

Tru64 UNIX

日本語機能ガイドブック

Part Number: AA-RK3FE-TE

2002 年 11 月

オペレーティング・システム: Tru64 UNIX V5.1B

本書では、Tru64 UNIX オペレーティング・システムの日本語機能を使用するために必要な諸設定、および操作方法について説明しています。また、Motif アプリケーションを開発するユーザ向けの情報として、Motif の日本語機能についても説明します。

日本ヒューレット・パカード株式会社

© 2002 日本ヒューレット・パッカード株式会社

本書の著作権は日本ヒューレット・パッカード株式会社が保有しており、本書中の解説および図、表は日本ヒューレット・パッカードの文書による許可なしに、その全体または一部を、いかなる場合にも再版あるいは複製することを禁じます。

日本ヒューレット・パッカードは、弊社または弊社の指定する会社から納入された機器以外の機器で対象ソフトウェアを使用した場合、その性能あるいは信頼性について一切責任を負いかねます。

本書に記載されている事項は、予告なく変更されることがありますので、あらかじめご承知おきください。万一、本書の記述に誤りがあった場合でも、弊社は一切その責任を負いかねます。

本書で解説するソフトウェア(対象ソフトウェア)は、所定のライセンス契約が締結された場合に限り、その使用あるいは複製が許可されます。

COMPAQ, Compaq ロゴ, Digital ロゴは U.S. Patent and Trademark Office に登録されています。Alpha, AlphaServer, NonStop, TruCluster, および Tru64 は米国 Compaq Computer Corporation の商標です。

Microsoft, Windows および Windows NT は米国 Microsoft 社の登録商標です。Intel は米国 Intel 社の登録商標です。Motif, OSF/1, UNIX, The Open Group および X/Open は、The Open Group の米国ならびに他の国における商標です。

このドキュメントに記載されているその他の会社名および製品名は、各社の商標または登録商標です。

目次

まえがき

1 Tru64 UNIX の日本語機能

1.1	国際化機能および国際化コマンド/ユーティリティ	1-2
1.1.1	国際化機能	1-2
1.1.2	コードセット	1-2
1.1.3	日本語ロケールと国際化コマンド/ユーティリティ	1-2
1.2	アジア言語処理用 tty サブシステム	1-3
1.2.1	STREAMS tty によるかな漢字変換入力	1-3
1.2.2	オンデマンド・フォント・ローディング(ODL)	1-3
1.2.3	コード変換	1-3
1.3	Wnn かな漢字変換システム	1-4
1.4	Mule エディタ	1-4
1.5	FD アクセス・ユーティリティ (FDIO)	1-4
1.6	かな漢字変換	1-4
1.6.1	かな漢字変換用辞書とデータ・ファイル	1-4
1.6.2	変換キー定義機能(kkseq)	1-5
1.7	日本語プリンタ用フィルタ	1-5
1.8	日本語 Advanced Printing Software	1-5
1.9	漢字コード変換ユーティリティ	1-5
1.10	変換キー定義ライブラリ(IMLIB)	1-5
1.11	ユーザ定義文字作成ツール	1-6
1.12	System V MNLS から Tru64 UNIX への 国際化機能移行支援 ツール	1-6
1.13	日本語入力システム	1-6

1.14	日本語リファレンス・ページ	1-6
1.15	日本語メッセージ・カタログ	1-6
1.16	日本語ドキュメント全文検索機能	1-7
1.17	日本語 TrueType フォント・キット	1-7
2	日本語環境の基本設定	
2.1	日本語環境のインストール	2-1
2.2	国際化環境の設定	2-3
2.3	CDE における言語設定	2-4
2.3.1	CDE における省略時の言語の設定	2-5
2.4	日本語ロケールの設定	2-6
2.4.1	日本語ロケールの設定方法	2-6
2.4.2	Unicode および dense コード・ロケールの切替え	2-8
2.5	日本語端末の設定	2-8
2.5.1	端末コードとアプリケーション・コードの設定	2-9
2.5.1.1	端末コードの設定	2-10
2.5.1.2	アプリケーション・コードの設定	2-10
2.5.2	8-bit データの取り扱い	2-11
2.5.3	端末の設定状態の確認	2-12
2.6	ファイル・コードの変換	2-12
2.7	メッセージ・カタログ	2-14
2.8	日本語リファレンス・ページの設定	2-15
3	日本語入力システムの設定	
3.1	dxjim	3-1
3.1.1	dxjim の起動	3-1
3.1.2	dxjim による日本語入力の概要	3-2
3.2	VJE-Delta	3-2
3.3	WX3	3-2

4	日本語プリンタの設定	
4.1	プリンタ・コード	4-1
4.2	日本語プリンタの設定方法	4-2
4.3	ESC/P プリンタを利用するための設定	4-3
4.4	ESC/Page プリンタを利用するための設定	4-4
4.5	Advanced Printing Software のインストレーションと設定 ...	4-5
5	日本語キーボードの設定	
5.1	CDE におけるキーボード設定	5-1
5.1.1	カナ・モード	5-2
5.1.2	変換キー	5-2
5.1.3	カナ LED	5-3
5.2	コンソール・モードおよびシングルユーザ・モードにおける日本語キーボードの設定	5-3
6	日本語ドキュメント全文検索機能のインストールおよび設定	
6.1	midoc のインストレーション	6-2
6.1.1	キットの展開先の確保	6-2
6.1.2	キットからのインストール	6-3
6.1.3	インストール後の設定作業	6-3
6.1.3.1	アイコン・ファイルのコピー	6-3
6.1.3.2	Perl スクリプトのコピー	6-4
6.1.3.3	マップ・ファイルの編集	6-4
6.1.4	検索の実行	6-5
6.1.5	midoc のアンインストール	6-5
6.2	miman のインストレーション	6-5
6.2.1	キットの展開先の確保	6-6
6.2.2	キットからのインストール	6-6

