すべてのターゲットは `0x3f8` に設定されます (ただし、`scsi-options` が `0x58` に設定される特定のディスクタイプを除く)。

ターゲット ID ごとに指定された `scsi-options` は、最も優先順位が高くなります。その次に優先順位が高いのは、デバイスの種類ごとに指定された `scsi-options` です。バスごとに指定される広域的な `scsi-options` (すべての mpt インスタンスに影響します) は、最も優先順位が低くなります。

指定した `scsi-options` を有効にするには、システムを再起動する必要があります。

## ドライバの機能

ドライバの機能を有効にするには、ターゲットドライバが mpt ドライバの機能を設定する必要があります。以下の機能は、ターゲットドライバによって照会および変更できません。 `synchronous`、`tagged-qing`、`wide-xfer`、`auto-rqsense`、`qfull-retries`、および `qfull-retry-interval`。その他の機能は、照会のみです。

デフォルトでは、`tagged-qing`、`auto-rqsense`、および `wide-xfer` 機能は無効で、`disconnect`、`synchronous`、および `untagged-qing` 機能は有効です。これらの機能には、2進値 (0 または 1) しか設定できません。一方、`qfull-retries` 機能のデフォルトは 10、`qfull-retry-interval` 機能のデフォルトは 100 です。`qfull-retries` 機能は `uchar_t` (0 から 255) で、`qfull-retry-interval` 機能は `ushort_t` (0 から 65535) です。

ターゲットドライバは、`tagged-qing` および `wide-xfer` を明示的に有効にする必要があります。`untagged-qing` 機能は常に有効で、その値を変更することはできません。

`scsi-options` の値と機能が矛盾している場合は、`scsi-options` に設定されている値が優先されます。`scsi_ifsetcap(9F)` 呼び出しでは、`whom != 0` のみを使用することができます。

詳細については、`scsi_ifsetcap(9F)` および `scsi_ifgetcap(9F)` を参照してください。

## ファイル

<code>/kernel/drv/mpt</code>	32-ビット ELF カーネルモジュール
<code>/kernel/drv/sparcv9/mpt</code>	64-bit SPARC ELF カーネルモジュール
<code>/kernel/drv/amd64/mpt</code>	64-bit x86 ELF カーネルモジュール
<code>/kernel/drv/mpt.conf</code>	省略可能な設定ファイル

## 属性

以下の属性については、`attributes(5)` を参照してください。

属性の種類	属性の値
Architecture	x86 SPARC (下記参照)

このドライバは、LSI 53c1030 SCSI I/O プロセッサを搭載した PCI ベースのシステムに限定されます。

## 関連項目

`prtconf(1M)`, `driver.conf(4)`, `pci(4)`, `attributes(5)`, `scsi_abort(9F)`, `scsi_hba_attach(9F)`, `scsi_ifgetcap(9F)`, `scsi_ifsetcap(9F)`, `scsi_reset(9F)`, `scsi_sync_pkt(9F)`, `scsi_transport(9F)`, `scsi_device(9S)`, `scsi_extended_sense(9S)`, `scsi_inquiry(9S)`, および `scsi_pkt(9S)`

Writing Device Drivers

ANSI Small Computer System Interface-2 (SCSI-2)

SYM53c1030 PCI-SCSI I/O processor Dual Channel Fast-160 — LSI Logic Inc.

## 診断

以下で説明するメッセージは、ログファイルに記録され、システムコンソールにも表示されます。

**Device is using a hilevel intr**

`mtp` ドライバが使用できない割り込みレベルでデバイスが設定されています。PCI デバイスを調べてください。

**Map setup failed**

ドライバがデバイスレジスタを割り当てることができませんでした。問題のあるハードウェアを調べてください。ドライバがデバイスに接続できないために、SCSI デバイスにアクセスすることができません。

