



JFP リファレンスマニュアル 1: ユーザーコマンド

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

Part No: 816-3995-10
2002 年 5 月

Copyright 2002 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の書面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。

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

本製品に含まれる HG 明朝 L、HG-MincyoL-Sun、HG ゴシック B、および HG-GothicB-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イスマスタをもとに作成されたものです。HG 平成明朝体 W3@X12 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イ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 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) の商標もしくは登録商標です。

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。

本製品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)。

本製品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 '98』に添付のものを使用しています。© 1997 ビレッジセンター

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。

DtComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

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。



020321@2851



目次

はじめに 5

JFP リファレンスマニュアル 1: ユーザーコマンド 9

Intro_jfp(1) 10

atok12(1) 14

atok12migd(1) 15

atok12migs(1) 18

atok12setup(1) 19

cs00toatok(1) 20

euctoibmj(1) 22

euctojis(1) 24

euctosj(1) 26

evftobdf(1) 28

ibmjtoeuc(1) 29

jistoeuc(1) 31

jistosj(1) 33

jpostprint(1) 35

jprconv(1) 38

jtops(1) 40

jtty(1) 42

kanji(1) 43

kkcvtoes00(1) 44

sdtudctool(1) 46

sdtudc_convert(1) 57

sdtudc_extract(1) 59

sdtudc_extract_ps(1) 61

sdtudc_register(1) 63
sjtoeuc(1) 64
sjojis(1) 66
uum(1) 68
Wnn6(1) 70
wnn6setup(1) 71
wnnatod(1) 72
wnnbushu(1) 73
wnndictutil(1) 74
wnndtoa(1) 75
wnnenvutil(1) 76
wnnotow(1) 77
wnnstat(1) 78
wnntouch(1) 79
xjsi(1) 80

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。

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せん。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。特定のデバイスに関する ioctl 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 ioctl は、mtio(7I) のように io 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。
	<p>コマンド</p> <p>修飾子</p> <p>変数</p> <p>式</p> <p>入力文法</p>
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。
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 (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。

JFP リファレンスマニュアル 1: ユー ザーコマンド

Intro_jfp(1)

名前	Intro_jfp, intro_jfp – JFP コマンドおよびアプリケーションプログラムの序章
機能説明	本セクションでは、JFP が提供する コマンドについて アルファベット順に説明します。
その他のセクションについて	<p>詳細は、『JFP リファレンスマニュアル』の各セクションを参照してください。</p> <ul style="list-style-type: none"> ■ JFP システム管理のコマンドについては、本マニュアルのセクション 1M を参照してください。 ■ JFP ファイルの形式については、本マニュアルのセクション 4 を参照してください。 ■ 公式に使用できる JFP ファイルやさまざまな情報については、本マニュアルのセクション 5 を参照してください。
マニュアルページの コマンドの構文	<p>特に説明しないかぎり、マニュアルページの「形式」の項で記述されるコマンドは、以下の構文に従って オプションやその他の引数を受け付けます。そして、以下のように解釈されなければなりません。</p> <p><i>name</i> [-<i>option</i>...] [<i>cmdarg</i>...]</p> <p>[] 必須ではない <i>option</i> (オプション) や <i>cmdarg</i> (引数) を囲みます。</p> <p>... <i>option</i> (オプション) や <i>cmdarg</i> (引数) が複数回発生することを意味します。</p> <p><i>name</i> 実行可能ファイルの名前です。</p> <p>{ } 中括弧で囲まれた、オプションまたは引数 (および両方) は独立しており、括弧内のすべてを 1 つの単位として扱わなければなりません。</p> <p><i>option</i> (常に “-” が先行します。) <i>noargletter</i>... または、<i>argletter optarg</i>[,...]</p> <p><i>noargletter</i> 引数が必要ないオプション 1 文字を表します。複数の <i>noargletter</i> を指定する場合、1 つの “-” の後にまとめて指定できます (後述のルール 5)。</p> <p><i>argletter</i> 引数が必要なオプション 1 文字を表します。</p> <p><i>optarg</i> <i>argletter</i> に必要なオプション引数 (文字列) です。複数の <i>optargs</i> を <i>argletter</i> に指定する場合、コンマで区切る、または、タブか空白文字で区切って引用符で囲まなければなりません (後述のルール)。</p> <p><i>cmdarg</i> “-” で始まらないパス名 (または他のコマンドの引数)。“-” だけを指定すると標準入力を表します。</p> <p>ここで説明するコマンド構文のルールは、既存のコマンドすべてに適用されているわけではありません。しかし、新規のコマンドはすべてこのルールに従う予定です。すべてのシェルプロシージャは <code>getopts(1)</code> を使って、定位置パラメタを構文解析し、オプションが合法かどうかチェックしなければなりません。<code>getopts(1)</code> は、以下に説明するルール 3 から 10 までをサポートします。その他の規則については、コマンド自身がチェックしなければなりません。</p>
コマンド構文規格：ルール	

1. コマンド名 (上記の *name*) は、2 文字から 9 文字までの長さでなければなりません。
2. コマンド名は、小文字と数字だけで構成されなければなりません。
3. オプション名 (上記の *option*) は、1 文字でなければなりません。
4. オプションには “-” が先行しなければなりません。
5. 引数なしのオプションは、1 つの “-” の後に複数個まとめて指定できます。
6. オプションとオプションの最初の引数 (上記の *optarg*) の間は、タブか空白文字で区切らなければなりません。
7. オプションの引数は、必ず指定しなければなりません。
8. オプションに複数のオプションの引数が続く場合、それぞれをコンマで区切る、または、タブか空白文字で区切って引用符で囲まなければなりません (たとえば、`-o xxx,z,yy` や `-o "xxx z yy"` など)。
9. コマンド行上では、オプションはオペランド (上記の *cmdarg*) より前に指定しなければなりません。
10. “-” を使って、オプションの終わりを示すことができます。
11. オプションの相対的な順番は問題になりません。
12. オペランド (上記の *cmdarg*) の相対的な順番は、その位置によって、コマンドが決めた意味に影響します。
13. “-” の前後に空白文字を指定した場合 (“-”だけを指定した場合)、標準入力を表します。

属性 このセクション中にリストされている属性については `attributes(5)` のマニュアルページを参照してください。

関連項目 `getopts(1)`, `wait(1)`, `exit(2)`, `getopt(3C)`, `attributes(5)`

診断 終了時、すべてのコマンドは状態を表す 2 バイトを返します。1 つは、システムから提供され、終了の原因を示します。もう 1 つ (正常な終了において) は、プログラムから提供されます (`exit(2)` を参照)。前のバイトが 0 の場合、正常な終了を表します。後のバイトが 0 の場合、正常な実行を表します。後のバイトがゼロでない場合、間違った引数を指定した、または不良で受け入れることができないデータを指定したなどの障害を示します。このバイトは、「終了コード」、「終了状態」、「リターンコード」などさまざまな呼ばれ方をします。そして、特別な使い方がある場合に限って説明されます。

警告 ヌル文字を含むファイルを処理しているときに、予測していなかった結果を出すコマンドがあります。通常このようなコマンドはテキスト入力行を文字列として扱っているので、行中のヌル文字 (つまり文字列の終端) に出会うと混乱してしまうからです。

コマンド一覧

名前	説明
<code>Intro_jfp(1)</code>	JFP コマンドおよびアプリケーションプログラムの序章
<code>atok12(1)</code>	ATOK12 日本語入力システム

Intro_jfp(1)

atok12migd(1)	ATOK8 から ATOK12 への辞書移行コマンド
atok12migs(1)	ATOK8 から ATOK12 へのスタイル設定移行
atok12setup(1)	ATOK12 を X 環境で使用する日本語入力システムにする
cs00toatok(1)	cs00 ユーザー辞書から ATOK ユーザー辞書への変換
euctoibmj(1)	日本語 EUC、IBM 日本語コード間のコード変換
euctojis(1)	jistoeuc(1) を参照
euctosj(1)	jistoeuc(1) を参照
evftobdf(1)	evfont から X11 BDF 2.1 フォーマットへの変換
ibmjtoeuc(1)	euctoibmj(1) を参照
jistoeuc(1)	JIS、PC 漢字、日本語 EUC 間のコード変換
jistosj(1)	jistoeuc(1) を参照
jpostprint(1)	日本語 PostScript プリンタに日本語テキストを出力するためのフィルタ
jprconv(1)	ドットマトリックス漢字プリンタ、日本語ページプリンタで日本語テキストを印刷するためのフィルタ
jtops(1)	Sun LaserWriter または日本語 PostScript プリンタに日本語ファイルを出力するためのフィルタ
jtty(1)	TTY/STREAMS に対する日本語処理環境の設定または変更
kanji(1)	漢字コードの一覧表を表示
kkcvtoacs00(1)	kkcv ユーザー辞書から cs00 ユーザー辞書への変換
sdtudctool(1)	Solaris 外字ツール
sdtudc_convert(1)	ユーザー定義文字変換ユーティリティ
sdtudc_extract(1)	ユーザー定義文字変換ユーティリティ
sdtudc_extract_ps(1)	ユーザー定義文字変換ユーティリティ
sdtudc_register(1)	ユーザー定義文字辞書登録用中間ユーティリティ
sjtoeuc(1)	jistoeuc(1) を参照
sjtojis(1)	jistoeuc(1) を参照
uum(1)	かな漢字変換フロントエンドプロセッサ
Wnn6(1)	Wnn6 日本語入力システム
wnn6setup(1)	Wnn6 を X 環境で使用する日本語入力システムにする
wnnatod(1)	テキスト形式辞書からバイナリ形式辞書への変換

wnnbushu(1)	Wnn6 部首入力ユーティリティ
wnndictutil(1)	辞書ユーティリティ
wnndtoa(1)	バイナリ形式辞書からテキスト形式辞書への変換
wnnenvutil(1)	環境設定ユーティリティ
wnnotow(1)	ユーザー辞書コンバータ
wnnstat(1)	Wnn6 かな漢字変換サーバーの状態を表示する
wnntouch(1)	ファイルのヘッダーを書き換えて i ノードと合わせる
xjsi(1)	Wnn6 かな漢字変換サーバー/htt インタフェースモジュール

atok12(1)

名前	atok12 – ATOK12 日本語入力システム
機能説明	<p>ATOK12 は、デスクトップ環境で日本語を入力するための手段を提供します。</p> <p>atok12setup(1) を実行すると ATOK12 を使用するように設定を変更できます。その後デスクトップ環境にログインし直すと ATOK12 が使用でき、ワークスペース上に ATOK パレットウィンドウが表示されます。</p> <p>ユーザーは、日本語の入力方法を好みの入力スタイルに設定できます。この場合、ATOK パレットウィンドウから環境設定ユーティリティを起動してください。</p> <p>また、ユーザーごとのかな漢字変換辞書に単語を登録し、使用することができます。この場合、ATOK パレットウィンドウから辞書ユーティリティを起動してください。</p> <p>ATOK12 は X ウィンドウアプリケーションから利用できるほか、IIIMP (Internet-Intranet Input Method Protocol) を使用するクライアントアプリケーションからも利用できます。たとえば Swing インタフェースを使用した Java2 アプリケーションがこれにあたります。</p> <p>かな漢字変換に使用する辞書は、サーバー上で集中管理できます。</p>
関連項目	<p>atok12setup(1), java(1)</p> <p>『日本語入力方式の概要と移行』</p> <p>『ATOK12 ユーザーズガイド』</p>

名前	atok12migd - ATOK8 から ATOK12 への辞書移行コマンド																										
形式	atok12migd [-h <i>part_of_speech</i>] <i>atok8_dic atok12_dic</i>																										
使用条件	JSatsvu																										
機能説明	atok12migd は、ATOK8 の辞書の内容を ATOK12 の辞書にマージします。ユーザーによって登録された単語だけが、ATOK12 の辞書にマージされます。ATOK8 システム提供時に辞書に含まれていた単語はマージされません。マージ元の ATOK8 の辞書の中身は、マージ処理によって破壊されたり変更されたりしません。																										
オプション	<p>-h <i>part_of_speech</i> 品詞指定。品詞を指定することにより、マージする単語を限定することができます。品詞は ATOK8 の品詞番号で指定してください。複数の品詞を指定する場合は + で区切ってください。all を指定した場合には、すべての品詞の単語がマージの対象となります。</p> <p>例：</p> <p style="margin-left: 40px;">1 2+3 3+4+8+9 all</p> <p>ATOK8 の品詞番号</p> <table border="1" style="margin-left: 40px; border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="text-align: left;">番号</th> <th style="text-align: left;">名前</th> </tr> </thead> <tbody> <tr><td>1</td><td>一般名詞</td></tr> <tr><td>2</td><td>固有人名</td></tr> <tr><td>3</td><td>固有地名</td></tr> <tr><td>4</td><td>固有組織</td></tr> <tr><td>5</td><td>固有一般</td></tr> <tr><td>6</td><td>名詞サ変</td></tr> <tr><td>7</td><td>名詞ザ変</td></tr> <tr><td>8</td><td>名詞形動</td></tr> <tr><td>9</td><td>独立語</td></tr> <tr><td>10</td><td>単漢字</td></tr> <tr><td>11</td><td>連体詞</td></tr> <tr><td>12</td><td>接続詞</td></tr> </tbody> </table>	番号	名前	1	一般名詞	2	固有人名	3	固有地名	4	固有組織	5	固有一般	6	名詞サ変	7	名詞ザ変	8	名詞形動	9	独立語	10	単漢字	11	連体詞	12	接続詞
番号	名前																										
1	一般名詞																										
2	固有人名																										
3	固有地名																										
4	固有組織																										
5	固有一般																										
6	名詞サ変																										
7	名詞ザ変																										
8	名詞形動																										
9	独立語																										
10	単漢字																										
11	連体詞																										
12	接続詞																										

atok12migd(1)

番号	名前
13	感動詞
14	接頭語
15	接尾辞
16	数詞
17	カ行五段
18	ガ行五段
19	サ行五段
20	タ行五段
21	ナ行五段
22	ハ行五段
23	バ行五段
24	マ行五段
25	ラ行五段
26	ワ行五段
27	一般動詞
28	カ変動詞
29	サ変動詞
30	ザ変動詞
31	形容詞
32	形容動詞
33	副詞

atok8_dic ATOK8 の辞書。

atok12_dic ATOK12 の辞書。

環境 LANG, LC atok12migd は LC_CTYPE カテゴリがロケール ja またはそれと同等なロケールに指定されている場合に限り動作します。上記の環境変数の使用法については、*environ(5)* を参照してください。

終了ステータス 次の終了値を返します。

0 正常終了

1 理解できないオプション

	127	その他のエラー
ファイル	/usr/bin/atok12migd	ATOK8 から ATOK12 への辞書移行コマンド
関連項目	environ(5)	『日本語入力方式の概要と移行』 『ATOK12 ユーザーズガイド』
注意事項	SIGTERM	シグナルをこのプロセスに送ると終了します。 atok12migd は、ファイル名や内容に EUC コードセット 3 の文字を含むファイルを扱うことができません。

atok12migs(1)

名前	atok12migs - ATOK8 から ATOK12 へのスタイル設定移行
形式	atok12migs [-k -r] <i>atok8_ucf atok12_sty</i>
使用条件	JSatsvu
機能説明	<p>atok12migs コマンドは、<i>atok8_ucf</i> で指定された ATOK8 環境設定ファイル内の機能のキーへの割り当てとローマ字変換の設定内容を、ATOK12 で使用される形式に変換し、<i>atok12_sty</i> で指定された ATOK12 スタイルファイルに出力します。</p> <p><i>atok12_sty</i> が既に存在する場合も、その内容は参照されません。既存の <i>atok12_sty</i> は、変換結果によって上書きされます。</p>
オプション	<p>以下のオプションを指定できます。</p> <p>-k 機能のキーへの割り当てに関する設定内容だけを変換します。ローマ字に関する設定内容は、ATOK12 のデフォルトが使用されます。</p> <p>-r ローマ字に関する設定内容だけを変換します。機能のキーへの割り当てに関する設定内容は、ATOK12 のデフォルトが使用されません。</p>
オペラント	<p>以下のオペラントを指定できます。</p> <p><i>atok8_ucf</i> ATOK8 環境設定ファイル</p> <p><i>atok12_sty</i> 出力先 ATOK12 スタイルファイル</p>
環境	<p>LANG, LC atok12migs は LC_CTYPE カテゴリがロケール ja またはそれと同等のロケールに指定されている場合に限り動作します。上記の環境変数の使用方法については、environ(5) を参照してください。</p>
終了ステータス	<p>次の終了値を返します。</p> <p>0 正常終了</p> <p>1 理解できないオプション</p> <p>127 その他のエラー</p>
ファイル	<p>/usr/bin/atok12migs</p> <p>ATOK8 から ATOK12 へのスタイル設定移行コマンド</p>
関連項目	<p>environ(5)</p> <p>『日本語入力方式の概要と移行』</p> <p>『ATOK12 ユーザーズガイド』</p>