6.2.3	キット展開後の作業	6-6
6.2.3.1	コマンド形式で利用するための設定	6-7
6.2.3.2	CGI 経由で利用するための設定	6-7
6.2.4	miman のアンインストールとその他の注意事項	6-8
 7 dxjim による日本語入力		
7.1	dxjim による日本語変換機能の概要	7-1
7.2	dxjim の使用方法	7-2
7.2.1	dxjim の設定	7-3
7.2.2	入力スタイル	7-4
7.2.3	リソース	7-4
7.3	日本語入力の変換キー	7-8
7.3.1	変換キー定義の変更	7-14
7.4	変換規則	7-14
7.4.1	ローマ字かな変換対応表	7-15
7.4.2	特殊文字変換対応表	7-17
7.4.3	記号変換対応表	7-18
7.4.4	コード入力	7-19
7.5	キーボードの「かなモード」の使用	7-19
7.6	変換キー付き日本語キーボードのサポート	7-20
 8 日本語 MH		
8.1	メール	8-1
8.2	日本語対応のフロント・エンド	8-1
8.3	MH の操作方法	8-2
8.3.1	メールの送信	8-2
8.4	コード変換	8-3
8.4.1	アプリケーション・コードとメール変換コード	8-4
8.4.1.1	メール変換コードの設定	8-4

8.4.1.2	メール変換コードの認識	8-5
8.4.2	メール送信時のコード変換	8-5
8.4.3	メール受信時のコード変換	8-6
8.5	MH コマンド一覧表	8-6
A 文字コード		
A.1	文字セット	A-1
A.1.1	7 単位系 JIS 漢字コードセット	A-3
A.1.2	8 単位系 JIS 漢字コードセット	A-4
A.1.3	日本語 EUC コードセット	A-5
A.1.4	DEC 漢字コードセット	A-7
A.1.5	Super DEC 漢字コードセット	A-8
A.1.6	シフト JIS コードセット	A-9
A.1.7	UTF-8	A-11
B 日本語ロケールの定義内容		
B.1	CHARMAP	B-1
B.2	LC_CTYPE	B-2
B.3	LC_COLLATE	B-5
B.4	LC_MESSAGES	B-5
B.5	LC_MONETARY	B-6
B.6	LC_NUMERIC	B-6
B.7	LC_TIME	B-7
C Motif の日本語機能		
C.1	X Window System	C-1
C.1.1	X Locale	C-1
C.1.2	入力サーバ	C-1
C.1.3	X Toolkit Intrinsics	C-2
C.1.4	Xlib	C-2

C.2	Motif ツールキット(Xm) および HP による Motif ツールキット の拡張 (DXm)	C-2
C.2.1	日本語の表示	C-2
C.2.2	日本語の入力	C-3
C.2.3	ロケールの設定	C-3
C.2.4	日本語 Motif 1.1.3 との互換性	C-4
C.2.5	DXmCSText のコンパウンド・ストリング	C-4
C.2.6	DXmCSText の日本語入力の制御	C-5
C.3	クライアント間の通信	C-5
C.4	日本語フォント	C-5
C.4.1	75 dpi フォント	C-5
C.4.2	100 dpi フォント	C-6
C.5	フォント・コンバータ	C-7
C.5.1	フォント・ファイル変換の方法	C-7
C.5.2	フォント・ファイル変換の例	C-8

索引

例

4-1	/etc/printcap の例	4-3
-----	------------------------	-----

図

2-1	日本語環境を使用するために必要とするソフトウェア・サブ セット	2-2
2-2	端末ドライバ (Asian tty)	2-9

表

1-1	提供する日本語 TrueType フォントの種類	1-7
2-1	CDE の言語設定オプションで選択できる日本語ロケール	2-5
2-2	Tru64 UNIX で有効な日本語ロケール名	2-6

2-3	ロケール・データベース	2-7
2-4	有効なカテゴリ名	2-7
2-5	端末コードとして指定できるコードセット名	2-10
2-6	アプリケーション・コードとして指定できるコードセット名	2-10
2-7	iconv で指定可能なコードセット名	2-13
2-8	iconv で指定可能なコードセット名 (他社汎用機)	2-14
2-9	日本語メッセージ・カタログのあるディレクトリ	2-14
2-10	リファンレンス・ページのセクション番号	2-15
7-1	日本語入力の変換機能	7-1
7-2	JVMS 変換キー定義	7-9
7-3	EVEJ 変換キー定義	7-10
7-4	KEYPAD 変換キー定義	7-11
7-5	TARO 変換キー定義	7-12
7-6	CTRL 変換キー定義	7-13
7-7	変換キー付き日本語キーボードの変換キー定義	7-20
8-1	MH コマンド一覧	8-7
A-1	JIS による文字セットの分類	A-1
A-2	ASCII と JIS X 0201 LH	A-2
A-3	各コードセットで利用できる文字セット	A-3
A-4	7-bit JIS コードセットの文字コード	A-3
A-5	8-bit JIS コードセットの文字コード	A-5
A-6	日本語 EUC の文字コード	A-6
A-7	DEC 漢字文字セット 90 年版	A-7
A-8	DEC 漢字コードの文字コード	A-8
A-9	Super DEC 漢字の文字コード	A-9
A-10	Super DEC 漢字におけるユーザ定義文字領域	A-9
A-11	シフト JIS コードの符号化	A-10
A-12	漢字, ユーザ定義文字の区点との対応	A-10
A-13	UCS-4 のエンコーディング範囲と UTF-8 のビット配列	A-11

B-1	日本語コードセットの属性	B-1
B-2	標準の文字クラス	B-2
B-3	拡張文字クラス	B-3
B-4	toupper および tolower に定義されているアルファベット	B-5
B-5	LC_MESSAGES の定義	B-6
B-6	LC_MONETARY の定義	B-6
B-7	LC_NUMERIC の定義	B-7
B-8	LC_TIME の定義	B-7

まえがき

本書では、Tru64 UNIX オペレーティング・システムの日本語機能を使用するために必要なシステム設定、および操作方法について説明しており、さらに関連知識についてまとめています。本書で、説明する個々のコマンドの詳細については、それぞれのコマンドのリファレンス・ページを参照してください。