**Cannot map configuration space**

ドライバは設定レジスタを割り当てるできませんでした。問題のあるハードウェアを調べてください。SCSI デバイスにアクセスできません。

**Attach failed**

ドライバは接続できませんでした。通常、接続に失敗した理由を示す別の警告が先に出力されます。これらは、ハードウェア障害とみなすことができます。

**Connected command timeout for Target <id>.**

これは、通常、SCSI バスの問題によるものです。ケーブルと終端を調べてください。

**Target <id> reducing sync. transfer rate**

データ転送の停滞または DATA-IN フェーズパリティエラーが検出されました。ドライバは、データの転送速度を下げることによって、この問題の解決を試みます。

**Target <id> reverting to async. mode**

このターゲットの2つ目のデータ転送の停滞が検出されました。ドライバは、データの転送速度を下げることによって、この問題の解決を試みます。

**Target <id> disabled wide SCSI mode**

このターゲットの2つ目のデータフェーズの停滞が検出されました。ドライバは、Wide SCSI モードを無効にすることによって、この問題の解決を試みます。

**注意事項**

mpt ドライバは LSI 53c1030 チップに対応しています。LSI 53c1030 チップシリーズは、Wide、Fast および Ultra SCSI 4 モードに対応しています。最大 LVD SCSI 帯域幅は 320M バイト/秒です。

mpt は、ターゲットごとのネゴシエートされた転送速度 (`target<n>-sync-speed`)、その特定のターゲット (`target<n>-scsi-options`) がワイドバスに対応しているかどうか (`target<n>-wide`)、およびタグ付き待ち行列化が有効であるかどうか (`target<n>-TQ`) を示す属性をエクスポートします。`sync-speed` の属性値は、KB/秒単位のデータ転送速度です。`target<n>-TQ` および `target<n>-wide` 属性は、当該ターゲットに対応する機能が有効であれば 1 を、無効であれば 0 の値を取ります。mpt 属性を表示する場合の詳細については、`prtconf(1M)` (表示オプション) を参照してください。

**scsi, instance #4****Driver properties:**

```
name='target8-TQ' type=int items=1 dev=none
value=00000001
name='target8-wide' type=int items=1 dev=none
value=00000001
name='target8-sync-speed' type=int items=1 dev=none
value=00013880
name='target5-TQ' type=int items=1 dev=none
value=00000001
name='target5-wide' type=int items=1 dev=none
value=00000001
name='target5-sync-speed' type=int items=1 dev=none
value=00013880
```



```
name='target4-TQ' type=int items=1 dev=none
value=00000001
name='target4-wide' type=int items=1 dev=none
value=00000001
name='target4-sync-speed' type=int items=1 dev=none
value=00013880
name='pm-components' type=string items=3 dev=none
value='NAME=mpt4' + '0=Off (PCI D3 State)' +
'3=On (PCI \ D0 State)'
name='scsi-selection-timeout' type=int items=1 dev=(238,0)
value=000000fa
name='scsi-options' type=int items=1 dev=(238,0)
value=00103ff8
name='scsi-watchdog-tick' type=int items=1 dev=(238,0)
value=0000000a
name='scsi-tag-age-limit' type=int items=1 dev=(238,0)
value=00000002
name='scsi-reset-delay' type=int items=1 dev=(238,0)
value=00000bb8
```

名前 pm – 電源管理システムのドライバ

形式 /dev/pm

機能説明 電源管理システム (pm) のドライバは、アプリケーションが電源管理用にシステム内でデバイスを設定するインタフェースを提供します。インタフェースは、`ioctl(2)` コマンドを経由して提供されます。pm ドライバは、`/dev/pm` を使用してアクセスすることができます。

電源管理システムのフレームワーク 電源管理システムのフレームワークのモデルは、システムをデバイスの集合体として見ます。各デバイスは部品の集合体であり、部品は電源管理が可能な最小単位です。デバイス内の電源管理が可能な部品は、デバイスドライバの制御によって異なります。

