



SunOS リファレンスマニュアル 9 : DDI/DKI カーネル関数

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

Part No: 819-1219-10
2005 年 1 月

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

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.

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, docs.sun.com, AnswerBook, AnswerBook2, JumpStart, Solaris Web Start, Power Management, Sun ONE Application Server, Solaris Flash, Solaris Live Upgrade は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標、登録商標もしくは、サービスマークです。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

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 OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」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 Server/ATOK12」に含まれる郵便番号辞書 (7 桁/5 桁) は郵政事業庁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)。

「ATOK Server/ATOK12」に含まれ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。

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 9: DDI and DK1 Kernel Functions*

Part No: 816-5180-10

Revision A



050111 @ 10536



目次

はじめに 5

序章 9

Intro(9F) 10

ドライバ用のカーネル関数 29

scsi_hba_attach(9F) 30

scsi_hba_attach_setup(9F) 33

scsi_hba_detach(9F) 36

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 (bold) 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 (Italic) 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 (bold) 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。ioctl(2) システムコールへのパラメータは ioctl 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

	<p>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。</p>
オプション	<p>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。</p>
オペランド	<p>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。</p>
出力	<p>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。</p>
戻り値	<p>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 <code>0</code> や <code>-1</code> 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。</p>
エラー	<p>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。</p>
使用法	<p>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。</p>
	<p>コマンド 修飾子 変数 式 入力文法</p>
使用例	<p>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を例にしています。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。</p>
環境	<p>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。</p>

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せん。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 (bold) 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。

序章

Intro(9F)

名前	Intro – DDI/DKI 関数の序章
機能説明	<p>このセクションでは、デバイスドライバで使用可能なカーネル関数について説明します。デバイスドライバインタフェースの概要については、Intro(9E)を参照してください。</p> <p>このセクションでは、各デバイスドライバ関数の情報を次の項目別に記載しています。</p> <ul style="list-style-type: none">■ 「名前」 - 関数の目的を簡単に示します。■ 「形式」 - ソースコードに含まれる関数のエントリポイントの構文を示します。必要なヘッダーには、<code>#include</code> 指示子を示します。■ 「インタフェースレベル」 - すべてのアーキテクチャ依存関係について説明します。■ 「引数」 - 関数を呼び出すのに必要なすべての引数について説明します。■ 「機能説明」 - 関数に関する概要情報を記載します。■ 「戻り値」 - 関数を呼び出した結果の戻り値とメッセージについて説明します。■ 「コンテキスト」 - 関数を呼び出すことのできるドライバコンテキスト (ユーザー、カーネル、割り込み、および高レベルな割り込み) を示します。■ ユーザースレッドにより直接呼び出された場合は、ドライバ関数にユーザーコンテキストがあります。ドライバの <code>read(9E)</code> エントリポイントを <code>read(2)</code> システムコールから呼び出した場合も、ユーザーコンテキストがあります。■ カーネルのほかの部分から呼び出された場合は、ドライバ関数にカーネルコンテキストがあります。ブロックデバイスドライバでは、デバイスにページを書き込むために、<code>strategy(9E)</code> エントリポイントがページデーモンによって呼び出される可能性があります。ページデーモンは現在のユーザーズレッドとは関係がないので、この場合、<code>strategy(9E)</code> がカーネルコンテキストを持ちます。■ 割り込みコンテキストはカーネルコンテキストですが、同時にそれに関連する割り込みレベルも持ちます。ドライバ割り込みルーチンは、割り込みコンテキストを持ちます。 <p>注 - 割り込みコンテキストでも入手できる <code>mutex</code> をユーザーコンテキストまたはカーネルコンテキストで入手した場合、この <code>mutex</code> を持つユーザーコンテキストスレッドまたはカーネルコンテキストスレッドは、この <code>mutex</code> を所有している限り、割り込みコンテキストによって強制されるすべての制限に影響されます。ドライバで <code>mutex</code> を適切に処理する方法については詳しくは、<code>mutex(9F)</code> のマニュアルページを参照してください。</p> <ul style="list-style-type: none">■ 高レベルの割り込みコンテキストは、割り込みコンテキストの、より制限の厳しい形式です。<code>ddi_intr_hilevel(9F)</code> で、割り込みが高レベルの場合、<code>ddi_add_intr(9F)</code> を使ってその割り込みに追加されたドライバ割り込みルーチンは、高レベルの割り込みコンテキストで実行します。このような割り込みルーチンは、<code>ddi_trigger_softintr(9F)</code>、<code>mutex_enter(9F)</code>、および <code>mutex_exit(9F)</code> を呼び出すことしかできません。さらに、<code>mutex_enter(9F)</code> と <code>mutex_exit(9F)</code> は、<code>ddi_get_iblock_cookie(9F)</code> で返された <code>ddi_iblock_cookie</code> で初期化される <code>mutex</code> 上でしか呼び出せない可能性があります。

- 「関連項目」 - 「使用法」とソースから参照される関連する関数で、さらに詳細な情報を参照できます。
- 「使用例」 - ドライバコードで関数を使用する方法を示します。

すべてのドライバには、`<sys/ddi.h>` および `<sys/sunddi.h>` が、ドライバに含まれる最後のファイルとしてこの順序で含まれていなければなりません。

次の表は、このセクションで説明する STREAMS 関数を示します。

STREAMS カーネル関数の概要

ルーチン	種類
adjmsg	DDI/DKI
allocb	DDI/DKI
backq	DDI/DKI
bcanput	DDI/DKI
bcanputnext	DDI/DKI
bufcall	DDI/DKI
canput	DDI/DKI
canputnext	DDI/DKI
clrbuf	DDI/DKI
copyb	DDI/DKI
copymsg	DDI/DKI
datamsg	DDI/DKI
dupb	DDI/DKI
dupmsg	DDI/DKI
enableok	DDI/DKI
esballoc	DDI/DKI
esbcall	DDI/DKI
flushband	DDI/DKI
flushq	DDI/DKI
freeb	DDI/DKI
freemsg	DDI/DKI
freezestr	DDI/DKI
getq	DDI/DKI

Intro(9F)