名前	atok12setup – ATOK12 を X 環境で使用する日本語入力システムにする
形式	atok12setup
使用条件	JSatsxw
機能説明	<p>atok12setup は、ATOK12 を X ウィンドウシステム環境でログインした時に起動される日本語入力システムとして設定します。atok12setup で設定した内容は、次にログインした時から有効になります。</p> <p>atok12setup は、次のファイルを編集します。</p> <p><code>\$HOME/.dtprofile</code> 共通デスクトップ環境 (CDE) でログインする時に実行されるスクリプト。atok12(1) を指定して <code>htt(1)</code> を起動する処理を追加します。</p> <p><code>\$HOME/.dtprofile</code> がすでに存在し、Solaris 2.6 およびそれ以降で提供される <code>wnn6setup(1)</code> を使用して日本語入力システムを起動するように設定されている場合、その設定を、ATOK12 を日本語入力システムとして使用する設定で置き換えます。</p> <p><code>\$HOME/.dtprofile</code> および <code>\$HOME/.openwin-init</code> に追加される ATOK12 を日本語入力システムとして使用するための設定は、次のように目印となる行 (コメント) で前後を囲まれます。</p> <pre>###== - Generated by atok12setup to launch XIM for Japanese. == BEGIN == -===### . . . 日本語入力システムの起動の設定 ###== - Generated by atok12setup to launch XIM for Japanese. == END == -===###</pre> <p>これらの目印となる行で囲まれた領域は、atok12setup(1) または wnn6setup(1) を実行することで書き換えられます。したがって、これらの目印の行の内側を編集した後、atok12setup(1) または wnn6setup(1) を実行すると、編集した内容は失われます。また、これらの目印となる行は編集しないでください。</p>
ファイル	<code>\$HOME/.dtprofile</code> <code>\$HOME/.openwin</code> <code>-init</code>
関連項目	atok12(1), dtlogin(1), wnn6setup(1)
注意事項	<p>ユーザーが <code>\$HOME/.dtprofile</code> を編集して、日本語入力システムを設定した場合、atok12setup コマンド実行後に再度ログインし直しても ATOK12 を利用できない場合があります。この場合は、ユーザーが設定した <code>\$HOME/.dtprofile</code> の日本語入力システムの設定を削除してください。</p>

cs00toatok(1)

名前 cs00toatok – cs00 ユーザー辞書から ATOK ユーザー辞書への変換
 形式 **cs00toatok** [*filename...*]
 使用条件 SUNWjfpu
 機能説明 cs00toatok は、cs00 のユーザー辞書に登録されている単語を ATOK12 上で利用するために、cs00 の単語リストファイルの形式から ATOK の辞書ユーティリティで利用できるテキスト形式の単語ファイルへ変換するフィルタです。

cs00toatok は *filename* に指定されたファイルを読み込みます。 *filename* が与えられなければ、標準入力からファイルを読み込みます。そのファイルの内容は、cs00 の単語リストファイルの形式である必要があります。 cs00toatok は、ATOK の辞書ユーティリティで利用できるテキスト形式の単語ファイルを標準出力に出力します。 cs00toatok は個々の単語を次の規則に従って変換します。

読み、単語の読みの文字種および長さ、単語（表記）の文字種および長さを変語（表記） 更しません。

品詞 変換元の単語の品詞を次に示す表に従って変換します。

cs00 における品詞		ATOK における品詞	
:N1	名詞 1	1	一般名詞
:N2	名詞 2	1	一般名詞
:M1	人名 1	2	固有人名
:M2	人名 2	2	固有人名
:T1	地名 1	3	固有地名
:T2	地名 2	3	固有地名
:NM	数詞	16	数詞
:NN	助数詞	15	接尾辞
:PR	接頭詞	14	接頭語
:SF	接尾詞	15	接尾辞
:AD	副詞	33	副詞
:CN	接続詞	12	接続詞
:RT	連体詞	11	連体詞
:AJ	形容詞	31	形容詞
:AV	形容動詞	32	形容動詞
:SH	サ変動詞	6	名詞サ変

:ZH	ザ変動詞	7	名詞ザ変
:1V	一段活用動詞	27	一段動詞
:KV	カ行五段活用動詞	17	カ行五段
:GV	ガ行五段活用動詞	18	ガ行五段
:SV	サ行五段活用動詞	19	サ行五段
:TV	タ行五段活用動詞	20	タ行五段
:NV	ナ行五段活用動詞	21	ナ行五段
:BV	バ行五段活用動詞	23	バ行五段
:MV	マ行五段活用動詞	24	マ行五段
:RV	ラ行五段活用動詞	25	ラ行五段
:WV	ワ行五段活用動詞	26	ワ行五段
:UN	分類不可	-	-
:TK	単漢字	10	単漢字
:BS	文節	-	-

:UN 「分類不可」、:BS 「文節」の品詞をもつ単語はデフォルトでは出力されませんが、-a オプションを指定して出力できます。また、変換元の単語が複数の品詞をもつ場合、cs00toatok はその個々の品詞をもつ独立した複数の単語に変換します。

オプション -a デフォルトで出力される単語に加えて、品詞が「分類不可」または「文節」である単語を品詞未定として出力します。

注意事項 cs00toatok は、読み、単語 (表記) に対しては一切変更しません。したがって、ATOK 辞書に登録する場合は次の点に注意してください。

- 単語の読みの文字種および長さ、単語 (表記) の文字種および長さによっては、ATOK 辞書には登録できない場合があります。
- 単語の表記の両端が「'」(全角の単一引用符) または「"」(全角の二重引用符) であると、ATOK 辞書には両端の文字が取り除かれて登録されます。
- 単語の表記に「、」(全角のコンマ) が含まれていると、ATOK 辞書には登録できません。

ATOK 辞書へ登録するには ATOK12 辞書ユーティリティを使用します。詳細は、『ATOK12 ユーザーズガイド』を参照してください。

関連項目 atok12(1), atok12wordlist(4)

euctoibmj(1)

名前	euctoibmj, ibmjtoeuc – 日本語 EUC、IBM 日本語コード間のコード変換
形式	euctoibmj [-t] [-u <i>code</i>] [-U] [<i>filename...</i>] ibmjtoeuc [-u <i>code</i>] [-U] [<i>filename...</i>]
使用条件	SUNWjfpu
機能説明	euctoibmj は、指定された <i>filename</i> のファイルの内容を ASCII/日本語 EUC から EBCDIC/IBM 日本語コードに変換します。ibmjtoeuc は、指定された <i>filename</i> のファイルの内容を EBCDIC/IBM 日本語コードから ASCII/日本語 EUC に変換します。これらのコマンドはいずれも変換結果を標準出力 (<i>stdout</i>) に出力します。 <i>filename</i> が指定されない場合には、標準入力から読み込まれます。 日本語をハンドルする場合、euctoibmj および ibmjtoeuc コマンドは、日本語 EUC および IBM-Japanese 間の変換のみを行います。日本語 EUC、JIS、および PC 漢字間のコード変換は、jistoec(1) ファミリまたは iconv(1) によってサポートされます。
オプション	-u <i>code</i> このオプションが指定されると、印字文字に変換されないコードは、euctoibmj の場合、4 桁の 16 進数 <i>code</i> を使って指定された IBM 日本語コード (Host Code) に変換され、ibmjtoeuc の場合、4 桁の 10 進数 <i>code</i> で指定された JIS 区点コードに対応する日本語 EUC コードに変換されます。このオプションが指定されない場合、IBM 日本語コード (Host Code) として 4040、JIS 区点コードとして 0101 が採用されます。 -U 出力はバッファリングされません (デフォルトではバッファリングされます)。 -t このオプションが指定されると、euctoibmj は、入力中のコードセット 2 のカタカナをコードセット 1 の対応するカタカナにあらかじめ変換して処理します。このオプションが指定されない場合、コードセット 2 のカタカナは不正なコードとして処理されません。
環境	euctoibmj および ibmjtoeuc の実行時には、環境変数 LC_CTYPE または LANG が、ja またはこれと同等な日本語ロケールに設定されていなくてはなりません。環境変数はこの順番で調べられ、有効なロケールの設定が検知された段階でロケールが決定されます。
ファイル	/usr/lib/jcodetables/ibmj-euc IBM 日本語コード用コード変換テーブル
関連項目	iconv(1), jistoec(1), iconv_ja(5)
診断	unexpected data encountered in input. 入力ファイル中に不正なコードが存在します。
使用上の留意点	ASCII/EBCDIC の変換テーブルは 1968 年 11 月版 CACM の 256 文字標準に基づいています。ibm 変換は、標準的ではありませんが、ある種の IBM 印字トレインの規約に適合しています。すべての場合に満足できるような解決方法はありません。

euctoibmj(1)

日本語 EUC と IBM 日本語コード間の変換テーブルは、JIS X 0201、JIS X 0208-1983、および IBM 漢字コード一覧表 (第 4 版、1987 年 9 月) に基づいています。

JIS X 0212 文字集合が入力された場合、euctoibmj は変換を正しく行えません。

euctojis(1)

名前	jistoeuc, jistosj, euctojis, euctosj, sjtojis, sjtoeuc – JIS、PC 漢字、日本語 EUC 間のコード変換
形式	jistoeuc [-8] [-U] [<i>filename...</i>] jistosj [-8] [-U] [<i>filename...</i>] euctojis [-8] [-U] [<i>filename...</i>] euctosj [-U] [<i>filename...</i>] sjtojis [-8] [-U] [<i>filename...</i>] sjtoeuc [-U] [<i>filename...</i>]
使用条件	SUNWjfpu
機能説明	<p>本コマンド群は、異なるコード規格の文字セットを相互に変換します。コマンド名 [<i>filename...</i>] は、指定されたファイルの内容に対して指定された変換を行い、標準出力に書き出します。</p> <p>ファイルを指定しないと、標準入力から文字を読み取り、変換します。</p> <p>jistoeuc JIS から日本語 EUC への変換 jistosj JIS から PC 漢字への変換 euctojis 日本語 EUC から JIS への変換 euctosj 日本語 EUC から PC 漢字への変換 sjtojis PC 漢字から JIS への変換 sjtoeuc PC 漢字から日本語 EUC への変換</p>
オプション	-8 jistoeuc、jistosj、sjtojis、sjtoeuc は、SI/SO を使わず、8 単位系 JIS で JIS X 0201 文字集合を処理、あるいはこれに変換します。 -U 出力はバッファリングされません (デフォルトではバッファリングされます)。
関連項目	iconv(1), iconv_ja(5)
注意事項	<p>jistoeuc では、文字集合指示エスケープ・シーケンスとして以下を処理することができます。</p> <p>JIS X 0208 指示 – \E\$B、\E\$ (B、\E\$@ JIS X 0212 指示 – \E\$ (D JIS X 0201 Roman 指示 – \E (J、\E (H ASCII 指示 – \E (B</p> <p>sjtojis、euctojis は文字集合指示エスケープ・シーケンスとして以下を出力します。</p>

JIS X 0208 指示 - \E\$B
JIS X 0212 指示 - \E\$(D(sjtojis では表示されません)
JIS X 0201 Roman 指示 - \E(J

本コマンドは、原則として入力ファイル中の各コードが正しいかどうかの確認は行いません。また PC 漢字との相互変換においては「TOG 日本ベンダ協議会推奨日本語 EUC・シフト JIS 間コード変換仕様」に基づくコード変換は行われません。iconv(1) ユーティリティがこれらの機能を提供します。iconv(1) および iconv_ja(5) を参照してください。

使用上の留意点

JIS X 0212 文字集合が入力された場合、jistosj および euctosj は変換を正しく行いません。euctosj、sjtoeuc、jistosj、sjtojis については、JIS X 0208 1 区 - 84 区だけが変換の対象になります。

euctosj(1)

名前	jistoeuc, jistosj, euctojis, euctosj, sjtojis, sjtoeuc – JIS、PC 漢字、日本語 EUC 間のコード変換
形式	jistoeuc [-8] [-U] [<i>filename...</i>] jistosj [-8] [-U] [<i>filename...</i>] euctojis [-8] [-U] [<i>filename...</i>] euctosj [-U] [<i>filename...</i>] sjtojis [-8] [-U] [<i>filename...</i>] sjtoeuc [-U] [<i>filename...</i>]
使用条件	SUNWjfpu
機能説明	<p>本コマンド群は、異なるコード規格の文字セットを相互に変換します。コマンド名 [<i>filename...</i>] は、指定されたファイルの内容に対して指定された変換を行い、標準出力に書き出します。</p> <p>ファイルを指定しないと、標準入力から文字を読み取り、変換します。</p> <p>jistoeuc JIS から日本語 EUC への変換 jistosj JIS から PC 漢字への変換 euctojis 日本語 EUC から JIS への変換 euctosj 日本語 EUC から PC 漢字への変換 sjtojis PC 漢字から JIS への変換 sjtoeuc PC 漢字から日本語 EUC への変換</p>
オプション	-8 jistoeuc、jistosj、sjtojis、sjtoeuc は、SI/SO を使わず、8 単位系 JIS で JIS X 0201 文字集合を処理、あるいはこれに変換します。 -U 出力はバッファリングされません (デフォルトではバッファリングされます)。
関連項目	iconv(1), iconv_ja(5)
注意事項	<p>jistoeuc では、文字集合指示エスケープ・シーケンスとして以下を処理することができます。</p> <p>JIS X 0208 指示 – \E\$B、\E\$ (B、\E\$@ JIS X 0212 指示 – \E\$ (D JIS X 0201 Roman 指示 – \E (J、\E (H ASCII 指示 – \E (B</p> <p>sjtojis、euctojis は文字集合指示エスケープ・シーケンスとして以下を出力します。</p>

JIS X 0208 指示 - \E\$B
JIS X 0212 指示 - \E\$(D(sjtojis では表示されません)
JIS X 0201 Roman 指示 - \E(J

本コマンドは、原則として入力ファイル中の各コードが正しいかどうかの確認は行いません。また PC 漢字との相互変換においては「TOG 日本ベンダ協議会推奨日本語 EUC・シフト JIS 間コード変換仕様」に基づくコード変換は行われません。iconv(1) ユーティリティがこれらの機能を提供します。iconv(1) および iconv_ja(5) を参照してください。

使用上の留意点

JIS X 0212 文字集合が入力された場合、jistosj および euctosj は変換を正しく行いません。euctosj、sjtoeuc、jistosj、sjtojis については、JIS X 0208 1 区 - 84 区だけが変換の対象になります。

evftobdf(1)

名前	evftobdf - evfont から X11 BDF 2.1 フォーマットへの変換
形式	evftobdf [-t] [-p <i>propertyfile</i>] <i>filename...</i>
使用条件	SUNWjfpu
機能説明	evftobdf は、X11/NeWS ウィンドウシステムで使用するフォントを作成するために使用し、evfont フォントフォーマットから X11 BDF 2.1 フォントフォーマットへ変換します。
オプション	<p>-t BDF フォントからプロパティ部分を抽出して標準出力に出力します。フォントのフォーマット変換は行いません。この場合、指定するファイル名のファイルは BDF フォーマットにする必要があります。evf には出力する evfont のファイル名を指定します。</p> <p>-p <i>propertyfile</i> BDF のプロパティ部分を <i>propertyfile</i> から取り込みます。1 つの <i>propertyfile</i> の中に 2 つの文字セットのプロパティを定義することができます。この場合、各文字セットは、{と} のブロックで区切られます。</p>