本書の対象読者

本書は、Tru64 UNIX オペレーティング・システムを日本語環境で使用するユーザを対象としています。また、日本語 Motif を使用してアプリケーションの開発を行うユーザも対象としています。

本書の構成

本書は次の章から構成されています。

第 1 章	Tru64 UNIX の日本語機能の概要について説明します。
第 2 章	Tru64 UNIX の日本語機能を使用するために必要な基本的な設定について説明します。
第 3 章	Tru64 UNIX の日本語入力システムの設定について説明します。
第 4 章	Tru64 UNIX における日本語プリンタの設定について説明します。
第 5 章	Tru64 UNIX における日本語キーボードの設定について説明します。
第 6 章	日本語ドキュメント全文検索機能 midoc および miman のインストールおよび設定方法について説明します。
第 7 章	Tru64 UNIX の標準の日本語入力機能である dxjim について説明します。
第 8 章	日本語 MH (Message Handling System) を使用するために必要な設定について説明します。
付録 A	Tru64 UNIX が処理することができる日本語文字コードと、端末コード、アプリケーション・コードの概念を説明します。

付録 B	Tru64 UNIX でサポートするロケールの定義内容について説明します。
付録 C	Tru64 UNIX で提供する Motif の日本語機能について説明します。

関連資料

- 『DEC 入力サーバ・ライブラリ』
- 『IMLIB ライブラリ・リファレンス・マニュアル』
- 『ユーザ・キー定義 利用者の手引』
- 『STREAMS tty によるかな漢字変換』
- 『Wnn によるかな漢字変換』
- 『ユーザ定義文字』
- Advanced Printing Software 日本語ドキュメント
- 『国際化プログラミング』(共立出版)

本書で使用する表記法

本書では次の表記法を使用します。

%

\$

パーセント記号は、C シェルのシステム・プロンプトを表します。ドル記号は、Bourne シェル、Korn シェル、および POSIX シェルの場合のシステム・プロンプトを表します。

#

番号記号は root としてログインした場合のシステム・プロンプトを表します。

file

イタリック体 (斜体) は、変数値、プレースホルダ、および関数の引数名を示します。

[]

{ }

構文定義では、大カッコはオプションの項目を示し、中カッコは必須項目を示します。大カッコまたは中カッコの中の項目を縦線で区切っている場合

は，そこに併記されている項目の中から 1 つの項目を選択することを示します。

...

構文定義では，水平の反復記号は，前の項目を 1 回以上繰り返して使用できることを示します。

:

垂直の反復記号は，実際には存在する例の一部が省略されていることを示します。

cat(1)

リファレンス・ページの参照には，該当するセクション番号をカッコ内に示します。たとえば，cat(1) は，cat コマンドについての情報が，リファレンス・ページのセクション 1 に記載されていることを示します。

Ctrl/x

この記号は，スラッシュの前に指定されているキーを押しながら，スラッシュの後のキーまたはマウス・ボタンを押すことを示します。例中では，このようなキーの組み合わせは，四角あるいは大カッコで囲まれて示されます(たとえば，`Ctrl/C`)。

`Return`

四角で囲まれたキー名はユーザがそのキーを押すことを示します。



Tru64 UNIX の日本語機能

Tru64 UNIX は、日本語を含む世界各国の言語に対応した国際化オペレーティング・システムです。Tru64 UNIX における日本語処理機能のほとんどは、国際標準の仕様に従って実装された国際化機能を使用して提供されています。Tru64 UNIX では、以下の日本語処理機能を提供します。

- 国際化機能および国際化コマンド/ユーティリティ
- アジア言語処理用ttyサブシステム
- Wnnかな漢字変換システム
- Mule エディタ
- FDアクセス・ユーティリティ (FDIO)
- かな漢字変換システムとライブラリ
- 日本語プリンタ用フィルタ
- 日本語 Advanced Printing Software
- 漢字コード変換ユーティリティ (iconv)
- 変換キー定義ライブラリ (IMLIB)
- ユーザ定義文字作成ツール
- dxjim 日本語入力システム
- VJE-Delta 日本語入力システム
- WX3 日本語入力システム
- 日本語リファレンス・ページ
- 日本語メッセージ・カタログ
- 日本語ドキュメント全文検索 (miman および midoc)
- System V MNLS から Tru64 UNIX への国際化機能移行支援ツール
- 日本語 TrueType フォント

また、Tru64 UNIX では、ほとんどのユーザ・インタフェースが日本語で表示されます。

以降の節で、これらの日本語処理機能について概要を説明します。

1.1 国際化機能および国際化コマンド/ユーティリティ

1.1.1 国際化機能

Tru64 UNIXでは XPG4 (X/Open CAE Specification System Interfaces and Headers, Issue 4)に定められたインタフェースを使用して国際化されたユーティリティを提供します。これはロケール (言語や文化/慣習に依存したユーザ環境を定義したもの) の設定を変更することにより、複数の言語およびコードセットの中から、使用する言語とコードセットを選択する機能を提供します。

1.1.2 コードセット

Tru64 UNIX の日本語環境では次の 5 種類のコードセットをサポートします。

- 日本語 EUC
- シフト JIS
- DEC 漢字
- Super DEC 漢字
- UTF-8

個々のコードセットについての詳細は、付録 Aを参照してください。

ユーザは日本語ロケールを指定するときに、5 種類のコードセットから任意のものを指定することができます。Super DEC 漢字は DEC 漢字と日本語 EUC のスーパーセットになるように設計されています。DEC 漢字文字セットの詳細は、『漢字コード表』に記述されています。

1.1.3 日本語ロケールと国際化コマンド/ユーティリティ

Tru64 UNIX では以下の日本語ロケールをサポートします。

- ja_JP.eucJP または ja_JP (日本語 EUC)
- ja_JP.SJIS (シフト JIS)
- ja_JP.deckanji (DEC 漢字)
- ja_JP.sdeckanji (Super DEC 漢字)

- ja_JP.UTF-8 (UTF-8)