ルーチン	種類
insq	DDI/DKI
linkb	DDI/DKI
msgdsize	DDI/DKI
msgpullup	DDI/DKI
mt-streams	Solaris DDI
noenable	DDI/DKI
OTHERQ	DDI/DKI
pullupmsg	DDI/DKI
put	DDI/DKI
putbq	DDI/DKI
putctl	DDI/DKI
putctl1	DDI/DKI
putnext	DDI/DKI
putnextctl	DDI/DKI
putq	DDI/DKI
qbufcall	Solaris DDI
qenable	DDI/DKI
qprocson	DDI/DKI
qprocsoff	DDI/DKI
qreply	DDI/DKI
qsize	DDI/DKI
qtimeout	Solaris DDI
qunbufcall	Solaris DDI
quntimeout	Solaris DDI
qwait	Solaris DDI
qwait_sig	Solaris DDI
qwriter	Solaris DDI
RD	DDI/DKI
rmvb	DDI/DKI

ルーチン	種類
rmvq	DDI/DKI
SAMESTR	DDI/DKI
strlog	DDI/DKI
strqget	DDI/DKI
strqset	DDI/DKI
testb	DDI/DKI
unbufcall	DDI/DKI
unfreezestr	DDI/DKI
unlinkb	DDI/DKI
WR	DDI/DKI

次の表は、STREAMS 固有ではない関数を示します。

ルーチン	種類
ASSERT	DDI/DKI
anocancel	Solaris DDI
aphysio	Solaris DDI
bcmp	DDI/DKI
bcopy	DDI/DKI
biodone	DDI/DKI
bioclone	Solaris DDI
biofini	Solaris DDI
bioinit	Solaris DDI
biomodified	Solaris DDI
biosize	Solaris DDI
bioerror	Solaris DDI
bioreset	Solaris DDI
biowait	DDI/DKI
bp_mapin	DDI/DKI
bp_mapout	DDI/DKI

Intro(9F)

ルーチン	種類
btop	DDI/DKI
btopr	DDI/DKI
bzero	DDI/DKI
cmn_err	DDI/DKI
copyin	DDI/DKI
copyout	DDI/DKI
cv_broadcast	Solaris DDI
cv_destroy	Solaris DDI
cv_init	Solaris DDI
cv_signal	Solaris DDI
cv_timedwait	Solaris DDI
cv_wait	Solaris DDI
cv_wait_sig	Solaris DDI
ddi_add_intr	Solaris DDI
ddi_add_softintr	Solaris DDI
ddi_btop	Solaris DDI
ddi_btopr	Solaris DDI
ddi_copyin	Solaris DDI
ddi_copyout	Solaris DDI
ddi_create_minor_node	Solaris DDI
ddi_dev_is_sid	Solaris DDI
ddi_dev_nintrs	Solaris DDI
ddi_dev_nregs	Solaris DDI
ddi_dev_regsize	Solaris DDI
ddi_device_copy	Solaris DDI
ddi_device_zero	Solaris DDI
ddi_devmap_segmap	Solaris DDI
ddi_dma_addr_bind_handle	Solaris DDI
ddi_dma_addr_setup	Solaris DDI

ルーチン	種類
ddi_dma_alloc_handle	Solaris DDI
ddi_dma_buf_bind_handle	Solaris DDI
ddi_dma_buf_setup	Solaris DDI
ddi_dma_burstsizes	Solaris DDI
ddi_dma_coff	Solaris SPARC DDI
ddi_dma_curwin	Solaris SPARC DDI
ddi_dma_devalign	Solaris DDI
ddi_dma_free	Solaris DDI
ddi_dma_free_handle	Solaris DDI
ddi_dma_getwin	Solaris DDI
ddi_dma_htoc	Solaris SPARC DDI
ddi_dma_mem_alloc	Solaris DDI
ddi_dma_mem_free	Solaris DDI
ddi_dma_movwin	Solaris SPARC DDI
ddi_dma_nextcookie	Solaris DDI
ddi_dma_nextseg	Solaris DDI
ddi_dma_nextwin	Solaris DDI
ddi_dma_numwin	Solaris DDI
ddi_dma_segtocookie	Solaris DDI
ddi_dma_set_sbus64	Solaris DDI
ddi_dma_setup	Solaris DDI
ddi_dma_sync	Solaris DDI
ddi_dma_unbind_handle	Solaris DDI
ddi_dmae	Solaris x86 DDI
ddi_dmae_1stparty	Solaris x86 DDI
ddi_dmae_alloc	Solaris x86 DDI
ddi_dmae_disable	Solaris x86 DDI
ddi_dmae_enable	Solaris x86 DDI
ddi_dmae_getattr	Solaris x86 DDI

Intro(9F)

ルーチン	種類
ddi_dmae_getcnt	Solaris x86 DDI
ddi_dmae_getlim	Solaris x86 DDI
ddi_dmae_prog	Solaris x86 DDI
ddi_dmae_release	Solaris x86 DDI
ddi_dmae_stop	Solaris x86 DDI
ddi_enter_critical	Solaris DDI
ddi_exit_critical	Solaris DDI
ddi_ffs	Solaris DDI
ddi_fls	Solaris DDI
ddi_get16	Solaris DDI
ddi_get32	Solaris DDI
ddi_get64	Solaris DDI
ddi_get8	Solaris DDI
ddi_get_cred	Solaris DDI
ddi_get_driver_private	Solaris DDI
ddi_get_iblock_cookie	Solaris DDI
ddi_get_instance	Solaris DDI
ddi_get_name	Solaris DDI
ddi_get_parent	Solaris DDI
ddi_get_soft_iblock_cookie	Solaris DDI
ddi_get_soft_state	Solaris DDI
ddi_getb	Solaris DDI
ddi_getl	Solaris DDI
ddi_getll	Solaris DDI
ddi_getlongprop	Solaris DDI
ddi_getlongprop_buf	Solaris DDI
ddi_getprop	Solaris DDI
ddi_getproplen	Solaris DDI
ddi_getw	Solaris DDI