名前	euctoibmj, ibmjtoeuc – 日本語 EUC、IBM 日本語コード間のコード変換
形式	euctoibmj [-t] [-u <i>code</i>] [-U] [<i>filename...</i>] ibmjtoeuc [-u <i>code</i>] [-U] [<i>filename...</i>]
使用条件	SUNWjfpu
機能説明	euctoibmj は、指定された <i>filename</i> のファイルの内容を ASCII/日本語 EUC から EBCDIC/IBM 日本語コードに変換します。ibmjtoeuc は、指定された <i>filename</i> のファイルの内容を EBCDIC/IBM 日本語コードから ASCII/日本語 EUC に変換します。これらのコマンドはいずれも変換結果を標準出力 (stdout) に出力します。 <i>filename</i> が指定されない場合には、標準入力から読み込まれます。 日本語をハンドルする場合、euctoibmj および ibmjtoeuc コマンドは、日本語 EUC および IBM-Japanese 間の変換のみを行います。日本語 EUC、JIS、および PC 漢字間のコード変換は、jistoec(1) ファミリまたは iconv(1) によってサポートされます。
オプション	-u <i>code</i> このオプションが指定されると、印字文字に変換されないコードは、euctoibmj の場合、4 桁の 16 進数 <i>code</i> を使って指定された IBM 日本語コード (Host Code) に変換され、ibmjtoeuc の場合、4 桁の 10 進数 <i>code</i> で指定された JIS 区点コードに対応する日本語 EUC コードに変換されます。このオプションが指定されない場合、IBM 日本語コード (Host Code) として 4040、JIS 区点コードとして 0101 が採用されます。 -U 出力はバッファリングされません (デフォルトではバッファリングされます)。 -t このオプションが指定されると、euctoibmj は、入力中のコードセット 2 のカタカナをコードセット 1 の対応するカタカナにあらかじめ変換して処理します。このオプションが指定されない場合、コードセット 2 のカタカナは不正なコードとして処理されません。
環境	euctoibmj および ibmjtoeuc の実行時には、環境変数 LC_CTYPE または LANG が、ja またはこれと同等な日本語ロケールに設定されていなくてはなりません。環境変数はこの順番で調べられ、有効なロケールの設定が検知された段階でロケールが決定されます。
ファイル	/usr/lib/jcodetables/ibmj-euc IBM 日本語コード用コード変換テーブル
関連項目	iconv(1), jistoec(1), iconv_ja(5)
診断	unexpected data encountered in input. 入力ファイル中に不正なコードが存在します。
使用上の留意点	ASCII/EBCDIC の変換テーブルは 1968 年 11 月版 CACM の 256 文字標準に基づいています。ibm 変換は、標準的ではありませんが、ある種の IBM 印字トレーンの規約に適合しています。すべての場合に満足できるような解決方法はありません。

ibmjtoeuc(1)

日本語 EUC と IBM 日本語コード間の変換テーブルは、JIS X 0201、JIS X 0208-1983、および IBM 漢字コード一覧表 (第 4 版、1987 年 9 月) に基づいています。

JIS X 0212 文字集合が入力された場合、euctoibmj は変換を正しく行えません。

名前	jistoeuc, jistosj, euctojis, euctosj, sjtojis, sjtoeuc – JIS、PC 漢字、日本語 EUC 間のコード変換
形式	jistoeuc [-8] [-U] [<i>filename...</i>] jistosj [-8] [-U] [<i>filename...</i>] euctojis [-8] [-U] [<i>filename...</i>] euctosj [-U] [<i>filename...</i>] sjtojis [-8] [-U] [<i>filename...</i>] sjtoeuc [-U] [<i>filename...</i>]
使用条件	SUNWjfpu
機能説明	<p>本コマンド群は、異なるコード規格の文字セットを相互に変換します。コマンド名 [<i>filename...</i>] は、指定されたファイルの内容に対して指定された変換を行い、標準出力に書き出します。</p> <p>ファイルを指定しないと、標準入力から文字を読み取り、変換します。</p> <p>jistoeuc JIS から日本語 EUC への変換 jistosj JIS から PC 漢字への変換 euctojis 日本語 EUC から JIS への変換 euctosj 日本語 EUC から PC 漢字への変換 sjtojis PC 漢字から JIS への変換 sjtoeuc PC 漢字から日本語 EUC への変換</p>
オプション	-8 jistoeuc、jistosj、sjtojis、sjtoeuc は、SI/SO を使わず、8 単位系 JIS で JIS X 0201 文字集合を処理、あるいはこれに変換します。 -U 出力はバッファリングされません (デフォルトではバッファリングされます)。
関連項目	iconv(1), iconv_ja(5)
注意事項	<p>jistoeuc では、文字集合指示エスケープ・シーケンスとして以下を処理することができます。</p> <p>JIS X 0208 指示 – \E\$B、\E\$ (B)、\E\$@ JIS X 0212 指示 – \E\$ (D) JIS X 0201 Roman 指示 – \E (J)、\E (H) ASCII 指示 – \E (B)</p> <p>sjtojis、euctojis は文字集合指示エスケープ・シーケンスとして以下を出力しません。</p>

jstoeuc(1)

JIS X 0208 指示 - \E\$B
JIS X 0212 指示 - \E\$(D (sjtojis では表示されません)
JIS X 0201 Roman 指示 - \E(J

本コマンドは、原則として入力ファイル中の各コードが正しいかどうかの確認は行いません。また PC 漢字との相互変換においては「TOG 日本ベンダ協議会推奨日本語 EUC・シフト JIS 間コード変換仕様」に基づくコード変換は行われません。iconv(1) ユーティリティがこれらの機能を提供します。iconv(1) および iconv_ja(5) を参照してください。

使用上の留意点

JIS X 0212 文字集合が入力された場合、jstosj および euctosj は変換を正しく行いません。euctosj、sjtoeuc、jstosj、sjtojis については、JIS X 0208 1 区 - 84 区だけが変換の対象になります。

名前	jistoeuc, jistosj, euctojis, euctosj, sjtojis, sjtoeuc – JIS、PC 漢字、日本語 EUC 間のコード変換
形式	jistoeuc [-8] [-U] [<i>filename...</i>] jistosj [-8] [-U] [<i>filename...</i>] euctojis [-8] [-U] [<i>filename...</i>] euctosj [-U] [<i>filename...</i>] sjtojis [-8] [-U] [<i>filename...</i>] sjtoeuc [-U] [<i>filename...</i>]
使用条件	SUNWjfpu
機能説明	<p>本コマンド群は、異なるコード規格の文字セットを相互に変換します。コマンド名 [<i>filename...</i>] は、指定されたファイルの内容に対して指定された変換を行い、標準出力に書き出します。</p> <p>ファイルを指定しないと、標準入力から文字を読み取り、変換します。</p> <p>jistoeuc JIS から日本語 EUC への変換 jistosj JIS から PC 漢字への変換 euctojis 日本語 EUC から JIS への変換 euctosj 日本語 EUC から PC 漢字への変換 sjtojis PC 漢字から JIS への変換 sjtoeuc PC 漢字から日本語 EUC への変換</p>
オプション	-8 jistoeuc、jistosj、sjtojis、sjtoeuc は、SI/SO を使わず、8 単位系 JIS で JIS X 0201 文字集合を処理、あるいはこれに変換します。 -U 出力はバッファリングされません (デフォルトではバッファリングされます)。
関連項目	iconv(1), iconv_ja(5)
注意事項	jistoeuc では、文字集合指示エスケープ・シーケンスとして以下を処理することができます。 <p>JIS X 0208 指示 – \E\$B、\E\$(B、\E\$@ JIS X 0212 指示 – \E\$(D JIS X 0201 Roman 指示 – \E(J、\E(H ASCII 指示 – \E(B</p> <p>sjtojis、euctojis は文字集合指示エスケープ・シーケンスとして以下を出力します。</p>

jistosj(1)

JIS X 0208 指示 - \E\$B
JIS X 0212 指示 - \E\$(D (sjtojis では表示されません)
JIS X 0201 Roman 指示 - \E(J

本コマンドは、原則として入力ファイル中の各コードが正しいかどうかの確認は行いません。また PC 漢字との相互変換においては「TOG 日本ベンダ協議会推奨日本語 EUC・シフト JIS 間コード変換仕様」に基づくコード変換は行われません。iconv(1) ユーティリティがこれらの機能を提供します。iconv(1) および iconv_ja(5) を参照してください。

使用上の留意点

JIS X 0212 文字集合が入力された場合、jistosj および euctosj は変換を正しく行いません。euctosj、sjtoeuc、jistosj、sjtojis については、JIS X 0208 1 区 - 84 区だけが変換の対象になります。

名前	jpostprint - 日本語 PostScript プリンタに日本語テキストを出力するためのフィルタ
形式	<pre> jpostprint [-c <i>number</i>] [-f <i>name</i>] [-l <i>number</i>] [-m <i>number</i>] [-n <i>number</i>] [-o <i>list</i>] [-p <i>mode</i>] [-r <i>number</i>] [-s <i>number</i>] [-t <i>number</i>] [-x <i>number</i>] [-y <i>number</i>] [-u <i>path</i>] [<i>filename...</i>] </pre> <p>/usr/lib/lp/postscript/jpostprint</p>
使用条件	SUNWjfpu
機能説明	<p>jpostprint は、標準入力から日本語文字を読み込み、日本語 PostScript に変換して標準出力に書き出します。入力ファイルを指定しない場合、あるいは - が入力ファイルとして指定された場合は、標準入力から読み取ります。入力文字コードに UDC (ユーザー定義文字)、JIS X 0212、IBM 拡張文字、あるいは NEC 選定 IBM 拡張文字が含まれている場合、これらのフォントも出力します。その他の文字の場合はプリンタ側のフォントを使用します。UDC は sdtudctool(1) により作成します。JIS X 0212、IBM 拡張文字、あるいは NEC 選定 IBM 拡張文字を出力するには、SUNWjcs3f パッケージが別に必要です。</p>
オプション	<p>-c <i>number</i> 各ページのコピー枚数を指定します。デフォルトは 1 枚です。</p> <p>-f <i>name</i> 印刷に使用するフォントを指定します。以下を指定することにより、JIS X 0208, JIS X 0201 カタカナ文字集合のフォントが変更できます。</p> <p>Ryumin-Light (Ryumin-Light-H と同じ) Ryumin-Light-H GothicBBB-Medium (GothicBBB-Medium-H と同じ) GothicBBB-Medium-H Ryumin-Light-V GothicBBB-Medium-V</p> <p>ASCII 部分のフォントは、次のフォントも使用できます。</p> <p>JIS X 0201 ロー マ文字集 合 Ryumin-Light.Hankaku GothicBBB-Medium.Hankaku</p> <p>英文フォント Courier LucidaSans-Typewriter</p> <p>同時に指定する場合は、次のように + を間に入れます。</p> <p>GothicBBB-Medium.Hankaku+GothicBBB-Medium</p> <p>デフォルトは、ASCII 部分は Courier、JIS X 0208 と JIS X 0201 のカタカナ部分は Ryumin-Light-H です。なお、JIS X 0212、VDC (ベンダー定義文字)、および UDC のフォントは変更できません。</p>

jpostprint(1)

- `-l`
number 1 ページの行数を指定します。デフォルトは 66 行です。0 を設定すると、jpostprint は使用しているフォントのポイントサイズに従って適当なページ数を設定します。
- `-m`
number *number* を指定して、論理ページを拡大します。ページは、ページの左上を原点として均一に拡大縮小されます。デフォルトは、1.0 です。
- `-n`
number 1 ページあたり *number* 個の論理ページで印刷します。*number* は自然数です。デフォルトは 1 です。
- `-o list` *list* で指定した範囲のページを印刷します。*list* はページ番号 *N* と範囲 *N1-N2* で構成されます。*N1* が指定されないときは最初のページから、*N2* が指定されないときは最後のページまでと認識されます。ページ範囲は、物理ページではなく論理ページ単位で指定します。たとえば、物理ページ 1 ページ当たりの論理ページが 2 のとき、*list* として 4 ページまでを指定すると (-4)、実際には 2 ページの紙が出力されます。
- `-p mode` 縦長 (ポートレートモード) か横長 (ランドスケープモード) かを指定します。*mode* の最初の文字が *p* の場合には縦長、1 の場合には横長になります。デフォルトは縦長です。これは物理ページでなく、論理ページに対して指定します。たとえば、論理ページ 2 のときの縦長出力は、物理ページでは横長出力になります。
- `-r`
number '\r' の動きを制御します。*number* が 0 なら無視、1 ならカラムを行頭に返し、2 なら改行とみなします。デフォルトは 0 です。
- `-s`
number *number* のポイントサイズでファイルを出力します。横長の場合は、*number* がデバイスの表示領域によって調整されます。縦長出力時のデフォルトサイズは 10 です。ポイントサイズを大きくすると、表示されるイメージの大きさが拡大します。したがって、ポイントサイズを大きくする場合には、より大きな紙を使うか、-10 オプションで 1 ページあたりの行数を調整する必要があります。
- `-t`
number タブが最初のカラムから *number* カラムごとにセットされているとみなします。デフォルトは 8 カラムごとです。
- `-x`
number x 軸に沿って原点を *number* インチ移動します。デフォルトの座標系の原点はページの左上付近から、右に x の正方向上に y の正方向をとります。デフォルト値は 0.25 インチです。
- `-y`
number y 軸に沿って原点を *number* インチ移動します。デフォルトは -0.25 インチです。
- `-u path` UDC フォントのある *path* を指定します。デフォルトは
\$HOME/.Xlocale/locale/fonts/UDC/Type1/UDCxx.pfa、jpostprint.conf
ファイルの fontpath udc の内容の順に調べます。

新しい論理ページは 66 行目を出力したあとか、\f 文字を読み込んだあとから始まります。1 ページの行数は -1 オプションを使って変更できます。非出力文字は無視されます。

使用例 `file1` と `file2` を横長で出力するには、次のコマンドを実行します。

```
example% jpostprint -pland file1 file2
```

論理ページ2で印刷するには、次のコマンドを実行します。

```
example% jpostprint -n2 file
```

UDC フォントとして /usr/local/UDC ディレクトリにある UDC1.pfa から UDC20.pfa のいずれかを使用するときには、次のコマンドを実行します。

```
example% jpostprint -u "/usr/local/UDC/UDC%d.pfa"
```

環境 このコマンドを実行する場合には、ロケールを ja、ja_JP.PCK、または ja_JP.UTF-8 に設定する必要があります。

ファイル /usr/lib/lp/postscript/forms.ps
/usr/lib/lp/postscript/jpostprint.ps
/usr/lib/lp/postscript/jpostprint.conf

関連項目 jtops(1), postprint(1), sdtudctool(1), lpfilter(1M)

注意事項 -n オプションで出力した PostScript ファイルは、印刷時のみ正しく表示します。UDC フォントは sdtudctool が出力する Type1 フォントのみ、JIS X 0212 フォントは SUNWjcs3f に含まれる Type1 フォントのみ対応します。

なお、UTF-8 の場合、UDC に関しては 6400 文字まで対応し、その他は日本語 EUC で表現できる文字のみ対応します。

jprconv(1)