電源管理可能な部品は、現在の電源レベルでは *busy* または *idle* のどちらかの状態となります。通常、電源管理フレームワークは、*idle* 状態にある部品を次に低い電源レベルにします。電源管理フレームワークは、部品がしきい値の時間以上アイドル状態にあることと、部品が属しているデバイスが依存条件を満たしていることの2つの要素によってこの遷移を判定します。依存関係は、あるデバイスが電源管理される前に別のデバイスが電源管理されている必要がある場合に発生します。依存関係はデバイス単位で発生します。依存関係が存在する場合、デバイスが依存するすべてのデバイスが最初に管理されないかぎり、そのデバイスの部品のいずれも管理することができません。

以下のコマンドを使用することによって、アプリケーションは電源管理フレームワークドライバからデバイスの電源管理の制御を受け取り、デバイスの電源レベルの遷移を直接管理することができます。

旧式の `ioctls` このセクションで記述されているすべての `ioctl` コマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。新しいコマンドについては、このマニュアルページの「新しい IOCTLS」セクションを参照してください。

コマンドのこのセットについて、`arg(ioctl(2))` を参照) は `<sys/pm.h>` で定義された型 `pm_request` の構造体を指します。

```
typedef struct {
    char *who;           /* 設定するデバイス */
    int select;         /* 部品またはデバイスの依存デバイスを選択 */
    int level;          /* 電源レベルまたはしきい値 */
    char *dependent;    /* 依存デバイスの名前を格納 */
    int size;           /* 依存デバイスのバッファの大きさ */
} pm_request;
```

フィールドには、以下のデータが含まれている必要があります。

`who` 設定されるデバイスの名前を指すポインタです。デバイスの専用のファイル名またはデバイスへの物理パスの部分文字列となることもあります。

`select` 設定される部品または依存デバイスを指定する正の整数です。番号付けは0で始まります。

<i>level</i>	秒単位のしきい値または目的の電源レベルを指定する正の整数です。
<i>dependent</i>	このデバイスが依存するデバイスの名前を格納しているバッファを指すポインタです。 <i>who</i> のフィールドと同じ形式を使用します。
<i>size</i>	依存デバイスのバッファの大きさです。

すべてのフィールドが、各コマンドで使用されるわけではありません。

#### PM\_DISABLE\_AUTOPM

*who* で指定されたデバイスに対しては、フレームワークによる電源管理が無効になっています。呼び出し元は、以下のコマンドを使用してこのデバイスの電源を管理します。このコマンドの実行が成功しなかった場合、それに続く `PM_SET_CUR_PWR` は失敗します。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに `PM_DIRECT_PM` コマンドを使用してください。

以下にエラーコードを示します。

EBUSY	フレームワークにより、デバイスの電源管理はすでに無効になっています。
EPERM	呼び出し元がスーパーユーザーまたはデバイスの所有者ではありません。

#### PM\_GET\_NORM\_PWR

*who* で指定されたデバイスの部品 *select* の通常の電源レベルが返されました。この部品の通常の電力は、この部品が再び使用中になるときにこの部品に対して設定される電源レベルです。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに `PM_GET_FULL_POWER` コマンドを使用してください。

以下にエラーコードを示します。

EINVAL	デバイスの部品が範囲外です。
EIO	デバイスには、電源管理が可能な部品がありません。

#### PM\_GET\_CUR\_PWR

*who* で指定されたデバイスの部品 *select* の現在の電源レベルが返されました。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに `PM_GET_CURRENT_POWER` コマンドを使用してください。

以下にエラーコードを示します。

EINVAL	デバイスの部品が範囲外です。
EAGAIN	デバイスの部品の電源レベルは、現在わかっていません。

#### PM\_SET\_CUR\_PWR

*who* で指定されたデバイスの部品 *select* に対して電源レベルが *level* になりました。*select* が 0 でなく、デバイスの部品 0 が電源レベル 0 にある場合、部品 0 に対して電源レベルが通常のレベルになります。このデバイスに依存する各デバイスの各部品に

対して、電源レベルは通常の電源レベルになります。影響を受ける各デバイスの各祖先の各部品に対して、電源レベルは通常の電源レベルになります。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに `PM_SET_CURRENT_POWER` コマンドを使用してください。