これらを動的に切り替えることにより、各国際化コマンドおよびユーティリティで日本語を処理することができます。

1.2 アジア言語処理用 `tty` サブシステム

アジア言語処理用 `tty` サブシステムは、非同期通信方式でシステムに接続されている端末装置をサポートします。Tru64 UNIXでは、日本語端末をサポートするためアジア言語処理用 `tty` サブシステムを提供し、以下の日本語機能をサポートしています(以下、説明では日本語 `tty` と呼びます)。なお、日本語 `tty` を使用するためには、`stty` コマンドで日本語ラインディシプリン (`jdec`) を選択しておく必要があります。

1.2.1 STREAMS `tty` によるかな漢字変換入力

ユーザは `stty` コマンドでセットアップを行うことによって、日本語 `tty` による、かな漢字変換入力機能を使用することができます。この機能により、アプリケーションは標準の入力ルーチンを呼び出すだけで日本語を入力することができます。詳細は『STREAMS `tty` によるかな漢字変換』を参照してください。

1.2.2 オンデマンド・フォント・ローディング(ODL)

ユーザは、`stty` コマンドで ODL (On-Demand font Loading) ファイルをセットアップすることによって、日本語 `tty` による ODL 機能を使用できます。この機能により、ユーザ定義文字フォントを必要なときに動的に端末やプリンタにローディングできます。

1.2.3 コード変換

日本語 `tty` では端末コードとアプリケーション・コードとの間でのコード変換を行うことができます。

端末コードとして下記のコードをサポートします。

- 7 ビット JIS
- 8 ビット JIS
- DEC 漢字
- 日本語 EUC

- シフト JIS
- UTF-8

アプリケーション・コードとして下記のコードをサポートします。

- 日本語 EUC
- シフト JIS
- DEC 漢字
- Super DEC 漢字
- UTF-8

1.3 Wnn かな漢字変換システム

フリー・ソフトウェアである Wnn 4.109 を日本語 Tru64 UNIX に移植したもので、かな漢字変換のフロント・エンドとして使用することができます。また、HP が提供するかな漢字変換辞書へのアクセスを可能にするための拡張が行われています。

1.4 Mule エディタ

Tru64 UNIX ではフリー・ソフトウェアである多言語対応エディタ Mule 2.3 をサポートしており、かな漢字変換入力その他の日本語処理機能を提供します。

1.5 FD アクセス・ユーティリティ (FDIO)

FDIO は Alpha システムに内蔵/接続されるフロッピー・ドライブに対するアクセス・ユーティリティです。このユーティリティは MS-DOS でフォーマットされたフロッピー・ディスクへの read/write 機能を提供します。テキスト・モードでは、ファイルの転送や表示の際に漢字コードや改行コードの変換を自動的に行います。

1.6 かな漢字変換

1.6.1 かな漢字変換用辞書とデータ・ファイル

かな漢字変換を行うための辞書は、一般的な国語辞書に加えて人名・地名などの固有名詞を含み、全体で約 10 万語のものを提供しています。システム全

体で 1 つ用意されているシステム辞書に加えて、ユーザごとに頻度情報などを保持するための個人辞書 (学習用辞書) があります。

また、ユーザごとに文節学習・文節切り学習の結果を保存するための文節学習用データ・ファイルがあります。

1.6.2 変換キー定義機能(kkseq)

ユーザごとにな漢字変換キー・シーケンスを定義するための機能を提供します。

1.7 日本語プリンタ用フィルタ

日本語 レーザ・プリンタなどを通信回線経由でハードコピー端末として利用できます。Tru64 UNIX のプリンタ・スプーラを用いて、日本語文書の印刷が可能です。また、本システムでサポートしているプリンタを使用して日本語文書の印刷を行うための出力フィルタ・プログラムが用意されています。

1.8 日本語 Advanced Printing Software

Advanced Printing Software のキットをインストールした後に Advanced Printing Software の日本語キットをインストールすることにより、日本語プリンタに日本語テキスト・ファイルおよび日本語 PostScript ファイル (プリンタが PostScript をサポートする場合) を出力できるようになります。詳細については 第 4 章 を参照してください。

1.9 漢字コード変換ユーティリティ

iconvユーティリティは、日本語 EUC、シフト JIS、DEC 漢字、Super DEC 漢字、7 ビット JIS、Unicode の各コードセット間でのコード変換を行います。また、IBM 漢字システム文字、富士通 JEF コード、日立 KEIS のそれぞれのコードセットからのファイル・コード変換をサポートしています。詳細については 2.6 節を参照してください。

1.10 変換キー定義ライブラリ(IMLIB)

日本語入力のためのキー定義処理用のプログラミング・ライブラリです。アプリケーションで IMLIB を使用することにより、ユーザの日本語入力キー定義に基づいた処理を行うことができます。IMLIB 用のキー定義では、kkseq より複雑な入力インタフェースの記述が可能です。また、kkseq のキー定義

ファイルから IMLIB 用のキー定義ファイルに変換するツールが提供されています。キー定義をカスタマイズするときには『ユーザ・キー定義 利用者の手引』を、IMLIB を使用してアプリケーションを開発するときには『IMLIB Tru64 UNIX ライブラリ・リファレンス・マニュアル』を参照してください。

1.11 ユーザ定義文字作成ツール

ユーザ定義文字を作成する `credit` ユーティリティと、作成したフォントをフォント・ロード用のファイルに変換する `cgen` ユーティリティの 2 つのツールによって、ユーザ定義文字の作成、登録、表示を行うことができます。詳細については『ユーザ定義文字』を参照してください。

1.12 System V MNLS から Tru64 UNIX への 国際化機能移行 支援ツール

System V MNLS の国際化機能を使用して作成されたアプリケーションの Tru64 UNIX への移行を支援するツールです。プログラム中で使用されている System V MNLS の機能の移行に関する情報の出力、メッセージ・ソース・ファイルの変換などを行います。詳細については『System V MNLS から Tru64 UNIX への国際化機能移行ガイド』を参照してください。

1.13 日本語入力システム

Tru64 UNIX では、ウィンドウ環境の標準の日本語入力サーバとして `dxjim` を提供しています。さらに、PC 上の FEP として実績のある VJE-Delta¹ と WX3² をサポートしています。