名前	jprconv - ドットマトリックス漢字プリンタ、日本語ページプリンタで日本語テキストを印刷するためのフィルタ								
形式	jprconv [-T <i>terminfo</i>] [-r] /usr/lib/lp/text/jprconv								
使用条件	SUNWjfpv								
機能説明	jprconv は、ドットマトリックス漢字プリンタ (EPSON VP-5085、NEC PC-PR201)、または日本語ページプリンタ (Canon LASERSHOT) を使用して日本語テキストを印刷するためのフィルタです。各プリンタのコントロールコードは次の通りです。								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">データベース</th> <th style="text-align: left;">コントロールコード</th> </tr> </thead> <tbody> <tr> <td>EPSON VP-5085</td> <td>EPSON ESC/P の ESC/P24-J84</td> </tr> <tr> <td>NEC PR201</td> <td>NEC 201PL</td> </tr> <tr> <td>Canon LASERSHOT</td> <td>LIPS に準拠したコントロールコード</td> </tr> </tbody> </table>	データベース	コントロールコード	EPSON VP-5085	EPSON ESC/P の ESC/P24-J84	NEC PR201	NEC 201PL	Canon LASERSHOT	LIPS に準拠したコントロールコード
データベース	コントロールコード								
EPSON VP-5085	EPSON ESC/P の ESC/P24-J84								
NEC PR201	NEC 201PL								
Canon LASERSHOT	LIPS に準拠したコントロールコード								
	<p>上記のコントロールコードをサポートしていれば、他のプリンタでも日本語テキストを印刷できます。</p> <p>jprconv は、標準入力から日本語文字を読み込み、各コントロールコードに変換して標準出力に書き出します。ユーザー定義文字、JIS X 0212、IBM 拡張文字、あるいは NEC 選定 IBM 拡張文字が入力文字コードに含まれている場合、これらのフォントも出力します。その他の文字の場合はプリンタ側のフォントを使用します。</p> <p>ユーザー定義文字は <code>sdtudctool</code> を使用して作成します (<code>sdtudctool(1)</code> 参照)。また、JIS X 0212、IBM 拡張文字、あるいは NEC 選定 IBM 拡張文字を出力するには、さらに <code>SUNWjcs3f</code> パッケージが必要です。</p>								
オプション	<p>-T <i>terminfo</i> データベースを指定します。次のいずれかを必ず指定する必要があります。</p> <p style="padding-left: 40px;"><code>canon-ls-408</code> LIPS 形式の場合</p> <p style="padding-left: 40px;"><code>nec-pr201</code> NEC 201PL 形式の場合</p> <p style="padding-left: 40px;"><code>epson-vp5085</code> ESC/P24-J84 形式の場合</p> <p>-r 出力時に NL を CR-NL に変換しません。デフォルトは変換します。</p>								
使用例	<p><code>file1</code> を ESC/P24-J84 形式で出力するには、次のようにします。</p> <pre>example% jprconv -T epson-vp5085 < file1</pre>								
環境	このコマンドを実行する場合には、ロケールを <code>ja</code> 、 <code>ja_JP.PCK</code> 、または <code>ja_JP.UTF-8</code> に設定する必要があります。								

ファイル	<code>/usr/lib/lp/text/jprconv.conf</code> <code>/usr/share/lib/terminfo/e/epson-vp5085</code> <code>/usr/share/lib/terminfo/n/nec-pr201</code> <code>/usr/share/lib/terminfo/c/canon-ls-a408</code>
関連項目	<code>jtops(1)</code> , <code>jpostprint(1)</code> , <code>sdtudctool(1)</code> , <code>lpfilter(1M)</code> 『日本語環境ユーザズガイド』
注意事項	通常、 <code>jprconv</code> はプリンタサーバー側でフィルタとして使用されるため、 <code>jprconv</code> コマンドを直接使用する必要はありません。プリンタサーバー側の設定方法は『日本語環境ユーザズガイド』を参照してください。 <code>ja_JP.UTF-8</code> ロケールでは日本語文字のみ印刷可能です。

jtops(1)

名前	jtops - Sun LaserWriter または日本語 PostScript プリンタに日本語ファイルを出力するためのフィルタ
形式	jtops [-l2rRjJvwWmg] [-l <i>n</i>] [-s <i>size</i>] [-f <i>font</i>] [<i>filename...</i>]
使用条件	SUNWjfpu
機能説明	<p>jtops は、標準入力から日本語文字を読み込み、Sun Laser Writer に出力するためにプリンタ側の漢字フォントを使う日本語 PostScript に変換して、標準出力に書き出します。</p> <p>入力ファイルを指定しない場合は、標準入力ファイルから読み取ります。</p> <p>特に指定のない場合、フォントサイズ 10、1 ページあたり 66 行で縦型に出力します。</p> <p>コマンド行で指定されているオプションを調べる前に、環境変数 JTOPS に設定されている文字列をオプションとして解釈し、設定します。</p>
オプション	<p>-1 1 カラム出力 (デフォルト値)</p> <p>-2 2 カラム出力</p> <p>-r 横型出力 (rotate)</p> <p>-R 縦型出力 (don't rotate) (デフォルト値)</p> <p>-v printer 側の漢字フォントを使います (日本語 PostScript プリンタ対応、デフォルト値)。</p> <p>-l <i>n</i> 1 ページあたりの行数を指定します。デフォルト値は、66 行です。</p> <p>-m 日本語フォントとして明朝体を使います (デフォルト値)。</p> <p>-g 日本語フォントとしてゴシック体を使います。英字フォントとしては、-f オプションで指定しない限り、Courier-Bold を使います。</p> <p>-f <i>font</i> 英字フォントを指定します。デフォルトには、-g オプションが指定されていない場合には Courier が、指定されている場合には Courier-Bold が使用されます。</p> <p>-s <i>size</i> フォントサイズを <i>size</i> に指定します。フォントサイズを指定した場合、1 ページあたりに出力される行数は、次のように設定されます。</p> <p style="padding-left: 40px;">縦型: 720 / (fontsize + 1) 横型: 550 / (fontsize + 1)</p> <p>-j 英数字フォントとして日本語フォントの半角英数部分を使います。</p> <p>-J 英数字フォントとして Courier を使います (デフォルト値)。</p>

	<code>-W</code>	英数フォントが可変幅の場合を除いて、英数字と日本語の割合を1:2にします(デフォルト値)。
	<code>-w</code>	英数字と日本語の割合を調整しません。
使用例		<code>example% jtops -j -140 -s11 filename lpr</code> <code>example% pr -1120 filename jtops -1120 -s5 lpr</code> <code>example% jtops -2r filename lpr</code>
環境		このコマンドを実行する場合には、ロケールを <code>ja</code> (または <code>japanese</code>)、 <code>ja_JP.PCK</code> 、あるいは <code>ja_JP.UTF-8</code> に設定する必要があります。
関連項目		<code>expand(1)</code> , <code>lp(1)</code> , <code>pr(1)</code> , <code>lpr(1B)</code> , <code>lpfilter(1M)</code>
注意事項		<code>jtops</code> は以下の文字セットのみ対応します。 <ul style="list-style-type: none">■ JIS X 0201 ローマ文字用図形キャラクタ集合■ JIS X 0201 カタカナ用図形キャラクタ集合■ JIS X 0208

jtty(1)

名前	jtty - TTY/STREAMS に対する日本語処理環境の設定または変更	
形式	jtty [-c <i>y</i> <i>n</i>] [-i <i>c</i>] [-o <i>c</i>]	
使用条件	SUNWjfpu	
機能説明	jtty は現在使用している TTY/STREAMS の日本語処理環境の値を変更します。引数をつけなければ、現在使用している TTY/STREAMS の日本語処理環境の値を返します。 現在、日本語処理環境として定義されているのは JIS コードの漢字インおよび漢字アウト・シーケンスの 3 文字目と、コード変換を行うかどうかの選択だけです。	
オプション	-c <i>y n</i>	<i>y</i> を指定すると、現在プッシュされているコード変換モジュールに対してコード変換を行うよう指示します。 <i>n</i> を指定すると、コード変換を行わないよう指示します。
	-i <i>c</i>	コード変換の際に用いる JIS の漢字イン・シーケンスの 3 文字目を <i>c</i> に変更します。
	-o <i>c</i>	コード変換の際に用いる JIS の漢字アウト・シーケンスの 3 文字目を <i>c</i> に変更します。
関連項目	stty(1), jaio(7I)	

名前	kanji – 漢字コードの一覧表を表示
形式	kanji [-j] [-s] [-e] [-k] [-K <i>n</i>] [-HK <i>n</i>] [-h]
使用条件	SUNWjfpu
機能説明	kanji は、現在のロケールにおいて印字可能な文字を表形式で表示します。 オプションなしで起動した場合は、現在のロケールで使われているエンコーディング番号と共に表形式で出力します。
オプション	<p>-j JIS コード番号を付けてすべての文字を出力します。</p> <p>-s PC 漢字コード番号を付けてすべての文字を出力します。このオプションは ja 相当のロケールでのみ有効です。</p> <p>-e 日本語 EUC を付けてすべての文字を出力します。このオプションは ja_JP.PCK 相当のロケールでのみ有効です。</p> <p>-k JIS 区点番号を付けてすべての文字を出力します。</p> <p>-K <i>n</i> ja 相当のロケールでは、JIS X 0208 文字セットの <i>n</i> 区の文字だけを一覧にして表示します。ja_JP.PCK 相当のロケールでは、印字可能文字セットの <i>n</i> 区の文字だけを一覧にして表示します。</p> <p>-HK <i>n</i> JIS X 0212 文字セットの <i>n</i> 区の文字だけを一覧にして表示します。このオプションは ja 相当のロケールでのみ有効です。</p> <p>-h ヘルプメッセージを出力します。</p>
関連項目	PCK(5), eucJP(5)

kkcvtocs00(1)

名前	kkcvtocs00 - kkcw ユーザー辞書から cs00 ユーザー辞書への変換
形式	kkcvtocs00 [<i>filename...</i>]
使用条件	SUNWjfpv
機能説明	kkcvtocs00 は、JLE のかな漢字変換システム kkcw のユーザー辞書単語を JFP のかな漢字変換システムの 1 つである cs00 でも利用しようとする場合に使用するユーティリティです。kkcvtocs00 はフィルタとして使用されます。 <i>filename</i> で指定されたファイルが入力対象となります。 <i>filename</i> を省略すると、標準入力が入力対象となります。変換結果は標準出力に出力されます。入力対象のファイルは kkcvdicedit が扱うことのできる「自立語ファイル」の形式でなければなりません。「自立語ファイル」の形式は行に対して定められており、kkcvtocs00 はこの形式に従わない行を発見するとエラーメッセージを画面に出力してその行を無視します。
使用例	<p>例 1</p> <p>kkcw ユーザー辞書から cs00 ユーザー辞書を作成するには次の手順で行います。</p> <ol style="list-style-type: none"> 1. JLE システム上であらかじめ kkcw のユーザー辞書から自立語ファイルを作成しておきます。 <pre>JLE% kkcvdicedit extract kkcw_u.dic kkcw_u.list</pre> 2. 次に、この自立語ファイルを JFP システム上で udicm 用の単語リストファイルに変換します。 <pre>JFP% kkcvtocs00 kkcw_u.list > cs00_u.list</pre> 3. 最後に、この単語リストファイルから cs00 ユーザー辞書を作成します。 <pre>JFP% udicm create /usr/lib/mle/ja/cs00/cs00_m.dic cs00_u.dic cs00_u.list</pre>
関連項目	mdicm(1), udicm(1)
診断	<p>kkcvtocs00 は kkcw から cs00 へのユーザー辞書の継承をその唯一の目的としており、kkcvdicedit により得られた自立語ファイルが入力ファイルとして与えられることを想定しています。そのため、エラー処理は必ずしも厳密ではありません。</p> <p>現在、次のエラーと警告のメッセージが定義されています。</p> <pre>Error: file 入力ファイル filename の number 行目の形式が正しくありません。 filename line この行は無視されます。 number: is not kkcvdicedit jiritsugo format, ignored.</pre>

```
Warning: file 入力ファイル filename の number 行目に無効な品詞が見つかりまし  
filename line た。それらの品詞は ":UN" に変換されます。  
number:  
"Invalid  
hinshi(s)  
have found.  
They are  
converted  
into ":UN".
```

sdtudctool(1)

名前 sdtudctool – Solaris 外字ツール

形式 /usr/dt/bin/sdtudctool [-f file]

機能説明 Solaris 2.6 から、既存のフォントファイルを編集せずに、ユーザー定義文字フォントファイルを別フォントファイルとして扱えるようになりました。また、日本語環境では、ユーザー定義文字領域が以下のように明確に定義されました。

ユーザー定義文字領域

ja/japanese	0xf5a1 - 0xfefe	JIS X 0208-1990 85 区 - 94 区
	0x8ff5a1- 0x8ffefe	JIS X 0212-1990 85 区 - 94 区

ja_JP.PCK	0xf040 - 0xf4fc	JIS X 0208-1990 85 区 - 94 区
	0xf540 - 0xf9fc	JIS X 0212-1990 85 区 - 94 区

ja_JP.UTF-8	0xee8080 - 0xefa3bf	Unicode U+E000 - U+F8FF
-------------	---------------------	-------------------------

sdtudctool は Solaris 共通デスクトップ環境 (以降、Solaris CDE とします) 上で、ユーザー定義文字を別フォントファイルとして上記の領域に登録するツールです。sdtudctool を使用すると、画面上で使われるすべてのサイズのビットマップフォントにユーザー定義文字を一括して登録でき、同時にアウトラインフォントにもユーザー定義文字を登録することができます。Solaris CDE 上で作成したフォントは Solaris 2.6 またはそれ以降が動作している OpenWindows 上でも利用できます。

sdtudctool は、ユーザー定義文字を保存するディレクトリが環境変数 DTUDCFONTPATH に指定されている場合、そのディレクトリの指定ファイルを編集対象とします。指定されていない場合は、以下のデフォルトディレクトリにあるフォントファイルを編集対象とします (環境変数 DTUDCFONTPATH が指定されていない、またはディレクトリも存在しない場合は自動的に作成されます)。

Solaris 2.6 以降のユーザー定義文字に関するフォントファイルディレクトリ

一般ユーザーが登録する場合	\$HOME/.Xlocale/<locale>/fonts/UDC/ Bitmaps Type1 CID
スーパーユーザーが登録する場合	\$OPENWINHOME/lib/locale/<locale>/X11/fonts/UDC/ Bitmaps

Type1

CID

なお、フォントエディタ (fontedit) またはフォントマネージャ (fontmanager) で登録したユーザー定義文字を再利用する場合は、各フォントファイルに対して移行作業が必要になります。詳細は、後述の「注意事項」、sdtudc_extract(1)、sdtudc_convert(1) のマニュアルページおよび、「日本語環境ユーザーズガイド」の「フォントの再利用」を参照してください。

sdtudctool で作成したアウトラインのユーザー定義文字は jpostprint(1) で印刷できます。詳細は jpostprint(1) のマニュアルページを参照してください。

画面構成 sdtudctool は基本的に次の 3 つのウィンドウで構成されています。

- エディタ
- 一覧表
- 参照表

その他に、オプションダイアログ、一覧表印刷ダイアログなどがあります。

エディタ sdtudctool を起動すると最初に表示されるウィンドウです。ユーザー定義文字が利用できる場合は、起動時に自動的に読み込んで表示します。

画面の構成は以下のようになります。

編集画面 ユーザー定義文字を編集する画面。

参照画面 ユーザー定義文字を作成するときに参照する文字を表示する画面。「表示」メニューから「参照」を選択するか、またはツールバー上の右にある「参照」ボタンを押すと、編集画面の右に表示されます。

確認画面 左下に表示される 2 つの画面。左にアウトラインでの文字を表示し、右にビットマップでの文字を表示します。

描画メニュー 描画の基本的な操作は、編集画面の左にある描画メニューから希望するメニューを選択したあと、編集画面上でマウスのセレクトボタンを押しながらドラッグし、離すことで行います。

ただし、折れ線と多角形の描画は、操作方法が異なります。編集画面上で、線分の開始地点と終了地点ごとにマウスのセレクトボタンをクリックすると、クリックした場所から新たに線分が描画されます。線分の最終地点でマウスのセレクトボタンをダブルクリックして描画を確定します。

次のメニューが提供されています。

自由線	自由線を描画します。
直線	直線を描画します。
折れ線	折れ線を描画します。
多角形	多角形を描画します。

四角形	四角形を描画します。
円	円を描画します。
消しゴム	カーソルで指した箇所を消します。ただし、アウトラインモードの時は選択できないため、削除は編集メニューから選択して行なってください。
範囲指定	編集する範囲の指定をします。編集メニューのコマンドを使う場合は、最初に「範囲指定」で範囲を指定する必要があります。指定された範囲内のオブジェクトはコントロールポイントが小さな四角形で表示されます。 オブジェクトの移動は、範囲を指定したあと、オブジェクトの中心付近でマウスのセレクトボタンを押しながらドラッグして行います。