以下にエラーコードを示します。

<code>EINVAL</code>	デバイスの部品が範囲外か、0より低い電源レベルです。
<code>EIO</code>	デバイスまたはその祖先、またはその依存先またはそれらの祖先に対して電源投入が失敗しました。
<code>EPERM</code>	呼び出し元がスーパーユーザーまたはデバイスの所有者ではありません。

#### `PM_REENABLE_AUTOPM`

*who* で指定されたデバイスに対して、フレームワークによる電源管理が再び有効になりました。デフォルトでは、すべての構成されたデバイスはフレームワークによって電源管理されます。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに `PM_RELEASE_DIRECT_PM` コマンドを使用してください。

以下にエラーコードを示します。

<code>EINVAL</code>	デバイスはすでにフレームワークによって電源管理されています。
<code>EPERM</code>	呼び出し元がスーパーユーザーまたはデバイスの所有者ではありません。

## 新しい `ioctl`s

このセクションの `ioctl` コマンドは、前述されているは旧式のコマンドに代わるものです。これらの新しいコマンドは、異なる構造へのポインタを取り、より完全な機能をサポートします。

`ioctl` コマンドのセットについて、`arg(ioctl(2))` を参照)は、`<sys/pm.h>` で定義された型 `pm_req` の構造体を指します。

```
typedef struct pm_req {
    char *physpath;      /* 設定するデバイスの物理パス */
                        /* libdevinfo(3LIB) を参照 */
    int component;      /* デバイスの部品 */
    int value;          /* 電源レベル、しきい値、またはカウント */
    void *data;         /* コマンドに依存する可変サイズデータ */
    size_t datasize;    /* データバッファの大きさ */
} pm_req_t;
```

このフィールドには以下のデータが含まれている必要があります。

<i>physpath</i>	デバイスの物理パスを指すポインタです。libdevinfo(3LIB)を参照してください。たとえば、デバイス /devices/pseudo/pm@0:pm では、 <i>physpath</i> の値は /pseudo/pm@0 です。
<i>component</i>	設定される部品または依存デバイスを指定する正の整数です。番号付けは0から始まります。
<i>value</i>	秒単位のしきい値または目的の電源レベルを指定する正の整数、あるいは指定されているレベルの数値です。
<i>data</i>	このデバイスが依存するデバイスの名前などの可変サイズデータを格納しているか受け取るバッファを指すポインタです。
<i>size</i>	データバッファのサイズ。

すべてのフィールドが、各コマンドで使用されるわけではありません。

#### PM\_DIRECT\_PM

*physpath* で指定されたデバイスに対しては、フレームワークによって行われていた電源管理が無効になっています。呼び出し元は、PM\_DIRECT\_NOTIFY、PM\_GET\_TIME\_IDLE、PM\_GET\_CURRENT\_POWER、PM\_GET\_FULL\_POWER および PM\_SET\_CURRENT\_POWER コマンドを使用してこのデバイスの電源を管理します。「このドライバが、*pm\_raise\_power(9F)*、*pm\_lower\_power(9F)*、*pm\_power\_has\_changed(9F)* のいずれかと呼び出す」、または「このデバイスが電源レベルが変化している別のデバイスの親であるか、このデバイスが依存しているデバイスの電源レベルが変化している」かのどちらかの理由により、このデバイスで電源レベルの変更が必要になる場合には、電源レベルの変更は行われず、呼び出し元は、以下に PM\_DIRECT\_NOTIFY コマンドについて記述されているように電源レベルの変化について通知されます。

以下にエラーコードを示します。

EBUSY	フレームワークにより、デバイスの電源管理はすでに無効になっています。
EPERM	呼び出し元がスーパーユーザーでないか、または実効グループIDが0ではありません。

#### PM\_RELEASE\_DIRECT\_PM

*physpath* (PM\_DIRECT\_PM コマンドのターゲットである必要がある) で指定されたデバイスに対して、フレームワークによる電源管理が再び有効になりました。