ルーチン	種類
ddi_intr_hilevel	Solaris DDI
ddi_io_get16	Solaris DDI
ddi_io_get32	Solaris DDI
ddi_io_get8	Solaris DDI
ddi_io_getb	Solaris DDI
ddi_io_getl	Solaris DDI
ddi_io_getw	Solaris DDI
ddi_io_put16	Solaris DDI
ddi_io_put32	Solaris DDI
ddi_io_put8	Solaris DDI
ddi_io_putb	Solaris DDI
ddi_io_putl	Solaris DDI
ddi_io_putw	Solaris DDI
ddi_io_rep_get16	Solaris DDI
ddi_io_rep_get32	Solaris DDI
ddi_io_rep_get8	Solaris DDI
ddi_io_rep_getb	Solaris DDI
ddi_io_rep_getl	Solaris DDI
ddi_io_rep_getw	Solaris DDI
ddi_io_rep_put16	Solaris DDI
ddi_io_rep_put32	Solaris DDI
ddi_io_rep_put8	Solaris DDI
ddi_io_rep_putb	Solaris DDI
ddi_io_rep_putl	Solaris DDI
ddi_io_rep_putw	Solaris DDI
ddi_iomin	Solaris DDI
ddi_iopb_alloc	Solaris DDI
ddi_iopb_free	Solaris DDI
ddi_map_regs	Solaris DDI

Intro(9F)

ルーチン	種類
ddi_mapdev_intercept	Solaris DDI
ddi_mapdev_nointercept	Solaris DDI
ddi_mapdev_set_device_acc_attr	Solaris DDI
ddi_mem_alloc	Solaris DDI
ddi_mem_free	Solaris DDI
ddi_mem_get16	Solaris DDI
ddi_mem_get32	Solaris DDI
ddi_mem_get64	Solaris DDI
ddi_mem_get8	Solaris DDI
ddi_mem_getb	Solaris DDI
ddi_mem_getl	Solaris DDI
ddi_mem_getll	Solaris DDI
ddi_mem_getw	Solaris DDI
ddi_mem_put16	Solaris DDI
ddi_mem_put32	Solaris DDI
ddi_mem_put64	Solaris DDI
ddi_mem_put8	Solaris DDI
ddi_mem_putb	Solaris DDI
ddi_mem_putl	Solaris DDI
ddi_mem_putll	Solaris DDI
ddi_mem_putw	Solaris DDI
ddi_mem_rep_get16	Solaris DDI
ddi_mem_rep_get32	Solaris DDI
ddi_mem_rep_get64	Solaris DDI
ddi_mem_rep_get8	Solaris DDI
ddi_mem_rep_getb	Solaris DDI
ddi_mem_rep_getl	Solaris DDI
ddi_mem_rep_getll	Solaris DDI
ddi_mem_rep_getw	Solaris DDI

ルーチン	種類
ddi_mem_rep_put16	Solaris DDI
ddi_mem_rep_put32	Solaris DDI
ddi_mem_rep_put64	Solaris DDI
ddi_mem_rep_put8	Solaris DDI
ddi_mem_rep_putb	Solaris DDI
ddi_mem_rep_putl	Solaris DDI
ddi_mem_rep_putll	Solaris DDI
ddi_mem_rep_putw	Solaris DDI
ddi_mmap_get_model	Solaris DDI
ddi_model_convert_from	Solaris DDI
ddi_node_name	Solaris DDI
ddi_peek16	Solaris DDI
ddi_peek32	Solaris DDI
ddi_peek64	Solaris DDI
ddi_peek8	Solaris DDI
ddi_peekc	Solaris DDI
ddi_peekd	Solaris DDI
ddi_peekl	Solaris DDI
ddi_peeks	Solaris DDI
ddi_poke16	Solaris DDI
ddi_poke32	Solaris DDI
ddi_poke64	Solaris DDI
ddi_poke8	Solaris DDI
ddi_pokec	Solaris DDI
ddi_poked	Solaris DDI
ddi_pokel	Solaris DDI
ddi_pokes	Solaris DDI
ddi_prop_create	Solaris DDI
ddi_prop_exists	Solaris DDI

Intro(9F)

ルーチン	種類
ddi_prop_free	Solaris DDI
ddi_prop_get_int	Solaris DDI
ddi_prop_lookup	Solaris DDI
ddi_prop_lookup_byte_array	Solaris DDI
ddi_prop_lookup_int_array	Solaris DDI
ddi_prop_lookup_string	Solaris DDI
ddi_prop_lookup_string_array	Solaris DDI
ddi_prop_modify	Solaris DDI
ddi_prop_op	Solaris DDI
ddi_prop_remove	Solaris DDI
ddi_prop_remove_all	Solaris DDI
ddi_prop_undefine	Solaris DDI
ddi_prop_update	Solaris DDI
ddi_prop_update_byte_array	Solaris DDI
ddi_prop_update_int	Solaris DDI
ddi_prop_update_int_array	Solaris DDI
ddi_prop_update_string	Solaris DDI
ddi_prop_update_string_array	Solaris DDI
ddi_ptob	Solaris DDI
ddi_put16	Solaris DDI
ddi_put32	Solaris DDI
ddi_put64	Solaris DDI
ddi_put8	Solaris DDI
ddi_putb	Solaris DDI
ddi_putl	Solaris DDI
ddi_putll	Solaris DDI
ddi_putw	Solaris DDI
ddi_regs_map_free	Solaris DDI
ddi_regs_map_setup	Solaris DDI