その他の編集については後述の「編集」の項目を参照してください。

保存ボタン 編集画面で作成した文字を登録する場合に選択します。登録した文字は、一覧表上に表示されます。

次ボタン 編集対象の文字を一覧表上の次の文字にします。

前ボタン 編集対象の文字を一覧表上の前の文字にします。

ツールバーには、以下が表示されます。

アウトライン 編集画面の表示・編集モードをアウトラインモードにします。

ビットマップ 編集画面の表示・編集モードをビットマップモードにします。

参照 参照画面を編集画面の右に表示、または表示を削除します。

一覧表 編集しているフォントファイルに含まれる文字の一覧表を表示します。

メニューバーには、以下が表示されます。

ファイル

ユーザー定義文字を開く Solaris 2.6 以降のユーザー定義文字に関連するフォントファイルを読み込んで表示します。環境変数 DTUDCFONTPATH が無効な場合は以下のディレクトリのフォントファイルを開きます。

一般ユーザーが登録する場合 `$HOME/.Xlocale/<locale>/fonts/UDC/`

Bitmaps

Type1

CID

スーパーユーザーが登録する場合 `$OPENWINHOME/lib/locale/<locale>/X11/fonts/UDC/`

Bitmaps

Type1

CID

開く... フォントファイルを指定して読み込む場合に使用します。この場合、作成した文字は別フォントファイルとして保存されず、直接読み込んだフォントファイルに書き込まれます。

読み込めるフォントの形式は、以下の通りです。

BDF 形式

PCF 形式

Type1 形式 (ただし、この形式については編集できません)

Type3 形式 (ただし、この形式については編集できません)

保存 「ユーザー定義文字を開く...」で Solaris 2.6 以降のユーザー定義文字に関連するフォントファイルを読み込んだ場合には、ユーザー定義文字フォントファイルとして保存し、ユーザー定義文字を利用するために必要な設定を行います。

必要な設定はフォントパスの設定およびフォントパスの保存ですが、設定を省略することもできます。「オプション」の項目を参照してください。

フォントファイルを直接指定して読み込んだ場合には、読み込んだフォントファイルに直接反映され、フォントパスの設定およびフォントパスの保存は行いません。

別名保存.. 編集した結果を別名のフォントファイルとして保存する場合に利用します。「ユーザー定義文字を開く...」で Solaris 2.6 以降のユーザー定義文字に関連するフォントファイルを読み込んだ場合は選択できません。

保存できるファイルの形式は、以下の通りです。

BDF 形式

PCF 形式

辞書登録ツール... ユーザー定義文字辞書登録用の中間ユーティリティ、sdtudc_register を起動します。sdtudc_register の画面には登録したユーザー定義文字が表示され、必要な情報を入力してユーザー定義文字を辞書に登録できます。詳細は sdtudc_register(1) のマニュアルページを参照してください。

オプション 次のオプションが指定できます。

グリッドのサイズ	アウトライン編集モード時のグリッドのサイズを指定します。単位はポイントです。
グリッドに合わせる	アウトライン編集モード時に描画位置をグリッド上に合わせる指定をします。
フォントパスを設定する	ユーザー定義文字の保存時に、現行セッションのフォントパスの最後にユーザー定義文字を保存したディレクトリを追加します。詳細は、xset(1)のマニュアルページを参照してください。
フォントパスの保存	ユーザー定義文字の保存時に保存するフォントパスを次のファイルに設定します。 一般ユーザーの場合 \$HOME/.OWfontpath スーパーユーザーの場合 \$OPENWINHOME/lib/locale/<locale>/OWfontpath ビットマップファイルの生成サイズ ユーザー定義文字を保存する場合、自動生成されるビットマップフォントのサイズを指定します。 以下のサイズを生成できません。

12、14、16、20、24

終了 sdtudctool を終了します。

編集

編集作業に入る前に描画ボタンの範囲指定を選択し、変更するオブジェクトを選択してください。オブジェクトを選択したあと、オブジェクトの中心付近でマウスのセレクトボタンを押してドラッグするとオブジェクトを移動できます。

元に戻す 直前の状態に戻ります。

カット 「範囲指定」で選択した領域を切り取り、バッファに取り込みます。

コピー 「範囲指定」で選択した領域をバッファに取り込みます。このあと、ペーストを行うとバッファの内容を編集画面に描くことができます。

ペースト バッファの内容を編集画面に描きます。このあと、ペーストを行うとバッファの内容を編集画面に描くことができます。

削除 「範囲指定」で選択した領域を削除します。削除された領域はバッファに取り込まれないため、ペーストできません。

回転	編集画面上で「範囲指定」によって選択した領域をマウスのセレクトボタンを押してドラッグすると回転します。
斜め	編集画面上で「範囲指定」によって選択した領域をマウスのセレクトボタンを押してドラッグすると斜めになります。
反転	<p>アウトラインモード時 選択されているフォントのパスの方向を逆にします。sdtudctool では、アウトラインモード時の描画は、非ゼロ巻き数規則 (non-zero winding number rule) で描かれます。パス方向では次のようになります。</p> <p>指定順にパスが描かれるもの 自由線、直線、折れ線、多角形</p> <p>右回りでパスが描かれるもの 四角形、円 たとえば、白抜き円を作成する場合、大きな円の中に小さな円を描き、小さな円を範囲指定で選択して「反転」を選ぶと、小さな円のパスが反転し、非ゼロ巻き数規則に基づいて白抜き円が描画できます。</p> <p>ビットマップモード時 「範囲指定」で選択した領域の白 / 黒を反転します。</p> <p>非ゼロ巻き数規則に関しては『PostScript リファレンス・マニュアル』(Adobe Systems, Inc.)を参照してください。</p>
表示	<p>塗りつぶし アウトライン表示の場合、アウトラインで囲まれた領域を塗り込みます。実際にアウトラインのフォントとして表示されるのはこのイメージです。</p> <p>コントロール アウトラインを描画する場合の基準点を表示します。アウトラインモードでは、1つのパスが持つすべてのコントロールポイントを指定領域に含んだ場合だけ範囲指定は有効になります。</p> <p>ドラッグ表示 領域指定の変更(回転 / 斜めなど)の時に、途中で表示されるイメージの表示方法を指定します。</p> <p>グリッド 編集画面上にグリッドを表示します。アウトラインモードでは、オプションダイアログの「グリッドのサイズ」によりサイズ変更できます。</p> <p>参照 参照画面を表示、または表示を削除します。</p>

	ヘルプ	
	概要	sdtudctool のマニュアルページが表示されます。
	ヘルプの使い方	AnswerBook を立ち上げます。
	Solaris 外字ツールの... について...	Solaris 外字ツールのバージョン番号を表示します。
一覧表	フォントファイルに登録されている文字の一覧表を表示します。sdtudctool に読み込んだフォントファイルにどのような文字が登録されているか知りたい場合に使用します。	
	ツールバーには、以下が表示されます。	
	ページ	スライダーを移動して希望するページに移動します。
	左矢印	前ページに移動します。
	右矢印	次ページに移動します。
	メニューバーには、以下が表示されます。	
	ファイル	
	印刷...	印刷ダイアログを開きます。印刷ダイアログでは、登録したユーザー定義文字の一覧を印刷できます。印刷される文字の大きさは 15、30 ポイントの 2 種類です。また、次の項目を指定できます。
	プリンタ名	出力先のプリンタ名を指定します。
	範囲	
	このページ	現在一覧表に表示されているユーザー定義文字の一覧表を出力します。
	全ページ	現在読み込まれているユーザー定義文字のすべての文字の一覧が表示されます。
	閉じる	一覧表を閉じます。
	編集	
	元に戻す	一つ前の編集状態に戻ります。
	カット	指定されている文字を削除し、バッファに格納します。
	コピー	指定されている文字をバッファに格納します。
	ペースト	バッファの内容を指定された場所に描画します。
	削除	指定されている文字を削除します。
	表示	
	次ページ	次ページがある場合、1 ページ進みます。

参照表

前ページ 前ページがある場合、1 ページ戻ります。

サイズ 表示されているフォントのサイズを変更します。ビットマップフォントを読み込んでいる場合はサイズの変更はできません。

コード 表示されている文字のコード体系を変更します。

ユーザー定義文字を編集する際、参考にした文字があるときに使用します。ユーザー定義文字のエディタのツールバーの左にある「表示」メニューから「参照」ボタンを選択すると、文字参照リストウィンドウが表示されます。

ツールバーには、以下が表示されます。

ページ スライダーを移動して希望するページに移動します。

左矢印 前ページに移動します。

右矢印 次ページに移動します。

メニューバーには、以下が表示されます。

ファイル

開く 参照するフォントを指定するファイル選択リストが開きます。BDF 形式、PCF 形式、Type1 形式、Type3 形式のフォント形式を指定できます。

インストールされているフォントを選択 システムの利用可能なフォントの一覧表を表示します。フォントを選択すると、参照表に表示されます。

閉じる 参照表を閉じます。

表示

次ページ 次ページがある場合、1 ページ進みます。

前ページ 前ページがある場合、1 ページ戻ります。

サイズ 表示されているフォントのサイズを変更します。ただし、ビットマップフォントを読み込んでいる場合はサイズの変更はできません。

コード 表示されている文字のコード体系を変更します。

その他の機能 「Solaris 外字ツール」では、文字のコードポイントの移動および文字のコピーのために、一覧表上でのドラッグ&ドロップ、一覧表から編集画面へのドラッグ&ドロップ、参照表から編集画面へのドラッグ&ドロップ、および参照表から一覧表へのドラッグ&ドロップをサポートしています。ドラッグ&ドロップを行うには、希望する文字をマウスのアジャストボタンで選択し、そのまま希望する場所まで文字をドラッグして離します。

sdtudctool(1)

オプション	<code>-f file</code>	編集するフォントファイルを指定します。
使用法		「Solaris 外字ツール」を起動するには、アプリケーション・マネージャの「デスクトップアプリケーション」を開いて「Solaris 外字ツール」を選択するか、端末エミュレータなどで次のコマンドを入力します。
		<pre>sun% /usr/dt/bin/sdtudctool</pre>
リソース	<code>utRefXFont</code>	参照表を開いた時の表示文字を XLFD 名で指定します。デフォルトは <code>-sun-gothic-medium-r-normal--16-140-75-75-c-140-jisx0208.1983-0</code> です。
	<code>utUDCPrefix</code>	ユーザー定義文字を保存するビットマップフォントとアウトラインフォントファイルの接頭辞を指定します。デフォルトは UDC です。
		この場合、保存するフォントファイル名は次のようになります。
		ビットマップフォントの場合 <code><utUDCPrefix の値><utBDFUDCSize>.pcf</code>
		アウトラインフォントの場合 <code><utUDCPrefix の値><utCIDUDCBase>-<utCIDUDCRange>.ps</code>
		たとえば、日本語のデフォルトでは以下のフォントファイルが保存されます。
		ビットマップフォントファイル <code>UDC{12,14,16,24}.pcf</code>
		Type1 のフォントファイル <code>UDC[1-20].pfa</code>
		JIS エンコーディングのアウトラインフォントファイル <code>NewGothicBBB-Medium-{H,V}.ps</code> <code>NewGothicBBB-Medium-Hojo-{H,V}.ps</code> <code>NewRyumin-Light-{H,V}.ps</code> <code>NewRyumin-Light-Hojo-{H,V}.ps</code>
		EUC エンコーディングのアウトラインフォントファイル <code>NewGothicBBB-Medium-EUC-{H,V}.ps</code> <code>NewGothicBBB-Medium-Hojo-EUC-{H,V}.ps</code> <code>NewRyumin-Light-EUC-{H,V}.ps</code> <code>NewRyumin-Light-Hojo-EUC-{H,V}.ps</code>

SJIS エンコーディングのアウトラインフォントファイル

```
NewGothicBBB-Medium-RKSJ-{H,V}.ps
NewRyumin-Light-RKSJ-{H,V}.ps
```

83pv-RKSJ エンコーディングのアウトラインフォントファイル

```
NewGothicBBB-Medium-83pv-RKSJ-{H,V}.ps
NewRyumin-Light-83pv-RKSJ-{H,V}.ps
```

utUDCBDFSize	ユーザー定義文字のビットマップフォントのサイズを指定します。デフォルトは 12、14、16、20、24 です。
utUDCBDFCR	ユーザー定義文字フォントの Charset Registry を指定します。デフォルトは sunudcja.1997 です。
utUDCBDFCE	ユーザー定義文字フォントの Charset encoding を指定します。デフォルトは 0 です。
utUDCCIDPrefix	ユーザー定義文字を組み込み作成するアウトラインフォント名の接頭辞です。デフォルトは New です。
utUDCCIDBase	ユーザー定義文字を組み込むアウトラインフォント名を指定します。デフォルトは GothicBBB-Medium, Ryumin-Light です。
utUDCBDFAliasTo	ユーザー定義文字を保存するビットマップフォントに対する別名を指定します。デフォルトは -sun-gothic-bold-r-normal-*, \ -sun-gothic-medium-r-normal-*, \ -dt-interface system-medium-r-normal-***-***-m-*, \ -dt-interface user-medium-r-normal-***-***-m-* です。

関連項目 jpostprint(1), sdtudc_convert(1), sdtudc_extract(1), sdtudc_register(1), sdtudc_map(4)

注意事項 Solaris 2.5.1 以前の環境で、fontedit、type3creator、fontmanager を使用してユーザー定義文字を登録している場合は、次の方法で移行作業を行なってください。以下の例は、ja ロケールにおける一般ユーザーのデフォルトのユーザー定義文字環境への移行方法です。

- fontedit で既存のフォントファイルにユーザー定義文字を登録している場合
sdtudc_extract(1) を使用して、登録済みのユーザー定義文字を、別フォントファイルとして抜き出します。この場合、フォントファイル名は UDC<font_size>.bdf とします。次に、生成されたファイルを保存するディレクトリに移動し、sdtudctool を立ち上げて保存します。

sdtudctool(1)

例: gotm14.pcf に UDC を追加して利用している場合

```
% sdtudc_extract gotm14.pcf > UDC14.bdf
% bdf2pcf -o UDC14.pcf UDC14.bdf
% mkdir -p ~/.Xlocale/ja/fonts/UDC/Bitmaps
% mv UDC14.pcf ~/.Xlocale/ja/fonts/UDC/Bitmaps
% /usr/dt/bin/sdtudctool
```

- type3creator および fontmanager で既存のフォントファイルにユーザー定義文字を登録している場合

sdtudc_extract(1) を使用して登録してあるユーザー定義文字を、別フォントファイルとして抜き出します。次に、生成されたファイルをユーザー定義文字を保存するディレクトリに移動し、sdtudctool を立ち上げて保存します。

例: fontmanager で作成した UDC.ps を利用している場合

```
% sdtudc_extract UDC.ps
    UDC1.pfa
    UDC2.pfa
    .
    .
% mkdir -p ~/.Xlocale/ja/fonts/UDC/Type1
% mv UDC*.pfa ~/.Xlocale/ja/fonts/UDC/Type1
% /usr/dt/bin/sdtudctool
```