以下にエラーコードを示します。

EINVAL	デバイスの部品が範囲外です。
--------	----------------

#### PM\_DIRECT\_NOTIFY PM\_DIRECT\_NOTIFY\_WAIT

これらのコマンドは、デバイスを直接電源管理しているプロセスに、このデバイスの電源レベルが変更されるかも知れないイベントについて通知します。このようなイベントが起こった場合は、このコマンドは該当イベントについての情報を返しません。

`arg` (`ioctl(2)`を参照) は、以下のように `<sys/pm.h>` で定義された型 `pm_state_change` の構造体を指します。

```
typedef struct pm_state_change {
    char    *physpath;    /* 状態が変化したデバイス */
    int     component;    /* どの部品の状態が変化したか */
#ifdef _BIG_ENDIAN
    ushort_t flags;      /* PSC_EVENT_LOST、PSC_ALL_LOWEST */
    ushort_t event;      /* イベントの型 */
#else
    ushort_t event;      /* イベントの型 */
    ushort_t flags;      /* PSC_EVENT_LOST、PSC_ALL_LOWEST */
#endif
    time_t  timestamp;    /* 状態の変化した時刻 */
    int     old_level;    /* 変更前の電源レベル */
    int     new_level;    /* 変更後の電源レベル */
    size_t  size;        /* physpath が指すバッファの大きさ */
} pm_state_change_t;
```

イベントが起こった場合は、`arg` が指す構造体に挿入されます。このイベントの型が `PSC_PENDING_CHANGE` の場合は、構造体の残りの部分の情報には、このデバイスが呼び出し元により直接電源管理されない場合にフレームワークがとるアクションが記述されます。呼び出し元は、以下に示す `PM_SET_CURRENT_POWER` を使用して指定された電源レベルの変更を完全に行います。

`PSC_HAS_CHANGED` のイベントの型は、デバイスを直接電源管理するドライバがそのデバイスの電源の変化のために、`pm_power_has_changed(9F)` を呼んだことを示します。これにより呼び出し元は、デバイスの電源の状態を追跡することができます。

システムは、イベントを循環バッファに入れてます。バッファがオーバーフローした場合、最も古いイベントが削除され、削除されたイベントの次に古いイベントが取り出された時に、`PSC_EVENT_LOST` フラグが設定されます。

保留中のイベントがない場合は、`PM_DIRECT_NOTIFY` は、`EWOULDBLOCK` を返します。また `PM_DIRECT_NOTIFY_WAIT` は、イベントが使用できるようになるまで実行されません。

`pm` は、`poll(2)` インタフェースもサポートします。イベントが保留中の場合、`/dev/pm` のファイル記述子を含み、かつイベントマスクに設定された `POLLIN` または `POLLRDNORM` を持つ `poll(2)` 呼び出しが返されます。

#### `PM_SET_CURRENT_POWER`

`physpath` (`PM_DIRECT_PM` コマンドを発行したプロセスに対するデバイスの物理パス) で指定されたデバイスの部品である `component` は、電源レベルが `value` に設定されています。`physpath` で指定されたデバイスのすべての部品の電源レベルが 0 で、`value` の値が 0 ではなく、しかも、このデバイスに依存するデバイスがある場合は、このコマンドが返される前に、デバイスのすべての部品がフルパワーになります。同様に、ターゲットデバイスの親の電源が切断されている場合、そのデバイスはこのコマン

ドが返される前に必要に応じて起動します。PM\_SET\_CURRENT\_POWERがデバイスに対して発行されると、結果として発生する電源の変更が、PM\_DIRECT\_NOTIFYのイベントリストに追加されます。

以下にエラーコードを示します。

EINVAL	デバイスの部品が範囲外か、0より低い電源レベルです。
EIO	デバイスまたはその祖先、またはその依存先またはそれらの祖先に対して電源投入が失敗しました。部品が <i>busy</i> の状態になったためにデバイスドライバがコマンドを不適切なものとして拒否した可能性があるため、このことは障害を示しているとは限らないことに注意してください。
EPERM	呼び出し元がこのデバイスに対して以前に行った PM_DIRECT_PM コマンドの実行が成功していません。