ルーチン	種類
<code>ddi_remove_intr</code>	Solaris DDI
<code>ddi_remove_minor_node</code>	Solaris DDI
<code>ddi_remove_softintr</code>	Solaris DDI
<code>ddi_rep_get16</code>	Solaris DDI
<code>ddi_rep_get32</code>	Solaris DDI
<code>ddi_rep_get64</code>	Solaris DDI
<code>ddi_rep_get8</code>	Solaris DDI
<code>ddi_rep_getb</code>	Solaris DDI
<code>ddi_rep_getl</code>	Solaris DDI
<code>ddi_rep_getll</code>	Solaris DDI
<code>ddi_rep_getw</code>	Solaris DDI
<code>ddi_rep_put16</code>	Solaris DDI
<code>ddi_rep_put32</code>	Solaris DDI
<code>ddi_rep_put64</code>	Solaris DDI
<code>ddi_rep_put8</code>	Solaris DDI
<code>ddi_rep_putb</code>	Solaris DDI
<code>ddi_rep_putl</code>	Solaris DDI
<code>ddi_rep_putll</code>	Solaris DDI
<code>ddi_rep_putw</code>	Solaris DDI
<code>ddi_report_dev</code>	Solaris DDI
<code>ddi_root_node</code>	Solaris DDI
<code>ddi_segmap</code>	Solaris DDI
<code>ddi_segmap_setup</code>	Solaris DDI
<code>ddi_set_driver_private</code>	Solaris DDI
<code>ddi_slaveonly</code>	Solaris DDI
<code>ddi_soft_state</code>	Solaris DDI
<code>ddi_soft_state_fini</code>	Solaris DDI
<code>ddi_soft_state_free</code>	Solaris DDI
<code>ddi_soft_state_init</code>	Solaris DDI

Intro(9F)

ルーチン	種類
ddi_soft_state_zalloc	Solaris DDI
ddi_trigger_softintr	Solaris DDI
ddi_umem_alloc	Solaris DDI
ddi_umem_free	Solaris DDI
ddi_unmap_regs	Solaris DDI
delay	DDI/DKI
devmap_default_access	Solaris DDI
devmap_devmem_setup	Solaris DDI
devmap_do_ctxmgt	Solaris DDI
devmap_load	Solaris DDI
devmap_set_ctx_timeout	Solaris DDI
devmap_setup	Solaris DDI
devmap_umem_setup	Solaris DDI
devmap_unload	Solaris DDI
disksort	Solaris DDI
drv_getparm	DDI/DKI
drv_hztousec	DDI/DKI
drv_priv	DDI/DKI
drv_usectohz	DDI/DKI
drv_usecwait	DDI/DKI
free_pktiopb	Solaris DDI
freerbuf	DDI/DKI
get_pktiopb	Solaris DDI
geterror	DDI/DKI
getmajor	DDI/DKI
getminor	DDI/DKI
getrbuf	DDI/DKI
hat_getkpfnum	DKI only
inb	Solaris x86 DDI

ルーチン	種類
inl	Solaris x86 DDI
inw	Solaris x86 DDI
kmem_alloc	DDI/DKI
kmem_free	DDI/DKI
kmem_zalloc	DDI/DKI
kstat_create	Solaris DDI
kstat_delete	Solaris DDI
kstat_install	Solaris DDI
kstat_named_init	Solaris DDI
kstat_queue	Solaris DDI
kstat_runq_back_to_waitq	Solaris DDI
kstat_runq_enter	Solaris DDI
kstat_runq_exit	Solaris DDI
kstat_waitq_enter	Solaris DDI
kstat_waitq_exit	Solaris DDI
kstat_waitq_to_runq	Solaris DDI
makecom_g0	Solaris DDI
makecom_g0_s	Solaris DDI
makecom_g1	Solaris DDI
makecom_g5	Solaris DDI
makedevice	DDI/DKI
max	DDI/DKI
min	DDI/DKI
minphys	Solaris DDI
mod_info	Solaris DDI
mod_install	Solaris DDI
mod_remove	Solaris DDI
mutex_destroy	Solaris DDI
mutex_enter	Solaris DDI

Intro(9F)

ルーチン	種類
mutex_exit	Solaris DDI
mutex_init	Solaris DDI
mutex_owned	Solaris DDI
mutex_tryenter	Solaris DDI
nochpoll	Solaris DDI
nodev	DDI/DKI
nulldev	DDI/DKI
numtos	Solaris DDI
outb	Solaris x86 DDI
outl	Solaris x86 DDI
outw	Solaris x86 DDI
pci_config_get16	Solaris DDI
pci_config_get32	Solaris DDI
pci_config_get64	Solaris DDI
pci_config_get8	Solaris DDI
pci_config_getb	Solaris DDI
pci_config_getl	Solaris DDI
pci_config_getw	Solaris DDI
pci_config_put16	Solaris DDI
pci_config_put32	Solaris DDI
pci_config_put64	Solaris DDI
pci_config_put8	Solaris DDI
pci_config_putb	Solaris DDI
pci_config_putl	Solaris DDI
pci_config_putw	Solaris DDI
pci_config_setup	Solaris DDI
pci_config_teardown	Solaris DDI
physio	Solaris DDI
pollwakeup	DDI/DKI

ルーチン	種類
proc_ref	Solaris DDI
proc_signal	Solaris DDI
proc_unref	Solaris DDI
ptob	DDI/DKI
repinsb	Solaris x86 DDI
repinsd	Solaris x86 DDI
repinsw	Solaris x86 DDI
repoutsb	Solaris x86 DDI
repoutsd	Solaris x86 DDI
repoutsw	Solaris x86 DDI
rmalloc	DDI/DKI
rmalloc_wait	DDI/DKI
rmallocmap	DDI/DKI
rmallocmap_wait	DDI/DKI
rmfree	DDI/DKI
rmfreemap	DDI/DKI
rw_destroy	Solaris DDI
rw_downgrade	Solaris DDI
rw_enter	Solaris DDI
rw_exit	Solaris DDI
rw_init	Solaris DDI
rw_read_locked	Solaris DDI
rw_tryenter	Solaris DDI
rw_tryupgrade	Solaris DDI
scsi_abort	Solaris DDI
scsi_alloc_consistent_buf	Solaris DDI
scsi_cname	Solaris DDI
scsi_destroy_pkt	Solaris DDI
scsi_dmafree	Solaris DDI

Intro(9F)