これらの入力システムの詳細については、第 3 章を参照してください。

1.14 日本語リファレンス・ページ

Tru64 UNIX の日本語リファレンス・ページをオンラインで参照することができます。

1.15 日本語メッセージ・カタログ

主要なコマンドおよびユーティリティが発行するメッセージを日本語で表示することができます。

¹ VJE-Delta は株式会社バックスの登録商標です。

² WX3 はエー・アイ・ソフト株式会社の商品名です。

1.16 日本語ドキュメント全文検索機能

MitakeSearch³ を利用した、日本語オンライン・ドキュメントおよびリファレンス・ページの日本語全文検索ツール (midoc および miman) を提供します。これらの機能のインストレーションおよび設定方法については 第 6 章 を参照してください。

1.17 日本語 TrueType フォント・キット

本リリースでは、日本語 Tru64 UNIX システムに日本語 TrueType フォントを追加するための日本語 TrueType フォント・キットを提供しています。このキットをインストールすることによって、Netscape Communicator をはじめとするデスクトップ・アプリケーションで日本語テキストが TrueType フォントで表示できるようになります。

本リリースで提供する日本語 TrueType フォントは以下のとおりです。

表 1-1: 提供する日本語 TrueType フォントの種類

フォント名	書体
G2 サンセリフ-U	極太ゴシック体
ゴシック-B	角ゴシック体
羽衣-M	筆文字書体
本明朝-L	明朝体
シリウス-L II	細めの丸ゴシック体
平成 明朝	明朝体

日本語 TrueType フォント・キットのインストレーション・キットおよびドキュメントは「日本語追加機能」CD-ROM の Japanese_TrueType_Fonts/ ディレクトリに含まれています。

提供する TrueType フォントについての詳細、フォント・キットのインストレーション方法、使用方法、および使用上の注意事項については、日本語 TrueType フォント・キットのインストレーション・ガイドおよびリリース・

³ MitakeSearch はコンパックコンピュータ株式会社の登録商標です。

ノートを参照してください。これらのドキュメントは「日本語追加機能」CD-ROM の下記のディレクトリに含まれています。

- インストレーション・ガイド：

`Japanese_TrueType_Fonts/doc/install_guide_eucjp.txt`

- リリース・ノート：

`Japanese_TrueType_Fonts/doc/release_notes_eucjp.txt`

日本語環境の基本設定

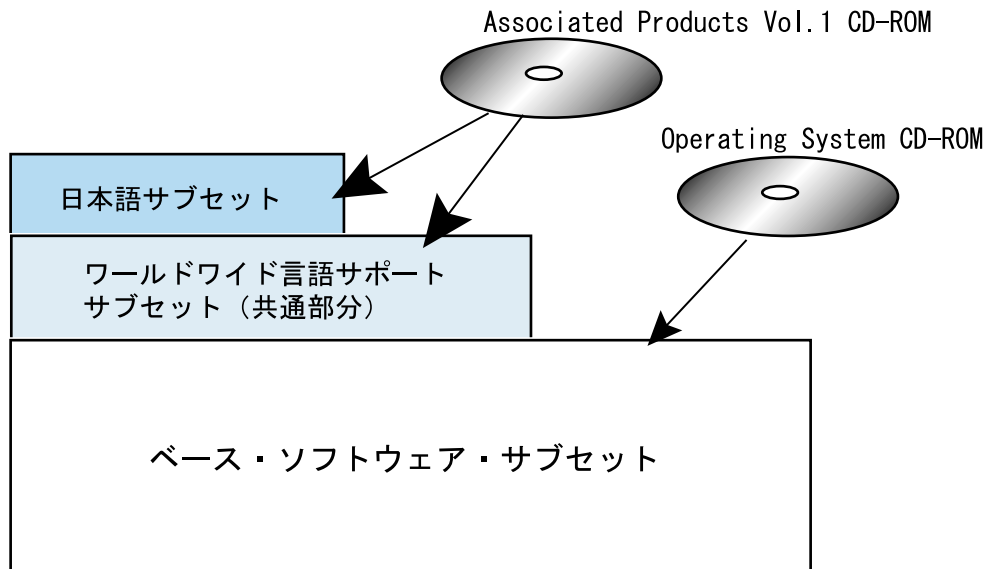
この章では、国際化されたソフトウェアを日本語の環境で使用するためのロケールの指定方法や、端末の入出力に関連した漢字コードの設定などの日本語環境の設定について説明します。

- 日本語環境のインストール
- 国際化環境の設定
- CDE の言語設定
- 言語 (ロケール) の設定
- 日本語端末の設定
- iconv コマンドによる日本語コード変換
- メッセージ・カタログの設定
- 日本語リファレンス・ページの設定

2.1 日本語環境のインストール

Tru64 UNIX で提供する日本語機能を使用するためには、Tru64 UNIX のベース・オペレーティング・システムに加えて、ワールドワイド言語サポート・サブセットと日本語サブセットをインストールする必要があります。

図 2-1: 日本語環境を使用するために必要とするソフトウェア・サブセット



日本語機能を使用するために必要なこれらのサブセットのインストールは、インストレーション・プロシージャで日本語を選択することにより自動的に行われます。具体的なインストレーション手順については、『インストレーション・ガイド』および『インストレーション・ガイド — 上級ユーザ編』を参照してください。

なお、下記の日本語機能を使用するためには、インストレーション・プロシージャによる Tru64 UNIX の日本語環境のインストレーションの後に、それぞれのソフトウェア・キットのインストレーションが必要になります。

- VJE-Delta

日本語入力システム VJE-Delta のキットは、「日本語入力システム VJE-Delta」CD-ROM に含まれています。このソフトウェアのインストール方法については、VJE-Delta のユーザーズ・ガイドを参照してください。

- WX3

日本語入力システム WX3 のキットは、「日本語入力システム WX3」CD-ROM 含まれています。このソフトウェアのインストール方法については、WX3 のユーザーズ・ガイドを参照してください。