#### PM\_GET\_FULL\_POWER

*physpath* で指定されたデバイスの部品 *component* の使用可能なもっとも高い電源レベルが返されました。

#### PM\_GET\_CURRENT\_POWER

*physpath* で指定されたデバイスの部品 *component* の現在の電源レベルが返されました。

以下にエラーコードを示します。

EAGAIN	デバイス部品の電源レベルは、現在わかっていません。
--------	---------------------------

#### PM\_GET\_TIME\_IDLE

PM\_GET\_TIME\_IDLE は、*physpath* で指定されたデバイスの部品 *component* が *idel* 状態になっている秒数を返します。デバイスが *idel* 状態でない場合は、0 が返されます。

プロセスが PM\_GET\_TIME\_IDLE コマンドと PM\_SET\_CURRENT\_POWER コマンドを実行する間に *idel* 状態にある部品の電源レベルを小さくするために、デバイスの状態が変化することがあります。そのためにプロセスは、PM\_SET\_CURRENT\_POWER コマンドの実行が成功しなかったと返された場合に対処する必要があります。これはデバイスの部品が *busy* の状態になったために、デバイスがコマンドを不適切であるとして拒否することがあるためです。PM\_GET\_TIME\_IDLE コマンドを再び実行して、その部品が *busy* の状態であるかどうかを判断することで、ほかのタイプの障害と区別することができます。

#### エラー

エラーが起きると、コマンドは -1 を返し、*errno* を設定します。上記のコマンドごとのエラーコードに加えて、すべてのコマンドに共通するエラーコードを以下に示します。

EFAULT	引数に指定したアドレスが間違っています。
ENODEV	デバイスは電源管理することができないか、構成されていません。
ENXIO	デバイスを開く試行回数が多すぎます。

#### 属性

以下の属性については、*attributes(5)* を参照してください。

---

属性タイプ	属性値
インタフェースの安定性	不安定 (廃止された IOCTL のインタフェースは使用できません。)

**関連項目**

pmconfig(1M), intro(2), ioctl(2), libdevinfo(3LIB), power.conf(4), attributes(5), attach(9E), detach(9E), power(9E), pm\_busy\_component(9F), pm\_idle\_component(9F), pm\_lower\_power(9F), pm\_power\_has\_changed(9F), pm\_raise\_power(9F)