ルーチン	種類
scsi_dmaget	Solaris DDI
scsi_dname	Solaris DDI
scsi_errmsg	Solaris DDI
scsi_free_consistent_buf	Solaris DDI
scsi_hba_attach	Solaris DDI
scsi_hba_attach_setup	Solaris DDI
scsi_hba_detach	Solaris DDI
scsi_hba_fini	Solaris DDI
scsi_hba_init	Solaris DDI
scsi_hba_lookup_capstr	Solaris DDI
scsi_hba_pkt_alloc	Solaris DDI
scsi_hba_pkt_free	Solaris DDI
scsi_hba_probe	Solaris DDI
scsi_hba_tran_alloc	Solaris DDI
scsi_hba_tran_free	Solaris DDI
scsi_ifgetcap	Solaris DDI
scsi_ifsetcap	Solaris DDI
scsi_init_pkt	Solaris DDI
scsi_log	Solaris DDI
scsi_mname	Solaris DDI
scsi_pktalloc	Solaris DDI
scsi_pktfree	Solaris DDI
scsi_poll	Solaris DDI
scsi_probe	Solaris DDI
scsi_resalloc	Solaris DDI
scsi_reset	Solaris DDI
scsi_reset_notify	Solaris DDI
scsi_resfree	Solaris DDI
scsi_rname	Solaris DDI

ルーチン	種類
scsi_slave	Solaris DDI
scsi_sname	Solaris DDI
scsi_sync_pkt	Solaris DDI
scsi_transport	Solaris DDI
scsi_unprobe	Solaris DDI
scsi_unslave	Solaris DDI
sema_destroy	Solaris DDI
sema_init	Solaris DDI
sema_p	Solaris DDI
sema_p_sig	Solaris DDI
sema_try	Solaris DDI
sema_v	Solaris DDI
sprintf	Solaris DDI
stoi	Solaris DDI
strchr	Solaris DDI
strcmp	Solaris DDI
strcpy	Solaris DDI
strlen	Solaris DDI
strncmp	Solaris DDI
strncpy	Solaris DDI
swab	DDI/DKI
timeout	DDI/DKI
uiomove	DDI/DKI
untimeout	DDI/DKI
ureadc	DDI/DKI
uwritec	DDI/DKI
va_arg	Solaris DDI
va_end	Solaris DDI
va_start	Solaris DDI

Intro(9F)

	ルーチン	種類
	vcmn_err	DDI/DKI
	vsprintf	Solaris DDI
関連項目	Intro(9E), mutex(9F)	

ドライバ用のカーネル関数

scsi_hba_attach(9F)

名前	scsi_hba_attach_setup, scsi_hba_attach, scsi_hba_detach – SCSI HBA 接続および切り離しルーチン
形式	<pre>#include <sys/scsi/scsi.h> int scsi_hba_attach_setup(dev_info_t *dip, ddi_dma_attr_t *hba_dma_attr, scsi_hba_tran_t *hba_tran, int hba_flags); int scsi_hba_attach(dev_info_t *dip, ddi_dma_lim_t *hba_lim, scsi_hba_tran_t *hba_tran, int hba_flags, void *hba_options); int scsi_hba_detach(dev_info_t *dip);</pre>
インタフェースレベル パラメータ	<p>Solaris アーキテクチャ固有 (Solaris DDI).</p> <p><i>dip</i> dev_info_t 構造に対するポインタで、HBA デバイスのインスタンスを参照します。</p> <p><i>hba_lim</i> ddi_dma_lim(9S) 構造に対するポインタ。</p> <p><i>hba_tran</i> scsi_hba_tran(9S) 構造に対するポインタ。</p> <p><i>hba_flags</i> フラグ修飾子。唯一の定義済みフラグ値は、SCSI_HBA_TRAN_CLONE です。</p> <p><i>hba_options</i> 将来の拡張に備えて HBA ドライバによって提供されるオプションの機能で、NULL でなければなりません。</p> <p><i>hba_dma_attr</i> ddi_dma_attr(9S) 構造に対するポインタ。</p>
機能説明	<p>scsi_hba_attach_setup() は、scsi_hba_attach() に代わる推奨インタフェースです。</p> <p>scsi_hba_attach_setup() および scsi_hba_attach() に関して：</p> <p>scsi_hba_attach() は、DMA 限界値 <i>hba_lim</i> と、<i>dip</i> によって定義される HBA デバイスの各インスタンスの転送ベクトル <i>hba_tran</i> を登録します。</p> <p>scsi_hba_attach_setup() は、DMA 属性 <i>hba_dma_attr</i> と、<i>dip</i> によって定義される HBA デバイスの各インスタンスの転送ベクトル <i>hba_tran</i> を登録します。HBA ドライバは、異なる DMA 限界値または DMA 属性、および必要に応じてデバイスの各インタフェースに対する転送ベクトルを渡すことができ、HBA 自身によって課せられたいかなる制約にも対応します。</p> <p>scsi_hba_attach() および scsi_hba_attach_setup() は、dev_ops(9S) 構造内の dev_bus_ops フィールドを使用します。HBA ドライバは、scsi_hba_attach() または scsi_hba_attach_setup() を呼び出す前に、このフィールドを NULL に初期化する必要があります。</p> <p>SCSI_HBA_TRAN_CLONE が <i>hba_flags</i> で要求される場合、<i>hba_tran</i> 構造は、各ターゲットが HBA に接続されるたびに 1 回複製されます。構造の複製は、ターゲットを初期化するために tran_tgt_init(9E) エントリポイントが呼び出される前に行われます。後続のすべての HBA エントリポイント (tran_tgt_init(9E) も含めて) で、引数として渡された、または scsi_address 構造で検出された scsi_hba_tran_t</p>

構造は、「複製された」 `scsi_hba_tran_t` 構造になるので、HBA は、`scsi_hba_tran_t` の `tran_tgt_private` フィールドを使用してターゲットごとのデータを指定できます。HBA は、切り離しの際に割り当てられたものと同じ `scsi_hba_tran_t` 構造だけを注意して解放する必要があります。システムによって割り当てられたすべての「複製された」 `scsi_hba_tran_t` 構造は、システムによって解放されます。