- 日本語 Advanced Printing Software

Advanced Printing Software の日本語キットは、「Associated Products Volume 1」CD-ROM に含まれています。このソフトウェアのインストールについては、4.5 節を参照してください。

- midoc および miman

日本語ドキュメント全文検索機能 midoc とリファレンス・ページ全文検索機能 miman のキットおよびインストールについては、第 6 章を参照してください。

2.2 国際化環境の設定

Tru64 UNIX の日本語機能を使用するためには、まず最初に、次にあげる国際化環境用の環境変数を設定します。

- I18NPATH

国際化環境の仮想的なルート・ディレクトリとなるディレクトリを設定します。

- PATH

jvi などの日本語サブセットに含まれるコマンドを優先して実行するように、PATH の先頭に \$I18NPATH/bin を設定します。

これらの環境変数の設定を正しく行うために、Tru64 UNIX では次に示す 3 つのスクリプト・ファイルを用意しています。

1. /etc/i18n_profile

ユーザの .profile ファイルに含めるべき設定が記述されています。

2. /etc/i18n_login

ユーザの .cshrc ファイルに含めるべき設定が記述されています。

3. /etc/i18n_path

他のコマンドが国際化環境用のコマンド、ユーティリティを使用するためのパスの設定が記述されています。

これらは、必要に応じて変更するためのテンプレート・ファイルです。システム管理者は、システムの構成や環境に合わせて /etc/i18n_profile および /etc/i18n_login の内容を適当に変更することができます。

注意

`/etc/il8n_path` はシステムのディレクトリ構造に依存しているため変更してはいけません。

`/usr/sbin/il8nsetup` というコマンドを使用すると、ユーザのスタートアップ・スクリプト (`$HOME/.profile` および `$HOME/.cshrc`) の最後にスケルトン・ファイルの内容が追加されます。この後でエディタを使用して、これらの追加された内容をファイルの中の適当な位置に移動します。

システム管理者 (root) であれば、`/usr/sbin/il8nsetup` を使用して他のユーザのスタートアップ・スクリプトの変更を行うことができます。この場合、次に示すように引数としてユーザ名を指定します。この例ではユーザ `kanai` と `shimada` のスタートアップ・ファイルが変更されます。

```
# /usr/sbin/il8nsetup kanai shimada
```

以上のスタートアップ・ファイルが正しく変更された場合、次回以降ログインしたとき、あるいは `eval (sh の場合)`、`source (csh の場合)` で現在のシェル環境に設定を反映すると Tru64 UNIX の国際化環境のコマンドやユーティリティを使用することができるようになります。

注意

これ以降解説する日本語環境の設定や操作は、この節で解説した環境変数の設定が行われたものと仮定しています。日本語環境の設定が正常に行われない場合、まずこれらの環境変数の値を調べてください。

2.3 CDE における言語設定

CDE におけるユーザごとの言語設定は、システムへのログイン時に、ログイン・ダイアログで「オプション」メニューの「言語」サブメニューから使用言語を選択することによって行います。「言語」サブメニューから、表 2-1 に示すいずれかの日本語ロケールを選択することにより、日本語でシステムを利用することができます。

表 2-1: CDE の言語設定オプションで選択できる日本語ロケール

日本語	ja_JP
日本語 (DEC 漢字)	ja_JP.deckanji
日本語 (EUC)	ja_JP.eucJP
日本語 (Super DEC 漢字)	ja_JP.sdeckanji
日本語 (Unicode)	ja_JP.UTF-8
日本語 (シフト JIS)	ja_JP.SJIS

CDE セッションの環境変数 LANG の値は、ログイン時に選択した言語にしたがって自動的に設定されます。このため、`.login` などの起動時スクリプトで LANG 環境変数を設定する必要はありません。

注意

CDE セッションをご使用の場合は、起動時スクリプトの中では環境変数 LANG を定義しないようにしてください。ログイン時に選択した言語とスクリプトが設定した環境変数 LANG の値が矛盾すると、セッションが正しく起動できない場合があります。

2.3.1 CDE における省略時の言語の設定

ワールドワイド言語サポート・ソフトウェアのインストール時に日本語だけを選択した場合、ウィンドウ環境の省略時の言語が自動的に日本語 EUC (ja_JP.eucJP) に設定されます。

ウィンドウ環境の省略時の言語設定を EUC 以外の日本語ロケールにしたい場合、あるいはワールドワイド言語サポート・ソフトウェアのインストール時に複数の言語を選択した場合は、以下の手順で日本語ウィンドウ環境の省略時の言語を設定 (変更) することができます。

1. `/etc/dt/config/Xconfig` ファイルの有無を調べ、存在する場合は削除します。
2. 以下の手順で新しい Xconfig ファイルを作成します。

```
# cd /etc/dt/config
# ln -s Xconfig.ロケール名 Xconfig
```

次の例では、省略時の言語を日本語 EUC に設定しています。

```
# ln -s Xconfig.ja_JP.euc.JP Xconfig
```

この設定はログイン画面の言語と省略時のセッション言語の両方に対して有効です。

2.4 日本語ロケールの設定

2.4.1 日本語ロケールの設定方法

Tru64 UNIX の国際化されたコマンドやライブラリを日本語に対応させるためには、明示的に言語、テリトリ、コードセット名の3つの組み合わせから表されるロケール (locale) を指定します。

ロケールを表すロケール名は、次のような形式で指定します。

Language_Territory.Codeset

Language , *Territory* , *Codeset* には、それぞれシステムで定義されている言語名、テリトリ名、文字コードセット名の二ーモニックを指定します。

Tru64 UNIX では、表 2-2 に示す日本語ロケールを使用することができます。これらはすべて「日本国内で使用されている日本語」というロケールを指定するものですが、文字データを表すための使用文字コードセットを明示的に区別しています。

表 2-2: Tru64 UNIX で有効な日本語ロケール名

ロケール名	使用する文字コードセット名
ja_JP.deckanji	DEC 漢字コードを使用
ja_JP.sdeckanji	Super DEC 漢字コードを使用
ja_JP.eucJP	日本語 EUC を使用
ja_JP.SJIS	シフト JIS を使用
ja_JP	日本語 EUC を使用
ja_JP.UTF-8	UTF-8 を使用