『Writing Device Drivers』



名前	qfe – SUNW,qfe Quad Fast-Ethernet デバイスドライバ
形式	/dev/qfe
機能説明	<p>SUNW,qfe Quad Fast-Ethernet ドライバは、マルチスレッド化された、読み込みや複製が可能な STREAMS ハードウェアドライバです。このドライバでは、SUNW,qfe Quad Fast-Ethernet コントローラにおいてコネクションレス型のデータリンクプロバイダインタフェース (dlpi(7P)) を使用することができます。このドライバでは、システムにインストールされた複数の SUNW,qfe コントローラを使用することができます。qfe ドライバは、SUNW,qfe ハードウェアの基本的な機能を提供します。このドライバは、SUNW,qfe デバイスを扱うために使用します。機能には、チップの初期化、フレームの転送と受信、マルチキャストとプロミスキューアスへの対応、エラーからの回復および報告が含まれます。</p>
SUNW,qfe	<p>SUNW,qfe デバイスは、100Base-TX ネットワークインタフェースを提供します。SUNW,qfe デバイスには、SBus に対応しているものと、PCI バスインタフェースに対応しているものの 2 種類があります。SBus 対応の SUNW,qfe デバイスは、サンの FEPS ASIC (SBus インタフェースと MAC 機能を提供します) を使用します。PCI 対応の SUNW,qfe デバイスは、サンの PFEX ASIC を使用して PCI インタフェースと MAC 機能を提供します。両方のデバイスとも 100Base-TX オンボードトランシーバに接続します。100Base-TX オンボードトランシーバは、RJ45 コネクタに接続し、物理層の機能と外部との接続を提供します。</p> <p>100Base-TX 規格によって、操作のモードと速度を自動的に選択するための「自動ネゴシエーション」プロトコルを指定されます。内部トランシーバは、接続の遠隔地側 (接続相手) と「自動ネゴシエーション」を行うことができ、遠隔地側の機能に関する情報を受け取ります。内部トランシーバは、属性に基づいて、「最も優先順位の高い共通のモード」を選択します。また、ドライバが操作のモードを選択することができる場合は、強制モードを使用することができます。</p>
アプリケーション プログラミング インタ qfe と DLPI	<p>複製された文字型特殊デバイスの /dev/qfe は、システムにインストールされたすべての SUNW,qfe コントローラに接続するために使用します。</p> <p>qfe ドライバは、「スタイル 2」のデータリンクサービスプロバイダです。すべての M_PROTO 型および M_PCPROTO 型のメッセージは、DLPI 基本式として解釈されます。有効な DLPI 基本式は &lt;sys/dlpi.h&gt; の中で定義されています。詳細は、dlpi(7P) を参照してください。開いているストリームに特殊デバイス (ppa) を関連付けるには、ユーザーからの明示的な DL_ATTACH_REQ メッセージが必要となります。この ppa ID は、符号なしの長整数として伝達され、対応するデバイスインスタンス (ユニット) 番号を示します。ppa フィールドの値がシステムにおける有効なデバイスインスタンス番号に対応していない場合は、エラー (DL_ERROR_ACK) が qfe ドライバによって返されます。デバイスは最初の接続時に初期化され、最後の切り離し時に終了 (停止) します。</p> <p>ユーザーからの DL_INFO_REQ に応答して qfe ドライバが返す DL_INFO_ACK 基本式の値を以下に示します。</p> <ul style="list-style-type: none"> <li>■ SDU の最大値は 1500 になります (&lt;sys/ethernet.h&gt; で定義されている ETHERMTU)。</li> </ul>

- SDUの最小値は0です。
- dlsapアドレス長は8です。
- MAC形式はDL\_ETHERです。
- sap長の値は-2で、これは、DLSAPアドレスの中で物理アドレス構成要素のすぐ後に2バイト長のsap構成要素が位置することを意味します。
- サービスモードはDL\_CLDLSです。
- サービスの品質(QOS: quality of service)オプションは現在のところ含まれていないため、QOSフィールドは0です。
- プロバイダスタイルはDL\_STYLE2です。
- バージョンはDL\_VERSION\_2です。
- ブロードキャストアドレスの値はEthernet/IEEEのブロードキャストアドレス(0xFFFFF)です。

DL\_ATTACHED状態になると、特定のSAP (Service Access Pointer) をストリームと関連付けるために、ユーザーはDL\_BIND\_REQを送る必要があります。qfeドライバは、DL\_BIND\_REQ内のsapフィールドをEthernetの「タイプ」として解釈します。したがって、sapフィールドの有効な値は[0-0xFFFF]の範囲になります。一度にストリームに接続できるのは1つのタイプのEthernetだけです。

ユーザーがsapの値として0を選択すると、受信側は802.3モードになります。タイプフィールドが[0-1500]の範囲内にある媒体から受信したすべてのフレームは802.3フレームとみなされ、sap値が0のすべての開いているストリームに配信されます。複数のストリームが802.3モードにある場合は、フレームは複製され、DL\_UNITDATA\_INDメッセージとして複数のストリームに配信されます。

qfeドライバは、送信の際にDL\_BIND\_REQのsapフィールドの値が0であるかどうかと、宛先タイプフィールドが[0-1500]の範囲内であるかどうかを確認します。どちらかの条件が満たされている場合、ドライバは最初のM\_PROTO mblk(メッセージブロック)を除いたすべての後続のDL\_UNITDATA\_REQメッセージの長さを計算し、MACフレームヘッダー長フィールドが計算結果である802.3フレームを送信します。

qfeドライバのDLSAPアドレス形式は、6バイトの物理(Ethernet)アドレス部分と、それに続く2バイトのsap(タイプ)部分とで構成されます。アプリケーションでは、DLSAPアドレスの作成や削除を行う際に、実装に依存したDLSAPアドレス形式を明示的に使用せずに、DL\_INFO\_ACK基本式で返される情報を使用してください。sap長、完全なDLSAP長、sap/物理の順序はDL\_INFO\_ACKに含まれます。物理アドレス長は、完全なDLSAPアドレス長からsap長を引くか、DL\_PHYS\_ADDR\_REQを発行して、ストリームに対応する現在の物理アドレスを取得することによって計算することができます。