`scsi_hba_attach()` および `scsi_hba_attach_setup()` は、同じ名前の属性がすでにそのノードに接続されていない限り、多数の整数値属性を `dip` に接続します。HBA ドライバは、これらの構成パラメータを `ddi_prop_get_int(9F)` を使用して検出し、HBA を提供した機能に対するあらゆる設定に従う必要があります。

scsi-options

省略可能な SCSI 構成ビット

SCSI_OPTIONS_DR

設定しない場合、HBA はターゲットデバイスに切り離し (Disconnect) 特権を認可しません。

SCSI_OPTIONS_LINK

設定しない場合、HBA はリンクされたコマンド (Linked Commands) を有効にしません。

SCSI_OPTIONS_TAG

設定しない場合、HBA は、コマンドタグ付き待ち行列 (Command Tagged Queing) モードでは動作しません。

SCSI_OPTIONS_PARITY

設定しない場合、HBA はパリティモードでは作動しません。

SCSI_OPTIONS_QAS

設定しない場合、HBA はクイックアービトレーション選択 (Quick Arbitration Select) 機能を利用しません。ご使用のマシンが QAS をサポートするかどうかは、サンのハードウェアマニュアルで確認してください。

SCSI_OPTIONS_FAST

設定しない場合、HBA は、バスを FAST SCSI モードで機能させません。

SCSI_OPTIONS_FAST20

設定しない場合、HBA は、バスを FAST20 SCSI モードで機能させません。

SCSI_OPTIONS_FAST40

設定しない場合、HBA は、バスを FAST40 SCSI モードで機能させません。

SCSI_OPTIONS_FAST80

設定しない場合、HBA は、バスを FAST80 SCSI モードで機能させません。

SCSI_OPTIONS_FAST160

設定しない場合、HBA は、バスを FAST160 SCSI モードで機能させません。

SCSI_OPTIONS_FAST320

設定しない場合、HBA は、バスを FAST320 SCSI モードで機能させません。

SCSI_OPTIONS_WIDE

設定しない場合、HBA は、バスを WIDE SCSI モードで機能させません。

scsi_hba_attach(9F)

	<p>SCSI_OPTIONS_SYNC 設定しない場合、HBA は、バスを同期転送モードで機能させません。</p> <p>scsi-reset-delay SCSI バスまたはデバイスのリセット回復時間 (ミリ秒)。</p> <p>scsi-selection-timeout デフォルトの SCSI 選択フェーズタイムアウト値 (ミリ秒)。HBA 固有の情報については、HBA の各マニュアルページを参照してください。</p> <p>scsi_hba_detach() に関して：</p> <p>scsi_hba_detach() は、DMA 限界値または属性の構造および HBA ドライバの指定されたインスタンスに対する転送ベクトルへの参照を除去します。</p>
戻り値	scsi_hba_attach()、scsi_hba_attach_setup()、および scsi_hba_detach() は、関数の呼び出しに成功すると DDI_SUCCESS を返し、失敗すると DDI_FAILURE を返します。
コンテキスト	scsi_hba_attach() および scsi_hba_attach_setup() は、attach(9E) から呼び出されます。scsi_hba_detach() は、detach(9E) から呼び出されます。
関連項目	attach(9E), detach(9E), tran_tgt_init(9E), ddi_prop_get_int(9F), ddi_dma_attr(9S), ddi_dma_lim(9S), dev_ops(9S), scsi_address(9S)、および scsi_hba_tran(9S) 『Writing Device Drivers』
注意事項	scsi_hba_detach() が呼び出された後に、SCSI ターゲットデバイスドライバの代わりに転送要求が行われないようにするのは HBA ドライバの役割です。 scsi_hba_attach() 関数は、古く、将来のリリースでは廃止されます。この関数は、scsi_hba_attach_setup() で置き換えられます。

scsi_hba_attach_setup(9F)

名前	scsi_hba_attach_setup, scsi_hba_attach, scsi_hba_detach – SCSI HBA 接続および切り離しルーチン
形式	<pre>#include <sys/scsi/scsi.h> int scsi_hba_attach_setup(dev_info_t *dip, ddi_dma_attr_t *hba_dma_attr, scsi_hba_tran_t *hba_tran, int hba_flags); int scsi_hba_attach(dev_info_t *dip, ddi_dma_lim_t *hba_lim, scsi_hba_tran_t *hba_tran, int hba_flags, void *hba_options); int scsi_hba_detach(dev_info_t *dip);</pre>
インタフェースレベル パラメータ	<p>Solaris アーキテクチャ固有 (Solaris DDI).</p> <p><i>dip</i> dev_info_t 構造に対するポインタで、HBA デバイスのインスタンスを参照します。</p> <p><i>hba_lim</i> ddi_dma_lim(9S) 構造に対するポインタ。</p> <p><i>hba_tran</i> scsi_hba_tran(9S) 構造に対するポインタ。</p> <p><i>hba_flags</i> フラグ修飾子。唯一の定義済みフラグ値は、SCSI_HBA_TRAN_CLONE です。</p> <p><i>hba_options</i> 将来の拡張に備えて HBA ドライバによって提供されるオプションの機能で、NULL でなければなりません。</p> <p><i>hba_dma_attr</i> ddi_dma_attr(9S) 構造に対するポインタ。</p>
機能説明	<p>scsi_hba_attach_setup() は、scsi_hba_attach() に代わる推奨インタフェースです。</p> <p>scsi_hba_attach_setup() および scsi_hba_attach() に関して：</p> <p>scsi_hba_attach() は、DMA 限界値 <i>hba_lim</i> と、<i>dip</i> によって定義される HBA デバイスの各インスタンスの転送ベクトル <i>hba_tran</i> を登録します。</p> <p>scsi_hba_attach_setup() は、DMA 属性 <i>hba_dma_attr</i> と、<i>dip</i> によって定義される HBA デバイスの各インスタンスの転送ベクトル <i>hba_tran</i> を登録します。HBA ドライバは、異なる DMA 限界値または DMA 属性、および必要に応じてデバイスの各インタフェースに対する転送ベクトルを渡すことができ、HBA 自身によって課せられたいかなる制約にも対応します。</p> <p>scsi_hba_attach() および scsi_hba_attach_setup() は、dev_ops(9S) 構造内の dev_bus_ops フィールドを使用します。HBA ドライバは、scsi_hba_attach() または scsi_hba_attach_setup() を呼び出す前に、このフィールドを NULL に初期化する必要があります。</p> <p>SCSI_HBA_TRAN_CLONE が <i>hba_flags</i> で要求される場合、<i>hba_tran</i> 構造は、各ターゲットが HBA に接続されるたびに 1 回複製されます。構造の複製は、ターゲットを初期化するために tran_tgt_init(9E) エントリポイントが呼び出される前に行われます。後続のすべての HBA エントリポイント (tran_tgt_init(9E) も含めて) で、引数として渡された、または scsi_address 構造で検出された scsi_hba_tran_t</p>

scsi_hba_attach_setup(9F)