なお、詳細については「日本語環境ユーザズガイド」を参照してください。

名前	sdtudc_convert - ユーザー定義文字変換ユーティリティ															
形式	<code>/usr/dt/bin/sdtudc_convert [-f <i>map_file</i>] [<i>text_file</i>]</code>															
使用条件	SUNWudct															
機能説明	<p>sdtudc_convert はテキストファイル中のコードセット 1 のコードポイントを マップファイルに指定されたコードポイントに変換して、標準出力へ出力するユーティリティで、Solaris 2.6 以降の日本語 EUC (ja、japanese) ロケール環境でのみサポートされます。</p> <p>Solaris 2.6 からユーザー定義文字は、既存のフォントファイルとは別のフォントファイルに分けて管理するように変更されました。また、日本語環境では、ユーザー定義文字領域が以下のように明確に定義されました。</p> <p>ユーザー定義文字領域</p> <table border="1"> <tr> <td>ja/japanese</td> <td>0xf5a1 - 0xfefe</td> <td>JIS X 0208-1990 85 区 - 94 区</td> </tr> <tr> <td></td> <td>0x8ff5a1- 0x8ffefe</td> <td>JIS X 0212-1990 85 区 - 94 区</td> </tr> </table> <table border="1"> <tr> <td>ja_JP.PCK</td> <td>0xf040 - 0xf4fc</td> <td>JIS X 0208-1990 85 区 - 94 区</td> </tr> <tr> <td></td> <td>0xf540 - 0xf9fc</td> <td>JIS X 0212-1990 85 区 - 94 区</td> </tr> </table> <table border="1"> <tr> <td>ja_JP.UTF-8</td> <td>0xee8080 - 0xefa3bf</td> <td>Unicode U+E000 - U+F8FF</td> </tr> </table> <p>デフォルトでは、以下のようなマップファイルを使い、コードセット 1 の 9 区 - 15 区のコードポイントをコードセット 1 の 85 区 - 91 区に変換します。マップファイルを編集すると、変換する領域や変換先を変更できます。</p> <p>マップファイル:</p> <pre>a9a1,a9ff f5a1 aaa1,aaFF f6a1 aba1,abff f7a1 aca1,acff f8a1 ada1,adff f9a1 aea1,aeff faa1 afa1,afff fba1</pre> <p>詳細については sdtudc_map(4) のマニュアルページを参照してください。</p>	ja/japanese	0xf5a1 - 0xfefe	JIS X 0208-1990 85 区 - 94 区		0x8ff5a1- 0x8ffefe	JIS X 0212-1990 85 区 - 94 区	ja_JP.PCK	0xf040 - 0xf4fc	JIS X 0208-1990 85 区 - 94 区		0xf540 - 0xf9fc	JIS X 0212-1990 85 区 - 94 区	ja_JP.UTF-8	0xee8080 - 0xefa3bf	Unicode U+E000 - U+F8FF
ja/japanese	0xf5a1 - 0xfefe	JIS X 0208-1990 85 区 - 94 区														
	0x8ff5a1- 0x8ffefe	JIS X 0212-1990 85 区 - 94 区														
ja_JP.PCK	0xf040 - 0xf4fc	JIS X 0208-1990 85 区 - 94 区														
	0xf540 - 0xf9fc	JIS X 0212-1990 85 区 - 94 区														
ja_JP.UTF-8	0xee8080 - 0xefa3bf	Unicode U+E000 - U+F8FF														

sdtudc_convert(1)

	<p>マップファイルを指定しない場合、最初に sdtudc_convert を起動したディレクトリの sdtudc_map を検索し、存在しない場合は /usr/dt/config/ja/sdtudc_map を参照します。マップファイルを変更する場合は、/usr/dt/config/ja/sdtudc_map を作業ディレクトリにコピーしてから変更してください。マップファイルの詳細については sdtudc_map(4) のマニュアルページを参照してください。</p> <p>このユーティリティはテキストファイルだけをサポートしています。その他のファイルの変換は、/usr/dt/share/examples/sdtudc/ja/sdtudc_convert.c を参考にしてください。</p>
オプション	<p>次のオプションを指定できます。</p> <p><code>-f map_file</code> 変換に使用するマップファイルを指定します。</p>
終了ステータス	<p>以下の終了ステータスが返されます。</p> <p>0 入力ファイルはすべて、正常に出力された</p> <p>>0 エラーが発生した</p>
ファイル	<p>/usr/dt/config/ja/sdtudc_map</p>
関連項目	<p>sdtudctool(1), sdtudc_extract(1), sdtudc_map(4)</p>
注意事項	<p>Solaris 2.6 から、ユーザー定義文字は既存のフォントファイルとは別のフォントファイルに分けて管理するように変更されました。このため、fontedit を使用して登録したコードセット 1 の 9 区 - 15 区のユーザー定義文字を含むテキストファイルを、Solaris 2.6 以降の環境で表示した場合、ユーザー定義文字が表示されなかったり、別の文字が表示される場合があります。そのような場合は、sdtudc_convert を使ってテキストファイルのコードポイントを変換してください。</p> <p>なお、フォントファイルなどの変換作業の詳細については、sdtudctool(1)、sdtudc_extract(1) のマニュアルページを参照してください。</p>

名前	sdtudc_extract - ユーザー定義文字変換ユーティリティ																
形式	/usr/dt/bin/sdtudc_extract [-f <i>map_file</i>] [-p <i>prefix</i>] <i>font_file</i>																
使用条件	SUNWudct																
機能説明	<p>Solaris 2.6 以降、ユーザー定義文字は既存のフォントファイルとは別のフォントファイルに分けて管理するように変更されました。また、日本語環境では、ユーザー定義文字領域が以下のように明確に定義されました。</p> <p>ユーザー定義文字領域</p> <table border="1"> <tr> <td>ja/japanese</td> <td>0xf5a1 - 0xfefe</td> <td>JIS X 0208-1990 85 区 - 94 区</td> </tr> <tr> <td></td> <td>0x8ff5a1- 0x8ffefe</td> <td>JIS X 0212-1990 85 区 - 94 区</td> </tr> </table> <table border="1"> <tr> <td>ja_JP.PCK</td> <td>0xf040 - 0xf4fc</td> <td>JIS X 0208-1990 85 区 - 94 区</td> </tr> <tr> <td></td> <td>0xf540 - 0xf9fc</td> <td>JIS X 0212-1990 85 区 - 94 区</td> </tr> </table> <table border="1"> <tr> <td>ja_JP.UTF-8</td> <td>0xee8080 - 0xefa3bf</td> <td>Unicode U+E000 - U+F8FF</td> </tr> </table> <p>sdtudc_extract は、フォントファイルからグリフを標準出力へ出力するユーティリティです。また、Solaris 8 から、sdtudc_extract_ps は廃止され、その機能は sdtudc_extract に統合されました。</p> <p>サポートするフォントファイルおよびその出力形態は、以下のようになります。</p> <ul style="list-style-type: none"> ■ ビットマップフォントファイル (.bdf, .bdf.Z, .pcf, .pcf.Z) <p>マップファイルに指定されたコードポイントに対応するグリフを、コードポイントを変換しながら BDF 形式で標準出力へ出力します。</p> <p><i>font_file</i> にビットマップフォントファイルを指定した場合、デフォルトでは、sdtudc_extract は 9 区 - 15 区に登録したユーザー定義文字を 85 区 - 91 区に変換しながら取り出します。マップファイルを変更することによって、取り出す領域や変換先を変更できます。</p> <p>マップファイル：</p> <pre> a9a1,a9ff f5a1 aaa1,aaff f6a1 aba1,abff f7a1 aca1,acff f8a1 ada1,adff f9a1 aea1,aeff faa1 afa1,afff fba1 </pre> 		ja/japanese	0xf5a1 - 0xfefe	JIS X 0208-1990 85 区 - 94 区		0x8ff5a1- 0x8ffefe	JIS X 0212-1990 85 区 - 94 区	ja_JP.PCK	0xf040 - 0xf4fc	JIS X 0208-1990 85 区 - 94 区		0xf540 - 0xf9fc	JIS X 0212-1990 85 区 - 94 区	ja_JP.UTF-8	0xee8080 - 0xefa3bf	Unicode U+E000 - U+F8FF
ja/japanese	0xf5a1 - 0xfefe	JIS X 0208-1990 85 区 - 94 区															
	0x8ff5a1- 0x8ffefe	JIS X 0212-1990 85 区 - 94 区															
ja_JP.PCK	0xf040 - 0xf4fc	JIS X 0208-1990 85 区 - 94 区															
	0xf540 - 0xf9fc	JIS X 0212-1990 85 区 - 94 区															
ja_JP.UTF-8	0xee8080 - 0xefa3bf	Unicode U+E000 - U+F8FF															

sdtudc_extract(1)

マップファイルを指定しない場合、最初に `sdtudc_extract` を起動したディレクトリの `sdtudc_map` を検索し、存在しない場合は `/usr/dt/config/$LANG/sdtudc_map` を参照します。マップファイルを変更する場合は、`/usr/dt/config/$LANG/sdtudc_map` を作業ディレクトリにコピーしてから変更してください。マップファイルの詳細については `sdtudc_map(4)` のマニュアルページを参照してください。

- `type3creator` および `fontmanager` を使用して作成されたユーザー定義文字フォントファイル (.ps) および Windows9X/NT ユーザー定義文字フォントファイル (.ttf)

デフォルトでは、UDC を接頭辞とする Type1 フォント形式フォントファイルを各区ごとに 20 個まで取り出します。これは、Solaris 2.6 で定義するユーザー定義文字領域の区数の総和に対応しています。たとえば、フォントマネージャで作成したフォントファイルを `font.ps` とすると、以下のようになります。

```
% sdtudc_extract font.ps
UDC1.pfa ... Done
UDC2.pfa ... Done
      :
UDC20.pfa ... Done
```

上記の処理を行なったあと、取り出した Type1 フォントファイルを所定のディレクトリに移動します。詳しくは、`sdtudctool(1)` のユーザー定義文字の移行の項目を参照してください。

オプション 次のオプションを指定できます。

- f *map_file* 変換に使用するマップファイルを指定します。
- p *prefix* 抜き出す Type1 フォントファイルの接頭辞を指定します。指定しない場合は、デフォルトとして UDC が使われます。

終了ステータス 以下の終了ステータスが返されます。

- 0 入力ファイルはすべて、正常に出力された
- >0 エラーが発生した

ファイル `/usr/dt/config/$LANG/sdtudc_map`

関連項目 `sdtudctool(1)`, `sdtudc_convert(1)`, `sdtudc_map(4)`

名前	sdtudc_extract_ps - ユーザー定義文字変換ユーティリティ												
形式	<code>/usr/dt/bin/sdtudc_extract_ps [-p <i>prefix</i>] <i>font_file</i></code>												
使用条件	SUNWudct												
機能説明	<p>sdtudc_extract_ps は type3creator および fontmanager を使用して作成されたユーザー定義文字を、Type1 フォント形式のフォントファイルとして抜き出すユーティリティです。</p> <p>Solaris 2.6 から、ユーザー定義文字は既存のフォントファイルとは別のフォントファイルに分けて管理するように変更されました。また、ユーザー定義文字領域が以下のように明確に定義されました。</p> <p>ユーザー定義文字領域</p> <table border="1"> <tr> <td>ja/japanese</td> <td>0xf5a1 - 0xfefe</td> <td>JIS X 0208-1990 85 区 - 94 区</td> </tr> <tr> <td></td> <td>0x8ff5a1 - 0x8ffefe</td> <td>JIS X 0212-1990 85 区 - 94 区</td> </tr> </table> <table border="1"> <tr> <td>ja_JP.PCK</td> <td>0xf040 - 0xf4fc</td> <td>JIS X 0208-1990 85 区 - 94 区</td> </tr> <tr> <td></td> <td>0xf540 - 0xf9fc</td> <td>JIS X 0212-1990 85 区 - 94 区</td> </tr> </table> <p>このため、type3creator および fontmanager を使用して登録したユーザー定義文字を Solaris 2.6 上で利用する場合は、Solaris 2.6 で定義されるユーザー定義文字領域とは別のフォントファイルに取り出す必要があります。</p> <p>デフォルトでは、sdtudc_extract_ps は UDC を接頭辞とするフォントファイルを各区ごとに 20 個まで取り出します。これは、Solaris 2.6 で定義するユーザー定義文字領域の区数の総和に対応しています。たとえば、フォントマネージャで作成したフォントファイルを font.ps とすると以下ようになります。</p> <pre>% sdtudc_extract_ps font.ps UDC1.pfa . . . Done UDC2.pfa . . . Done : UDC20.pfa . . . Done</pre> <p>上記の処理を行なったあと、取り出した Type1 フォントファイルを所定のディレクトリに移動します。詳しくは、sdtudctool(1) のユーザー定義文字の移行の項目を参照してください。</p>	ja/japanese	0xf5a1 - 0xfefe	JIS X 0208-1990 85 区 - 94 区		0x8ff5a1 - 0x8ffefe	JIS X 0212-1990 85 区 - 94 区	ja_JP.PCK	0xf040 - 0xf4fc	JIS X 0208-1990 85 区 - 94 区		0xf540 - 0xf9fc	JIS X 0212-1990 85 区 - 94 区
ja/japanese	0xf5a1 - 0xfefe	JIS X 0208-1990 85 区 - 94 区											
	0x8ff5a1 - 0x8ffefe	JIS X 0212-1990 85 区 - 94 区											
ja_JP.PCK	0xf040 - 0xf4fc	JIS X 0208-1990 85 区 - 94 区											
	0xf540 - 0xf9fc	JIS X 0212-1990 85 区 - 94 区											
オプション	<p>次のオプションを指定できます。</p> <p><code>-p <i>prefix</i></code> 抜き出す Type1 フォントファイルの接頭辞を指定します。指定しない場合 デフォルトとして UDC が使われます。</p>												
終了ステータス	以下の終了ステータスが返されます。												

sdtudc_extract_ps(1)

	0	入力ファイルはすべて、正常に出力された。
	>0	エラーが発生した
ファイル		/usr/dt/config/\$LANG/sdtudc_map
関連項目		sdtudctool(1), sdtudc_convert(1), sdtudc_extract(1), sdtudc_map(4)

名前	sdtudc_register - ユーザー定義文字辞書登録用中間ユーティリティ
形式	<code>/usr/dt/bin/sdtudc_register</code>
使用条件	SUNWudct
機能説明	<p>sdtudc_register は sdtudctool で作成するユーザー定義文字を辞書に登録するために、sdtudctool から起動される辞書登録用の中間ユーティリティです。sdtudc_register 単独では利用できません。必ず sdtudctool から起動してください。</p> <p>sdtudc_register の画面構成は、登録したい辞書の種類を指定するボタン、文字の表示コードを変更するためのメニューからなり、sdtudctool を使用して登録されたユーザー定義文字の一覧が、コードポイントとグリフとして表示されます。同時に複数の辞書を指定できます。</p> <p>現在 sdtudc_register は、日本語環境 (ja, ja_JP.PCK, ja_JP.UTF-8 ロケール) のみでサポートされています。</p> <p>sdtudc_register には、日本語入力システム ATOK12 および Wnn6 の辞書ファイルへ単語登録するために、次の機能が含まれています。</p> <p>atok12wordlist sdtudc_register の辞書ファイル登録用画面で入力した「ユーザー定義文字」、「読み」を atok12wordlist 形式のテキストデータで保存します (atok12wordlist(4) 参照)。品詞は「単漢字」になります。ATOK12 辞書メンテナンスツールを使用して、辞書ファイルに登録できます。</p> <p>Wnn6 テスト形式 sdtudc_register の辞書ファイル登録用画面で入力した「ユーザー定義文字」、「読み」、「コメント」を Wnn6 辞書ファイル形式のテキストデータで保存します。品詞は「単漢字」になります。wnndictutil を起動して、辞書ファイルに登録できます。</p> <p>辞書ファイルへの単語登録についての詳細は、「ATOK12 ユーザーズガイド」、「Wnn6 ユーザーズガイド」、および各コマンドのマニュアルページを参照してください。</p>
ファイル	<p><code>/var/tmp/UDC*</code> 中間ファイル</p> <p><code>\$OPENWINHOME/lib/locale/\$LANG/udcscript/*</code> 各辞書の単語登録ユーティリティ</p>
関連項目	atok12(1), sdtudctool(1), wnndictutil(1)
注意事項	<p>Solaris 2.6 から、ユーザー定義文字を既存のフォントファイルとは別のフォントファイルに分けて管理するようになりました。このため、以前 fontedit を使用して登録したユーザー定義文字を Solaris 2.7 上で利用する場合は、Solaris 2.7 で定義されるユーザー定義文字領域に、別のフォントファイルとして取り出す必要があります。</p> <p>詳しくは sdtudctool(1)、sdtudc_extract(1)、sdtudc_convert(1) のマニュアルページを参照してください。</p>

sjtoeuc(1)