DL\_BOUND状態になると、qfeドライバにDL\_UNITDATA\_REQメッセージを送信することによって、ユーザーはEthernet上でフレームを送信することができます。qfeドライバは、DL\_UNITDATA\_INDメッセージのEthernetのタイプと一致するsap値を持ち、開いていて結合されているストリームすべてに、受信されたEthernetフレームを配信します。受信されたEthernetフレームは複製され、必要に応じて複数の開いているスト

リームに配信されます。DL\_UNITDATA\_REQ および DL\_UNITDATA\_IND メッセージに含まれる DLSAP アドレスは、sap (タイプ) と物理 (Ethernet) 部分で構成されます。

qfe ドライバは、強制的なコネクションレス DLPI メッセージセットに加えて、以下の基本式に対応しています。

#### qfe の基本式

DL\_ENABMULTI\_REQ および DL\_DISABMULTI\_REQ 基本式は、個々のマルチキャストグループアドレスの受信を有効または無効にします。これらの基本式を使用して、一群のマルチキャストアドレスをストリームごとに繰り返し作成および変更することができます。これらの基本式は、DL\_ATTACHED 後の状態にある場合に、qfe ドライバによって受け付けられます。

dl\_level フィールド内に DL\_PROMISC\_PHYS フラグが設定された状態での DL\_PROMISCON\_REQ および DL\_PROMISCOFF\_REQ の基本式は、ローカルホストによって生成されたフレームを含む、媒体上のすべての(「プロミスキュアスモード」の)フレームの受信を有効または無効にします。

DL\_PROMISC\_SAP フラグが設定されている状態で使用すると、sap 値 (Ethernet のタイプ) の受信を有効または無効にします。DL\_PROMISC\_MULTI フラグが設定されている状態で使用すると、すべてのマルチキャストグループアドレスの受信を有効または無効にします。どちらの効果も、常にストリームごとに適用され、当該ストリームやその他のストリームの sap および物理レベルの設定の影響は受けません。

DL\_PHYS\_ADDR\_REQ 基本式は、DL\_PHYS\_ADDR\_ACK 基本式の中のストリームに対応する(接続されている)Ethernet アドレス (6 桁の数) を返します。この基本式は、DL\_ATTACH\_REQ の成功後の状態においてのみ有効です。

DL\_SET\_PHYS\_ADDR\_REQ 基本式は、このストリームに対応する(接続されている)Ethernet アドレスを変更します。このストリームを最初に開く処理は、root でのみ行うことができます。root でない場合は、DL\_ERROR\_ACK で EPERM が返されます。この基本式は、このデバイスに接続された現在および未来のすべてのストリームに影響する意味で破壊的です。この基本式がこのストリームに対して成功すると、このデバイスに接続された他のすべてのストリームに M\_ERROR が送信されます。物理アドレスが一度変更されると、その後開かれ、このデバイスに接続されたストリームは、この新しい物理アドレスを取得します。物理アドレスが一度変更されると、物理アドレスを再び変更するためにこの基本式が使用されるか、システムが再起動されるまでこの物理アドレスは存続します。

#### qfe ドライバ

デフォルトでは、qfe ドライバは接続のモードおよび速度を選択するために「自動ネゴシエーション」を実行します。

接続は、以下の4つのモードのいずれかになります。

- 100 Mbps、全二重
- 100 Mbps、半二重
- 10 Mbps、全二重
- 10 Mbps、半二重

これらの速度とモードについては、100Base-TX 規格に記述されています。