Tru64 UNIX では、インストール時に日本語ロケール用のロケール・データベースを表 2-3 に示す場所に格納します。

表 2-3: ロケール・データベース

ロケール名	ロケール・データベース
ja_JP.deckanji	/usr/lib/nls/loc/ja_JP.deckanji
ja_JP.sdeckanji	/usr/lib/nls/loc/ja_JP.sdeckanji
ja_JP.eucJP	/usr/lib/nls/loc/ja_JP.eucJP
ja_JP.SJIS	/usr/lib/nls/loc/ja_JP.SJIS
ja_JP.UTF-8	/usr/lib/nls/loc/ja_JP.UTF-8

日本語ロケールを指定するには、環境変数 LANG に表 2-2 で示したロケール名のうちのいずれか 1 つを設定します。Tru64 UNIX の国際化されたコマンドやライブラリは、起動時に環境変数 LANG を参照して、そこに設定されているロケール名に対応します。

Tru64 UNIX では、国際化されたコマンドやライブラリの処理の内容がロケールによって異なる状況を表 2-4 の 6 種類のカテゴリに分類しています。これらのカテゴリ名は、環境変数の名前と同じです。ユーザは、これらの環境変数に適切なロケール名をそれぞれ独立に指定することができるため、処理の内容に応じて細かなロケールの指定を行うことが可能です。

表 2-4: 有効なカテゴリ名

カテゴリ名(環境変数名)	選択内容
LC_COLLATE	照合順序 (collation)
LC_CTYPE	文字の分類、大文字小文字などの文字の属性
LC_MESSAGES	メッセージの表示形式
LC_MONETARY	金銭の表記形式
LC_NUMERIC	数値データの表記形式
LC_TIME	日時、時間の表記形式

これらのカテゴリのすべてに、同時に同じロケール名を設定するには環境変数 LC_ALL を使用します。

システムは次の優先順位で各環境変数を参照し、正しいロケール名が設定されていればその変数値を使用してロケールの設定を行います。

1. 環境変数 LC_ALL
2. 環境変数 LC_COLLATE, LC_CTYPE, LC_MESSAGES, LC_MONETARY, LC_NUMERIC, LC_TIME
3. 環境変数 LANG

環境変数 LANG および各カテゴリにロケール名を何も指定していない場合や不適当なロケール名を指定した場合、C ロケールが指定されたものとされます。C ロケールは、ASCII 文字しか処理できない最も基本的な環境を示すロケールです。C ロケールでは、日本語の処理はできません。

現在のロケール設定は `locale` コマンドで表示することができます。

```
% locale
```

詳細については、`locale(1)` リファレンス・ページを参照してください。

2.4.2 Unicode および dense コード・ロケールの切替え

システムに Tru64 UNIX のワールドワイド言語サポートをインストールすると、2 種類のロケール (Unicode ロケールと dense コード・ロケール) がローカリゼーション・サポート用にインストールされます。

インストール直後のデフォルト設定では、dense コード・ロケールが使用されます。システムでどちらのロケールを設定するかは、ユーザが選択して設定することができます。Unicode ロケールと dense コード・ロケールの相違点および切替え方法についての詳細は、『国際化機能ユーザズ・ガイド』の第 1 章の「ロケールと言語の設定」の項を参照してください。

2.5 日本語端末の設定

端末への日本語の表示や、キーボードからの日本語入力を行うためには端末ドライバを日本語用のラインディシプリン（回線制御規則）に設定します。設定は次のように行います。

```
% stty jdec
```

このコマンドを実行した直後の端末ドライバの設定状態は、次のようになります。

項目	該当する stty のオプション	設定状態
端末入出力コード	tcode dec	DEC 漢字コード
アプリケーション・コード	acode コードセット名	ロケールに設定されている日本語コードセット (ロケールが設定されていない場合は eucJP)
日本語入力モード	imode hiragana	8-bit コードを2バイトのひらがなコードとして入力
漢字変換機能	-ikk	変換機能 OFF

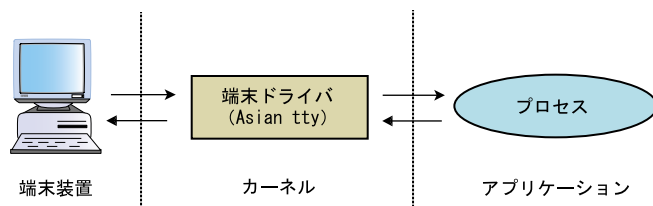
ユーザが使用している環境がこれらの設定とは異なる場合、必要に応じて **stty** コマンドを使用して各項目の再設定を行います。以下のセクションでは、端末コード、アプリケーションコード、8-bit データの取り扱いをユーザの環境に合わせて設定するための **stty** コマンドの使用法を解説します。

2.5.1 端末コードとアプリケーション・コードの設定

端末コードとアプリケーション・コードには、それぞれ独立に漢字コードを設定することができるので、様々な組合せが可能です。

Tru64 UNIX の端末装置を使用するアプリケーション・プログラムは、図 2-2 に示すように、端末装置とアプリケーションの間をカーネルに組み込まれた端末ドライバ **atty**¹ が仲介して通信しています。

図 2-2: 端末ドライバ (Asian tty)



Tru64 UNIX では、端末、端末ドライバ、アプリケーションの間で送受信される文字コードをユーザがそれぞれ選択することができます。以後、端末が

¹ 端末ドライバ **atty** は、Asian tty driver の略で、Tru64 UNIX で提供されるアジア圏の各国のロケールでの端末装置との通信や、漢字変換、ユーザ定義文字の読み込み / 表示などを行うために標準の端末ドライバを拡張したものです。

ホストと通信を行うときの文字コードを端末コード，アプリケーションが処理する文字コードをアプリケーション・コードと呼びます。

2.5.1.1 端末コードの設定

DEC 漢字コード以外の文字コードセットを端末コードとして設定するには `stty` コマンドの `tcode` オプションを使用します。

% `stty tcode` コードセット名

指定可能なコードセット名を表 2-5 に示します。

表 2-5: 端末コードとして指定できるコードセット名

コードセット名	対応するニーマニック
DEC 漢字	dec
シフト JIS	SJIS
日本語 EUC	eucJP
7-bit JIS	jis7
8-bit JIS	jis8
UTF-8	UTF-8