名前	jistoeuc, jistosj, euctojis, euctosj, sjtojis, sjtoeuc – JIS、PC 漢字、日本語 EUC 間のコード変換
形式	jistoeuc [-8] [-U] [<i>filename...</i>] jistosj [-8] [-U] [<i>filename...</i>] euctojis [-8] [-U] [<i>filename...</i>] euctosj [-U] [<i>filename...</i>] sjtojis [-8] [-U] [<i>filename...</i>] sjtoeuc [-U] [<i>filename...</i>]
使用条件	SUNWjfpu
機能説明	<p>本コマンド群は、異なるコード規格の文字セットを相互に変換します。コマンド名 [<i>filename...</i>] は、指定されたファイルの内容に対して指定された変換を行い、標準出力に書き出します。</p> <p>ファイルを指定しないと、標準入力から文字を読み取り、変換します。</p> <p>jistoeuc JIS から日本語 EUC への変換 jistosj JIS から PC 漢字への変換 euctojis 日本語 EUC から JIS への変換 euctosj 日本語 EUC から PC 漢字への変換 sjtojis PC 漢字から JIS への変換 sjtoeuc PC 漢字から日本語 EUC への変換</p>
オプション	-8 jistoeuc、jistosj、sjtojis、sjtoeuc は、SI/SO を使わず、8 単位系 JIS で JIS X 0201 文字集合を処理、あるいはこれに変換します。 -U 出力はバッファリングされません (デフォルトではバッファリングされます)。
関連項目	iconv(1), iconv_ja(5)
注意事項	<p>jistoeuc では、文字集合指示エスケープ・シーケンスとして以下を処理することができます。</p> <p>JIS X 0208 指示 – \E\$B、\E\$ (B、\E\$@ JIS X 0212 指示 – \E\$ (D JIS X 0201 Roman 指示 – \E (J、\E (H ASCII 指示 – \E (B</p> <p>sjtojis、euctojis は文字集合指示エスケープ・シーケンスとして以下を出力します。</p>

JIS X 0208 指示 - \E\$B
JIS X 0212 指示 - \E\$(D(sjtojis では表示されません)
JIS X 0201 Roman 指示 - \E(J

本コマンドは、原則として入力ファイル中の各コードが正しいかどうかの確認は行いません。また PC 漢字との相互変換においては「TOG 日本ベンダ協議会推奨日本語 EUC・シフト JIS 間コード変換仕様」に基づくコード変換は行われません。iconv(1) ユーティリティがこれらの機能を提供します。iconv(1) および iconv_ja(5) を参照してください。

使用上の留意点

JIS X 0212 文字集合が入力された場合、jistosj および euctosj は変換を正しく行いません。euctosj、sjtoeuc、jistosj、sjtojis については、JIS X 0208 1 区 - 84 区だけが変換の対象になります。

sjtojis(1)

名前	jistoeuc, jistosj, euctojis, euctosj, sjtojis, sjtoeuc – JIS、PC 漢字、日本語 EUC 間のコード変換
形式	jistoeuc [-8] [-U] [<i>filename...</i>] jistosj [-8] [-U] [<i>filename...</i>] euctojis [-8] [-U] [<i>filename...</i>] euctosj [-U] [<i>filename...</i>] sjtojis [-8] [-U] [<i>filename...</i>] sjtoeuc [-U] [<i>filename...</i>]
使用条件	SUNWjfpu
機能説明	<p>本コマンド群は、異なるコード規格の文字セットを相互に変換します。コマンド名 [<i>filename...</i>] は、指定されたファイルの内容に対して指定された変換を行い、標準出力に書き出します。</p> <p>ファイルを指定しないと、標準入力から文字を読み取り、変換します。</p> <p>jistoeuc JIS から日本語 EUC への変換 jistosj JIS から PC 漢字への変換 euctojis 日本語 EUC から JIS への変換 euctosj 日本語 EUC から PC 漢字への変換 sjtojis PC 漢字から JIS への変換 sjtoeuc PC 漢字から日本語 EUC への変換</p>
オプション	-8 jistoeuc、jistosj、sjtojis、sjtoeuc は、SI/SO を使わず、8 単位系 JIS で JIS X 0201 文字集合を処理、あるいはこれに変換します。 -U 出力はバッファリングされません (デフォルトではバッファリングされます)。
関連項目	iconv(1), iconv_ja(5)
注意事項	<p>jistoeuc では、文字集合指示エスケープ・シーケンスとして以下を処理することができます。</p> <p>JIS X 0208 指示 – \E\$B、\E\$(B、\E\$@ JIS X 0212 指示 – \E\$(D JIS X 0201 Roman 指示 – \E(J、\E(H ASCII 指示 – \E(B</p> <p>sjtojis、euctojis は文字集合指示エスケープ・シーケンスとして以下を出力します。</p>

JIS X 0208 指示 - \E\$B
JIS X 0212 指示 - \E\$(D (sjtojis では表示されません)
JIS X 0201 Roman 指示 - \E(J

本コマンドは、原則として入力ファイル中の各コードが正しいかどうかの確認は行いません。また PC 漢字との相互変換においては「TOG 日本ベンダ協議会推奨日本語 EUC・シフト JIS 間コード変換仕様」に基づくコード変換は行われません。iconv(1) ユーティリティがこれらの機能を提供します。iconv(1) および iconv_ja(5) を参照してください。

使用上の留意点

JIS X 0212 文字集合が入力された場合、jistosj および euctosj は変換を正しく行いません。euctosj、sjtoeuc、jistosj、sjtojis については、JIS X 0208 1 区 - 84 区だけが変換の対象になります。

uum(1)

名前	uum - かな漢字変換フロントエンドプロセッサ
形式	<code>/usr/bin/uum [-J -U -S -T] [-j -u -s -t] [-h -H] [-x -X] [-k filename] [-c filename] [-r filename] [-D hostname] [-n username] [-l number]</code>
機能説明	<p>uum は、端末上で日本語入出力環境を提供するコマンドです。uum は起動時に、初期化ファイルを次の順に検索し、読み込みます。</p> <ol style="list-style-type: none"> 1. 環境変数 UUMRC に設定されたファイル 2. \$HOME/.uumrc 3. /etc/lib/locale/ja/wnn/ja/uumrc 4. /usr/lib/locale/ja/wnn/ja/uumrc <p>uum は、同じマシンおよびネットワークで接続された マシンで動作するかな漢字変換サーバー jserver と接続することが可能です。サーバーと接続できない場合でも uum は起動でき、かな漢字変換サーバーとの通信を必要としない操作を行えます。また、変換キーが押された時に jserver と接続してない場合は、自動的に接続を確立しようとします。</p>
オプション	<p>次のオプションが指定できます。</p> <p>-H 変換オンの状態で立ち上げます (デフォルト)。</p> <p>-h 変換オフの状態で立ち上げます。</p> <p>-U アプリケーション側 (仮想端末側) を日本語 EUC (UJIS) コードにします (デフォルト)。</p> <p>-J アプリケーション側 (仮想端末側) を JIS コードにします。</p> <p>-S アプリケーション側 (仮想端末側) を PC 漢字 (シフト JIS) コードにします。</p> <p>-T アプリケーション側 (仮想端末側) を UTF-8 コードにします。</p> <p>-u 端末側を日本語 EUC (UJIS) コードにします (デフォルト)。</p> <p>-j 端末側を JIS コードにします。</p> <p>-s 端末側を PC 漢字 (シフト JIS) コードにします。</p> <p>-t 端末側を UTF-8 コードにします。</p> <p>-X 端末のフロー制御をオンにして立ち上げます (デフォルト)。</p> <p>-x 端末のフロー制御をオフにして立ち上げます。</p> <p>-k キー割り当て定義ファイルを指定します。このオプションが省略された場合は、次の順に検索します。</p> <ol style="list-style-type: none"> 1. 初期化ファイル uumrc の setuumkey エントリで指定されたファイル 2. /usr/lib/locale/ja/wnn/ja/uumkey <p>-c キーコード変換表ファイルを指定します。このオプションが省略された場合は、次の順に検索します。</p> <p><i>filename</i></p>

1. 初期化ファイル `uumrc` の `setconvkey` エントリで指定されたファイル
 2. `/usr/lib/locale/ja/wnn/cvt_key_tbl`
- r** *filename* ローマ字かな変換 (`wnn_automaton(4)` を参照) のモード定義表ファイルを指定します。ディレクトリ名の場合、その下の `mode` ファイルが対象になります。このオプションが省略された場合は、次の順に検索します。
1. 初期化ファイル `uumrc` の `setrkfile` エントリで指定されたファイル
 2. `/usr/lib/locale/ja/wnn/ja/rk/mode`
- l** *number* かな漢字変換に使用する行数 ($0 < \text{変換行数} < \text{画面行数} - 1$) を指定します。デフォルトは 1 です。
- D** *hostname* かな漢字変換サーバー (`jserver`) を *hostname* で指定します。このオプションが省略された場合は、次の優先順位で接続するかな漢字変換サーバーを決定します。
1. 環境変数 `JSERVER`
 2. `localhost`
 3. UNIX ドメインソケット
- n** *username* `Wnn6` の環境名として使用するユーザー名を指定します。このオプションが省略された場合は、次の優先順位で使用するユーザー名を決定します。
1. 環境変数 `WNNUSER`
 2. `uum` を実行したユーザーのユーザー名

属性 次の属性については `attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWjwncu

関連項目 `jserver(1M)`, `uumkey(4)`, `uumrc(4)`, `wnn_automaton(4)`, `wnn_cvt_key_tbl(4)`, `wnn_mode(4)`

注意事項 `uum` は、仮想端末を 1 つ獲得します。仮想端末が獲得できない場合は起動しません。また、初期化ファイル、キー割り当て定義ファイル、ローマ字かな変換のモード定義表ファイルが見つからない場合も起動しません。

辞書管理は、`Wnn6` の環境名で行われます。

Wnn6(1)

名前	Wnn6 – Wnn6 日本語入力システム
機能説明	<p>Wnn6 は、デスクトップ環境や端末上で日本語を入力するための手段を提供します。</p> <p>デスクトップ環境で日本語を入力するには、htt(1) または xjsi(1) を使用します。シリアル接続された端末上で日本語を入力するには、uum(1) を使用します。</p> <p>日本語デスクトップに標準の状態ではログインすると、日本語入力システムとして Wnn6 が使用できます。他の日本語入力システムを使用するように設定している場合は、wnn6setup(1) を実行すると Wnn6 を使用するように設定を変更できます。</p> <p>ユーザーは、日本語の入力方法を好みの入力スタイルに設定できます (GUI ユーティリティ、wnnenvutil(1) を参照)。</p> <p>また、ユーザーごとのかな漢字変換辞書に単語を登録し、使用することができます (GUI ユーティリティ、wnndictutil(1) を参照)。</p> <p>Wnn6 はクライアント/サーバーモデルで構成されており、サーバーは同時に複数のクライアント (ユーザー) をサポートします。</p> <p>かな漢字変換に使用する辞書は、サーバー上で集中管理できます。</p>
関連項目	<p>htt(1), xjsi(1), uum(1), wnn6setup(1), wnnatod(1), wnnbushu(1), wnndictutil(1), wnndtoa(1), wnnenvutil(1), wnnnotow(1), wnnstat(1), wnntouch(1),</p> <p>jserver(1M), wnnaccess(1M), wnnnds(1M), wnnkill(1M), wnnoffline(1M), wnnudmerge(1M), dpkeyserv(1M), dpkeystat(1M),</p> <p>wnnenvrc(4), wnnhosts(4), jserverrc(4), uumkey(4), uumrc(4), wnn_2A_CTRL(4), wnn_2B_ROMKANA(4), wnn_automaton(4), wnn_cvt_key_tbl(4), wnn_cvt_xim_tbl(4), wnn_hinsi.data(4), wnn_mode(4), wnn_serverdefs(4), wnn_ximrc(4)</p> <p>『日本語入力方式の概要と移行』</p> <p>『Wnn6 ユーザーズガイド』</p> <p>『Wnn6 上級ユーザーおよびシステム管理者ガイド』</p>

名前	wnn6setup - Wnn6 を X 環境で使用する日本語入力システムにする				
形式	wnn6setup				
機能説明	<p>wnn6setup は、Wnn6 を X ウィンドウシステム環境でログインした時に起動される日本語入力システムとして設定します。wnn6setup で設定した内容は、次にログインした時から有効になります。</p> <p>wnn6setup は、次のファイルを編集します。</p> <pre>\$HOME/.dtprofile 共通デスクトップ環境 (CDE) でログインする時に実行されるスクリプト。Wnn6 用に htt(1) を起動する処理を追加します。</pre> <pre>\$HOME/.Xlocale/ja/app-defaults/Htt htt(1) が参照するリソースファイル。xjsi(1) をインタフェースモジュールと して使用する設定を追加します。</pre> <p><code>\$HOME/.dtprofile</code> がすでに存在し、<code>atok12setup(1)</code> を使用して日本語入力システムを起動するよう設定されている場合、その設定を、Wnn6 を日本語入力システムとして使用する設定で置き換えます。</p> <p><code>\$HOME/.dtprofile</code> に追加される Wnn6 を日本語入力システムとして使用するための設定は、次のように目印となる行 (コメント) で前後を囲まれます。</p> <pre>###== - Generated by wnn6setup to launch japanese XIM. == BEGIN == -==### . . . 日本語入力システムの起動の設定 ###== - Generated by wnn6setup to launch japanese XIM. == END == -==###</pre> <p>これらの目印となる行で囲まれた領域は <code>atok12setup(1)</code> を実行することで書き換えられます。したがって、これらの目印の行の内側を編集した後、<code>atok12setup(1)</code> を実行すると、編集した内容は失われます。また、これらの目印となる行は編集しないでください。</p>				
ファイル	<pre>\$HOME/.dtprofile \$HOME/.Xlocale/ja/app-defaults/Htt</pre>				
属性	次の属性については <code>attributes(5)</code> のマニュアルページを参照してください。				
	<table border="1"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWjwncx</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWjwncx
属性タイプ	属性値				
使用条件	SUNWjwncx				
関連項目	<code>atok12setup(1)</code> , <code>dtlogin(1)</code> , <code>htt(1)</code> , <code>xjsi(1)</code>				
注意事項	ユーザーが <code>\$HOME/.dtprofile</code> を編集して、日本語入力システムを設定した場合、 <code>wnn6setup</code> コマンド実行後に再度ログインし直しても Wnn6 を利用できない場合があります。この場合は、ユーザーが設定した <code>\$HOME/.dtprofile</code> の日本語入力システムの設定を削除してください。				

wnnatod(1)

名前	wnnatod - テキスト形式辞書からバイナリ形式辞書への変換				
形式	<code>/usr/bin/wnnatod [-s num] [-R] [-S] [-U] [-r] [-N] [-n] [-P filename] [-p filename] [-I] [-e] [-h filename] binary_dictionary_filename</code>				
機能説明	wnnatod は、標準入力のテキスト形式辞書(日本語 EUC)をバイナリ形式辞書に変換して、指定された <i>binary_dictionary_filename</i> に出力します。				
オプション	<p>以下のオプションが指定できます。</p> <p><code>-s num</code> 割り当てるメモリーの量を指定します。辞書の語数より少し大きい数字を <i>num</i> に指定しますが、通常は指定する必要はありません。デフォルトの値は 70,000 です。wnnatod がメモリー領域不足のメッセージを表示して終了した場合は、<code>-s</code> を指定して再実行してください。</p> <p><code>-R</code> テキスト形式辞書を逆引き可能なバイナリ形式辞書へ変換します(デフォルト)。</p> <p><code>-S</code> テキスト形式辞書を固定形式辞書に変換します。</p> <p><code>-U</code> テキスト形式辞書を登録可能形式辞書に変換します。</p> <p><code>-r</code> テキスト形式辞書を変換するとき、「読み」と「単語」を入れ替えます。</p> <p><code>-N</code> 辞書のパスワードをアスタリスク(*)に設定します。</p> <p><code>-n</code> 頻度のパスワードをアスタリスク(*)に設定します。</p> <p><code>-P filename</code> 辞書のパスワードのファイル名を設定します。</p> <p><code>-p filename</code> 頻度のパスワードのファイル名を設定します。</p> <p><code>-I</code> システム辞書を作成します。</p> <p><code>-e</code> 「読み」と「単語」が等しい(「単語」がひらがなだけで構成される)場合、「読み」を「単語」としても扱う形式でバイナリ形式辞書に登録します。これにより、バイナリ形式辞書のサイズを小さくします。</p> <p>このオプションを指定した場合、逆引き可能なバイナリ形式辞書へ変換することはできません。</p> <p><code>-h filename</code> 使用する品詞データファイル名を指定します。</p>				
属性	<p><i>filename</i> 次の属性については attributes(5) のマニュアルページを参照してください。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWjwncu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWjwncu
属性タイプ	属性値				
使用条件	SUNWjwncu				
関連項目	wnndictutil(1), wnndtoa(1), wnnnotow(1), wnnntouch(1)				