構造は、「複製された」 `scsi_hba_tran_t` 構造になるので、HBA は、`scsi_hba_tran_t` の `tran_tgt_private` フィールドを使用してターゲットごとのデータを指定できます。HBA は、切り離しの際に割り当てられたものと同じ `scsi_hba_tran_t` 構造だけを注意して解放する必要があります。システムによって割り当てられたすべての「複製された」 `scsi_hba_tran_t` 構造は、システムによって解放されます。

`scsi_hba_attach()` および `scsi_hba_attach_setup()` は、同じ名前の属性がすでにそのノードに接続されていない限り、多数の整数値属性を `dip` に接続します。HBA ドライバは、これらの構成パラメータを `ddi_prop_get_int(9F)` を使用して検出し、HBA を提供した機能に対するあらゆる設定に従う必要があります。

scsi-options

省略可能な SCSI 構成ビット

SCSI_OPTIONS_DR

設定しない場合、HBA はターゲットデバイスに切り離し (Disconnect) 特権を認可しません。

SCSI_OPTIONS_LINK

設定しない場合、HBA はリンクされたコマンド (Linked Commands) を有効にしません。

SCSI_OPTIONS_TAG

設定しない場合、HBA は、コマンドタグ付き待ち行列 (Command Tagged Queing) モードでは動作しません。

SCSI_OPTIONS_PARITY

設定しない場合、HBA はパリティモードでは作動しません。

SCSI_OPTIONS_QAS

設定しない場合、HBA はクイックアービトレーション選択 (Quick Arbitration Select) 機能を利用しません。ご使用のマシンが QAS をサポートするかどうかは、サンのハードウェアマニュアルで確認してください。

SCSI_OPTIONS_FAST

設定しない場合、HBA は、バスを FAST SCSI モードで機能させません。

SCSI_OPTIONS_FAST20

設定しない場合、HBA は、バスを FAST20 SCSI モードで機能させません。

SCSI_OPTIONS_FAST40

設定しない場合、HBA は、バスを FAST40 SCSI モードで機能させません。

SCSI_OPTIONS_FAST80

設定しない場合、HBA は、バスを FAST80 SCSI モードで機能させません。

SCSI_OPTIONS_FAST160

設定しない場合、HBA は、バスを FAST160 SCSI モードで機能させません。

SCSI_OPTIONS_FAST320

設定しない場合、HBA は、バスを FAST320 SCSI モードで機能させません。

SCSI_OPTIONS_WIDE

設定しない場合、HBA は、バスを WIDE SCSI モードで機能させません。

scsi_hba_attach_setup(9F)

	<p>SCSI_OPTIONS_SYNC 設定しない場合、HBA は、バスを同期転送モードで機能させません。</p> <p>scsi-reset-delay SCSI バスまたはデバイスのリセット回復時間 (ミリ秒)。</p> <p>scsi-selection-timeout デフォルトの SCSI 選択フェーズタイムアウト値 (ミリ秒)。HBA 固有の情報については、HBA の各マニュアルページを参照してください。</p> <p>scsi_hba_detach() に関して：</p> <p>scsi_hba_detach() は、DMA 限界値または属性の構造および HBA ドライバの指定されたインスタンスに対する転送ベクトルへの参照を除去します。</p>
戻り値	<p>scsi_hba_attach(), scsi_hba_attach_setup(), および scsi_hba_detach() は、関数の呼び出しに成功すると DDI_SUCCESS を返し、失敗すると DDI_FAILURE を返します。</p>
コンテキスト	<p>scsi_hba_attach() および scsi_hba_attach_setup() は、attach(9E) から呼び出されます。scsi_hba_detach() は、detach(9E) から呼び出されます。</p>
関連項目	<p>attach(9E), detach(9E), tran_tgt_init(9E), ddi_prop_get_int(9F), ddi_dma_attr(9S), ddi_dma_lim(9S), dev_ops(9S), scsi_address(9S)、および scsi_hba_tran(9S)</p> <p>『Writing Device Drivers』</p>
注意事項	<p>scsi_hba_detach() が呼び出された後に、SCSI ターゲットデバイスドライバの代わりに転送要求が行われないようにするのは HBA ドライバの役割です。</p> <p>scsi_hba_attach() 関数は、古く、将来のリリースでは廃止されます。この関数は、scsi_hba_attach_setup() で置き換えられます。</p>

scsi_hba_detach(9F)

名前	scsi_hba_attach_setup, scsi_hba_attach, scsi_hba_detach – SCSI HBA 接続および切り離しルーチン
形式	<pre>#include <sys/scsi/scsi.h> int scsi_hba_attach_setup(dev_info_t *dip, ddi_dma_attr_t *hba_dma_attr, scsi_hba_tran_t *hba_tran, int hba_flags); int scsi_hba_attach(dev_info_t *dip, ddi_dma_lim_t *hba_lim, scsi_hba_tran_t *hba_tran, int hba_flags, void *hba_options); int scsi_hba_detach(dev_info_t *dip);</pre>
インタフェースレベル パラメータ	<p>Solaris アーキテクチャ固有 (Solaris DDI).</p> <p><i>dip</i> dev_info_t 構造に対するポインタで、HBA デバイスのインスタンスを参照します。</p> <p><i>hba_lim</i> ddi_dma_lim(9S) 構造に対するポインタ。</p> <p><i>hba_tran</i> scsi_hba_tran(9S) 構造に対するポインタ。</p> <p><i>hba_flags</i> フラグ修飾子。唯一の定義済みフラグ値は、SCSI_HBA_TRAN_CLONE です。</p> <p><i>hba_options</i> 将来の拡張に備えて HBA ドライバによって提供されるオプションの機能で、NULL でなければなりません。</p> <p><i>hba_dma_attr</i> ddi_dma_attr(9S) 構造に対するポインタ。</p>
機能説明	<p>scsi_hba_attach_setup() は、scsi_hba_attach() に代わる推奨インタフェースです。</p> <p>scsi_hba_attach_setup() および scsi_hba_attach() に関して：</p> <p>scsi_hba_attach() は、DMA 限界値 <i>hba_lim</i> と、<i>dip</i> によって定義される HBA デバイスの各インスタンスの転送ベクトル <i>hba_tran</i> を登録します。</p> <p>scsi_hba_attach_setup() は、DMA 属性 <i>hba_dma_attr</i> と、<i>dip</i> によって定義される HBA デバイスの各インスタンスの転送ベクトル <i>hba_tran</i> を登録します。HBA ドライバは、異なる DMA 限界値または DMA 属性、および必要に応じてデバイスの各インタフェースに対する転送ベクトルを渡すことができ、HBA 自身によって課せられたいかなる制約にも対応します。</p> <p>scsi_hba_attach() および scsi_hba_attach_setup() は、dev_ops(9S) 構造内の dev_bus_ops フィールドを使用します。HBA ドライバは、scsi_hba_attach() または scsi_hba_attach_setup() を呼び出す前に、このフィールドを NULL に初期化する必要があります。</p> <p>SCSI_HBA_TRAN_CLONE が <i>hba_flags</i> で要求される場合、<i>hba_tran</i> 構造は、各ターゲットが HBA に接続されるたびに 1 回複製されます。構造の複製は、ターゲットを初期化するために tran_tgt_init(9E) エントリポイントが呼び出される前に行われます。後続のすべての HBA エントリポイント (tran_tgt_init(9E) も含めて) で、引数として渡された、または scsi_address 構造で検出された scsi_hba_tran_t</p>

構造は、「複製された」 `scsi_hba_tran_t` 構造になるので、HBA は、`scsi_hba_tran_t` の `tran_tgt_private` フィールドを使用してターゲットごとのデータを指定できます。HBA は、切り離しの際に割り当てられたものと同じ `scsi_hba_tran_t` 構造だけを注意して解放する必要があります。システムによって割り当てられたすべての「複製された」 `scsi_hba_tran_t` 構造は、システムによって解放されます。

`scsi_hba_attach()` および `scsi_hba_attach_setup()` は、同じ名前の属性がすでにそのノードに接続されていない限り、多数の整数値属性を `dip` に接続します。HBA ドライバは、これらの構成パラメータを `ddi_prop_get_int(9F)` を使用して検出し、HBA を提供した機能に対するあらゆる設定に従う必要があります。

scsi-options

省略可能な SCSI 構成ビット

SCSI_OPTIONS_DR

設定しない場合、HBA はターゲットデバイスに切り離し (Disconnect) 特権を認可しません。

SCSI_OPTIONS_LINK

設定しない場合、HBA はリンクされたコマンド (Linked Commands) を有効にしません。

SCSI_OPTIONS_TAG

設定しない場合、HBA は、コマンドタグ付き待ち行列 (Command Tagged Queing) モードでは動作しません。

SCSI_OPTIONS_PARITY

設定しない場合、HBA はパリティモードでは作動しません。

SCSI_OPTIONS_QAS

設定しない場合、HBA はクイックアービトレーション選択 (Quick Arbitration Select) 機能を利用しません。ご使用のマシンが QAS をサポートするかどうかは、サンのハードウェアマニュアルで確認してください。

SCSI_OPTIONS_FAST

設定しない場合、HBA は、バスを FAST SCSI モードで機能させません。

SCSI_OPTIONS_FAST20

設定しない場合、HBA は、バスを FAST20 SCSI モードで機能させません。

SCSI_OPTIONS_FAST40

設定しない場合、HBA は、バスを FAST40 SCSI モードで機能させません。

SCSI_OPTIONS_FAST80

設定しない場合、HBA は、バスを FAST80 SCSI モードで機能させません。

SCSI_OPTIONS_FAST160

設定しない場合、HBA は、バスを FAST160 SCSI モードで機能させません。

SCSI_OPTIONS_FAST320

設定しない場合、HBA は、バスを FAST320 SCSI モードで機能させません。

SCSI_OPTIONS_WIDE

設定しない場合、HBA は、バスを WIDE SCSI モードで機能させません。

scsi_hba_detach(9F)

	<p>SCSI_OPTIONS_SYNC 設定しない場合、HBA は、バスを同期転送モードで機能させません。</p> <p>scsi-reset-delay SCSI バスまたはデバイスのリセット回復時間 (ミリ秒)。</p> <p>scsi-selection-timeout デフォルトの SCSI 選択フェーズタイムアウト値 (ミリ秒)。HBA 固有の情報については、HBA の各マニュアルページを参照してください。</p> <p>scsi_hba_detach() に関して：</p> <p>scsi_hba_detach() は、DMA 限界値または属性の構造および HBA ドライバの指定されたインスタンスに対する転送ベクトルへの参照を除去します。</p>
戻り値	scsi_hba_attach()、scsi_hba_attach_setup()、および scsi_hba_detach() は、関数の呼び出しに成功すると DDI_SUCCESS を返し、失敗すると DDI_FAILURE を返します。
コンテキスト	scsi_hba_attach() および scsi_hba_attach_setup() は、attach(9E) から呼び出されます。scsi_hba_detach() は、detach(9E) から呼び出されます。
関連項目	attach(9E), detach(9E), tran_tgt_init(9E), ddi_prop_get_int(9F), ddi_dma_attr(9S), ddi_dma_lim(9S), dev_ops(9S), scsi_address(9S)、および scsi_hba_tran(9S) 『Writing Device Drivers』
注意事項	scsi_hba_detach() が呼び出された後に、SCSI ターゲットデバイスドライバの代わりに転送要求が行われないようにするのは HBA ドライバの役割です。 scsi_hba_attach() 関数は、古く、将来のリリースでは廃止されます。この関数は、scsi_hba_attach_setup() で置き換えられます。