名前	wnnbushu - Wnn6 部首入力ユーティリティ				
形式	<code>/usr/openwin/bin/wnnbushu -s -D jserver_name [-h] [-b filename] [-f filename]</code>				
機能説明	wnnbushu は、xjsi に部首入力機能を提供するユーティリティです。通常、wnnbushu は xjsi によって起動されます。部首名および総画数で漢字を検索し、候補を表示、選択できます。				
オプション	<p>以下のオプションを指定できます。</p> <p><code>-s</code> 部首入力ユーティリティを xjsi からではなく、単独で起動します。</p> <p><code>-D jserver_name</code> 部首名または総画数から漢字の検索を行う jserver のホスト名を指定します。本オプションを省略することはできません。</p> <p><code>-h</code> ヘルプメッセージを出力します。</p> <p><code>-b filename</code> 部首名または総画数から漢字を検索するための、部首入力用辞書ファイル名を指定します。本オプションを指定しない場合は、<code>/usr/lib/locale/ja/wnn/ja/dic/bushu/bushu.dic</code> を使用します。</p> <p><code>-f filename</code> 部首名または総画数から漢字を検索するための、部首入力用付属語ファイル名を指定します。本オプションを指定しない場合は、<code>/usr/lib/locale/ja/wnn/ja/dic/bushu/bushu.fzk</code> を使用します。</p>				
属性	次の属性については <code>attributes(5)</code> のマニュアルページを参照してください。				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWjwncx</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWjwncx
属性タイプ	属性値				
使用条件	SUNWjwncx				
関連項目	xjsi(1)				

wnndictutil(1)

名前	wnndictutil – 辞書ユーティリティ				
形式	<code>/usr/openwin/bin/wnndictutil [-D <i>jserver_name</i>] [-E <i>user_name</i>] [<i>toolkit options</i>]</code>				
機能説明	<p>wnndictutil は、Wnn6 における辞書操作機能を提供します。主な機能を次に示します。</p> <ul style="list-style-type: none"> ■ 単語登録機能 (1 語単位、一括) ■ 単語削除機能 (1 語単位、一括) ■ 単語情報操作機能 (使用/不使用、コメント、頻度) ■ 辞書併合機能 ■ ユーザー辞書、学習情報の保存、復元機能 				
オプション	<p>以下のオプションが指定できます。</p> <p><code>-D <i>jserver_name</i></code> 接続するかな漢字変換サーバー <i>jserver</i> を起動しているホスト名で指定します。このオプションが省略された場合、次の優先順位で接続するかな漢字変換サーバーを決定します。</p> <ol style="list-style-type: none"> 1. 環境変数 <code>JSERVER</code> 2. リソース <code>Dictutil.serverName</code> 3. <i>jserver</i> (ホスト名) 4. UNIX ドメインソケット <p><code>-E <i>user_name</i></code> Wnn6 の環境名として使用するユーザー名を指定します。このオプションが省略された場合、次の優先順位で Wnn6 の環境名として使用するユーザー名を決定します。</p> <ol style="list-style-type: none"> 1. 環境変数 <code>WNNUSER</code> 2. リソース <code>Wnndictutil.userName</code> 3. <code>wnndictutil</code> を実行したユーザーのユーザー名 <p><i>toolkit options</i> X Toolkit 初期化関数の標準オプションを使用することができます。</p>				
属性	<p>次の属性については <code>attributes(5)</code> のマニュアルページを参照してください。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWjwncx</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWjwncx
属性タイプ	属性値				
使用条件	SUNWjwncx				
関連項目	<code>wnnatod(1)</code> , <code>wnndtoa(1)</code> , <code>wnnotow(1)</code> , <code>xjsi(1)</code>				

名前	wnndtoa – バイナリ形式辞書からテキスト形式辞書への変換				
形式	<code>/usr/bin/wnndtoa [-n] [-s] [-e -E] [-h filename] binary_dictionary_filename [frequency_filename. ...]</code>				
機能説明	<p>wnndtoa は、指定されたバイナリ形式辞書ファイル (<i>binary_dictionary_filename</i>) をテキスト形式辞書 (日本語 EUC) に変換して、標準出力に出力します。</p> <p>頻度ファイル (<i>frequency_filename</i>) を 1 つ以上指定すると、その頻度情報がテキスト形式辞書に反映されます。</p>				
オプション	<p>以下のオプションが指定できます。</p> <ul style="list-style-type: none"> -n 「読み」で整列します (「一」、ひらがな、ASCII 文字 (英数字) の順)。 -s シリアル番号をつけて出力します。 -e 特殊表現に展開します。このオプションをつけると、空白文字、タブなどを 8 進表現に展開します (デフォルト)。 -E 特殊表現に展開しません。このオプションをつけると、空白文字、タブなどを 8 進表現に展開しません。 -h 品詞ファイル名を指定します。デフォルト <i>filename</i> は、<code>/usr/lib/locale/ja/wnn/ja/hinsi.data</code> です。 				
属性	<p>次の属性については <code>attributes(5)</code> のマニュアルページを参照してください。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWjwncu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWjwncu
属性タイプ	属性値				
使用条件	SUNWjwncu				
関連項目	<code>wnnatod(1)</code> , <code>wnndictutil(1)</code> , <code>wnn_hinsi.data(4)</code>				

wnnenvutil(1)

名前 wnnenvutil - 環境設定ユーティリティ

形式 **/usr/openwin/bin/wnnenvutil**

機能説明 wnnenvutil は、Wnn6 における動作環境のカスタマイズ機能を提供します。Wnn6 に対して特別な知識がなくても、簡単に環境を設定できます。

属性 次の属性については `attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWjwncx

関連項目 `uum(1)`, `xjsi(1)`, `jserver(1M)`, `wnnds(1M)`

名前	wnnotow - ユーザー辞書コンバータ				
形式	<code>/usr/bin/wnnotow [-i input_file] [-f format_file] [-o output_file] [-l log_file] [-h frequency_value]</code>				
機能説明	wnnotow は、Wnn6 のテキスト辞書形式に、他の日本語入力システム (ATOK7/ATOK8/cs00/EGBRIDGE/VJE-Delta) で作成された単語のテキスト形式辞書を変換します。				
オプション	<p>以下のオプションが指定できます。</p> <p><code>-i input_file</code> 変換を行うテキスト形式辞書を指定します。文字コードは日本語 EUC でなければなりません。このオプションが省略された場合は、標準入力から入力します。</p> <p><code>-f format_file</code> 入力ファイルに対応した書式を定義している書式ファイル名を指定します。文字コードは日本語 EUC でなければなりません。各日本語入力システムに対応した書式ファイルは、次の通りです。</p> <p>ディレクトリ: <code>/usr/lib/locale/ja/wnn/ja/otow.format/</code></p> <p>ファイル: <code>atok7-wnn6.fmt (ATOK7) atok8-wnn6.fmt (ATOK8) cs00-wnn6.fmt (cs00) egbridge-wnn6.fmt (EGBRIDGE) vje-wnn6.fmt (VJE-Delta)</code> このオプションが省略された場合は、ATOK7 の書式であると見なします。</p> <p><code>-o output_file</code> 指定されたファイルに変換結果を出力します。出力する文字コードは日本語 EUC です。このオプションが省略された場合は、標準出力に出力します。</p> <p><code>-l log_file</code> 変換不可能な単語がある場合、そのエラー情報を保存するログファイル名を指定します。出力する文字コードは日本語 EUC です。このオプションが省略された場合は、ログファイルは作成されません。</p> <p><code>-h frequency_value</code> 出力ファイルに適用される頻度値を指定します。このオプションが省略された場合は、頻度値 1 が設定されます。</p>				
属性	次の属性については <code>attributes(5)</code> のマニュアルページを参照してください。				
	<table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWjwncu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWjwncu
属性タイプ	属性値				
使用条件	SUNWjwncu				
関連項目	wnnatod(1), wnnictutil(1)				

wnnstat(1)

名前	wnnstat - Wnn6 かな漢字変換サーバーの状態を表示する				
形式	<code>/usr/bin/wnnstat [-w] [-e] [-E] [-f] [-F] [-d] [-D] [-J -U -S -T] [-L language] [hostname]</code>				
機能説明	wnnstat は、指定されたホスト (<i>hostname</i>) 上で稼働している Wnn6 かな漢字変換サーバー (<i>jserver</i>) の状態を表示します。ホストが指定されていない場合は、wnnstat を実行したホスト上で稼働する Wnn6 かな漢字変換サーバーの状態を表示します。				
オプション	<p>以下のオプションが指定できます。</p> <ul style="list-style-type: none"> -w ユーザー名、ホスト名、ソケット番号、環境番号を表示します。 -e 環境番号、環境名、参照数を表示します。 -E 環境番号、環境名、参照数、付属語、辞書数 (辞書番号)、ファイル名を表示します。 -f 辞書ファイル識別子 (Fid)、種類、場所、参照数、ファイル名を表示します。 -F 辞書ファイル識別子 (Fid)、種類、場所、参照数、ファイル名を表示します。 -d 辞書番号、種類、略称、辞書ファイル識別子、ファイル名を表示します。 -D 辞書番号、種類、語数、更新の不可、頻度ファイル更新の不可、使用の不可、優先順位、[略称]、辞書ファイル識別子、ファイル名、[(頻度: 頻度ファイル名)][パスワード、(頻度パスワード)] を表示します。 -U 日本語 EUC (UJIS) で出力します。 -J JIS コードで出力します。 -S PC 漢字 (シフト JIS) コードで出力します。 -T UTF-8 コードで出力します。 -L <i>language</i> で指定した言語をサポートするサーバーの状態を表示します。Solaris の日本語環境では <i>ja</i> を指定します。 				
属性	次の属性については <code>attributes(5)</code> のマニュアルページを参照してください。				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWjwncu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWjwncu
属性タイプ	属性値				
使用条件	SUNWjwncu				
関連項目	<code>jserver(1M)</code>				

名前 | wnntouch - ファイルのヘッダーを書き換えて i ノードと合わせる

形式 | **/usr/bin/wnntouch** *binary_filename* ..

機能説明 | wnntouch は、*binary_filename* で指定したバイナリ辞書や付属語のヘッダーを書き換えて、ファイルの i ノードと合わせます。

属性 | 次の属性については `attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWjwncu

関連項目 | `wminatod(1)`

xjsi(1)

名前	xjsi - Wnn6 かな漢字変換サーバー/htt インタフェースモジュール
形式	htt -if xjsi -so -nosm [<i>htt_options...</i>]
機能説明	xjsi は、Wnn6 かな漢字変換サーバー jserver と X 入力サーバー htt を接続するためのインタフェースモジュールです。xjsi は共用ライブラリとして htt によって読み込まれます。
設定ファイル	xjsi は、起動時に次の設定ファイルを読み込みます。
ximrc	<p>X Input Method (XIM) かな漢字変換インタフェースを設定します。ximrc のパス名は、以下の優先順位で決定されます。</p> <ol style="list-style-type: none"> 1. リソース ximrcName 2. 環境変数 XIMRC 3. \$HOME/.Wnn6/ximrc 4. /etc/lib/locale/ja/wnn/ximrc 5. /usr/lib/locale/ja/wnn/ximrc
uumrc	<p>標準かな漢字変換インタフェースを設定します。uumrc のパス名は、以下の優先順位で決定されます。</p> <ol style="list-style-type: none"> 1. ximrc ファイルの setuumrc エントリ 2. @HOME/.Wnn6/uumrc 3. /etc/lib/locale/ja/wnn/ja/uumrc 4. /usr/lib/locale/ja/wnn/ja/uumrc
wnnenvrc	<p>かな漢字変換辞書/変換パラメタを設定します。wnnenvrc のパス名は、以下の優先順位で決定されます。</p> <ol style="list-style-type: none"> 1. uumrc ファイルの setconvenv エントリ 2. \$HOME/.Wnn6/wnnenvrc 3. /etc/lib/locale/ja/wnn/ja/wnnenvrc 4. /usr/lib/locale/ja/wnn/ja/wnnenvrc
uumkey	<p>キーの割り当てを設定します。uumkey のパス名は、以下の優先順位で決定されます。</p> <ol style="list-style-type: none"> 1. uumrc ファイルの setuumkey エントリ 2. /etc/lib/locale/ja/wnn/ja/uumkey 3. /usr/lib/locale/ja/wnn/ja/uumkey
rk/mode	<p>ローマ字かな漢字変換で使用するモード定義表を設定します。rk/mode のパス名は、以下の優先順位で決定されます。</p> <ol style="list-style-type: none"> 1. uumrc ファイルの setrkfile エントリ 2. /usr/lib/locale/ja/wnn/ja/rk/mode
cvt_xim_tbl	<p>X キーコード変換表を設定します。X キーコード変換表のパス名は、以下の優先順位で決定されます。</p> <ol style="list-style-type: none"> 1. リソース cvtximName 2. /usr/lib/locale/ja/wnn/cvt_xim_tbl

リソース	<p>xjsi は、名前 xjsi とクラス名 xjsi に、以下のリソース名とリソースクラスを持っています。</p> <p>serverName (ServerName) かな漢字変換サーバーのマシン名を指定します。xjsi は、次の優先順位で接続するかな漢字変換サーバーを決定します。</p> <ol style="list-style-type: none"> 1. リソースの設定 2. 環境変数 JSERVER 3. wnnenvrc の setconvenv エントリ 4. localhost 5. UNIX ドメインソケット <p>userName (UserName) Wnn6 の環境名として使用するユーザー名を指定します。xjsi は、次の優先順位で Wnn6 の環境名として使用するユーザー名を決定します。</p> <ol style="list-style-type: none"> 1. リソースの指定 2. 環境変数 WNNUSER 3. htt を実行したユーザーのユーザー名 <p>ximrcName (XimrcName) xjsi の初期化ファイル ximrc ファイルのパス名を指定します。ximrc ファイルの詳細については、wnn_ximrc(4) のマニュアルページを参照してください。</p> <p>cvtximName (CvtximName) X キーコード変換表のパス名を指定します。</p> <p>xjsi をマルチスクリーンシステムで使用する場合、以降のリソースは、スクリーンごとに個別に設定できます。スクリーンごとのサブの名前とサブのクラス名はそれぞれ screenN、ScreenN です。N にはスクリーン番号が入ります。</p> <p>foreground (Foreground) テキストの色を指定します。デフォルトは、black です。</p> <p>background (Background) 背景の色を指定します。デフォルトは、white です。</p> <p>borderColor (BorderColor) 枠の色を指定します。デフォルトは、black です。</p> <p>fontSet (FontSet) フォント名のリストを指定します。リストの書式は、fontname { , fontname } です。</p>
ファイル	<p>/usr/openwin/lib/locale/ja/xim/xjsi.so xjsi インタフェースモジュール</p> <p>ximrc XIM かな漢字変換インタフェースの設定ファイル</p> <p>uumrc 標準かな漢字変換インタフェースの設定ファイル</p>

xjsi(1)

wnnenvrc
かな漢字変換辞書/変換パラメタ設定ファイル

uumkey
キーの割り当て設定ファイル

cvt_xim_tbl
Xキーコード変換表

mode
ローマ字かな変換設定ファイル

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWjwncx

関連項目 htt(1),wnn6setup(1) wnn_ximrc(4), uumrc(4), wnnenvrc(4), uumkey(4),
wnn_cvt_xim_tbl(4), wnn_mode(4), wnn_automaton(4)