2.5.1.2 アプリケーション・コードの設定

アプリケーション・コードを設定するには `stty` コマンドの `acode` オプションを使用します。

% `stty acode` コードセット名

指定可能なコードセット名を表 2-6 に示します。

表 2-6: アプリケーション・コードとして指定できるコードセット名

コードセット名	対応するニーマニック
DEC 漢字	deckanji
Super DEC 漢字	sdeckanji
シフト JIS	SJIS
日本語 EUC	eucJP
UTF-8	UTF-8

2.5.2 8-bit データの取り扱い

端末から 8-bit データ，すなわち 8 ビット目がオンになっているデータが入力された場合，次のように解釈されます。

1. 1 バイトのカナ文字と解釈され
 - 1 バイトの半角カナ文字としてそのままアプリケーションに渡される。
 - そのカナ文字に対応する 2 バイトの全角ひらがなのコードに変換される。
 - そのカナ文字に対応する 2 バイトの全角カタカナにコードに変換される。
2. 複数バイトの漢字コードの一部として解釈される。

これらは `stty` コマンドの `imode` オプションによって指定します。

- 8-bit コード 1 バイトの半角カナ・コード
入力された 8-bit データを 1 バイトのカナ文字として，アプリケーションに渡すには次のように設定します。ただし 1 バイトのカナ文字をサポートしていない DEC 漢字コードがアプリケーションコードに設定されているときにかぎり，このコードが表す 1 バイトのカナ文字を対応する 2 バイトのカタカナのコードに変換します。

```
% stty imode hankaku
```
- 8-bit コード 2 バイトの全角ひらがなコード
入力された 1 バイトの 8-bit データが表すカナ文字を 2 バイトのひらがなコードへ変換するには，`stty` コマンドを使用して次のように指定します。

```
% stty imode hiragana
```
- 8-bit コード 2 バイトの全角カタカナ・コード
入力された 1 バイトの 8-bit データが表すカナ文字を 2 バイトのカタカナ・コードへ変換するには，`stty` コマンドを使用して次のように指定します。

```
% stty imode katakana
```

- 8-bit コード 複数バイト文字の一部

端末コードに DEC 漢字コードが設定されている場合，入力された 1 バイトの 8-bit データを漢字コードの一部として扱うためには，`stty` コマンドを使用して次のように指定します。

```
% stty imode kanji
```

2.5.3 端末の設定状態の確認

現在の端末ドライバの設定状態を表示するには，`stty(1)` コマンドをフラグやオプション与えずに実行します。

```
% stty
atty disc;speed 9600 baud; -parity hupcl
erase = ^?
brkint -inpck -istrip icrnl -ixany onlcr
echo echoe echok
imode=hiragana
tcode=dec acode=deckanji
%
```

表示された各項目は `stty(1)` で指定するオプションに対応しています。ハイフン (-) が前についているものは，現在無効 (disable) なオプション，そうでないものは現在有効 (enable) なオプションです。 `stty` コマンドのオプションについては，`stty(1)` リファレンス・ページを参照してください。

`stty(1)` に `-a` フラグを指定すると，`stty(1)` で設定できる全オプションの設定の状態が表示されます。

2.6 ファイル・コードの変換

Tru64 UNIX は，`iconv(1)` というファイル・コードを変換するコマンドを提供しています。 `iconv(1)` は次のように起動します。

```
iconv -f 入力コード -t 出力コード ファイル名 ...
```

`iconv(1)` は `-f` フラグに指定された「入力コード」から，`-t` フラグに指定された「出力コード」へ指定されたファイルを変換し，結果を標準出力に出力します。コードセット名は，表 2-7 に示されたものを指定することができ，ファイル名は，複数指定することができます。

表 2-7: `iconv` で指定可能なコードセット名

コードセット名	対応するニーマニック
DEC 漢字	<code>deckanji</code>
Super DEC 漢字	<code>sdeckanji</code>
日本語 EUC	<code>eucJP</code>
シフト JIS	<code>SJIS</code>
7-bit JIS	<code>JIS7</code>
7-bit JIS ^a	<code>jiskanji7</code>
ISO-2022-JP 日本語メール・コード	<code>ISO-2022-JP</code>
ISO-2022-JP 拡張日本語メール・コード	<code>ISO-2022-JPext</code>
UCS-4	<code>UCS-4</code>
UCS-2	<code>UCS-2</code>
UTF-8	<code>UTF-8</code>

^aJIS7 と `jiskanji7` では、JIS X0201 カタカナ文字セットの表現方法が異なります。詳細は `jiskanji(5)` を参照してください。

たとえばシフト JIS を使用したテキスト・ファイル `foobar.txt` を Super DEC 漢字のテキスト・ファイル `foobar.sdec` に変換するには、コマンド行上で次のように指定します。

```
% iconv -f SJIS -t sdeckanji foobar.txt >foobar.sdec
```

各コードセットのユーザ定義文字領域はそれぞれ異なります。`iconv` でコード変換を行うときには、ユーザ定義文字の文字コードに割当てられるコードの値も変換されますので注意が必要です。各コードセットのユーザ定義文字領域と、コードセット間での文字コードの対応関係は、`udc(5)` リファレンス・ページを参照してください。

表 2-7 に示すコードセットに加えて、他社製汎用機で使用されているコードセットへのファイル・コード変換をサポートしています。

表 2-8: iconv で指定可能なコードセット名 (他社汎用機)

コードセット名	対応するニーモニック
IBM 漢字システム文字	ibmkanji
富士通 JEF コード	JEF
日立 KEIS	KEIS

詳細については、`iconv_ibmkanji(5)`、`iconv_JEF(5)`、`iconv_KEIS(5)` の各リファレンス・ページを参照してください。

2.7 メッセージ・カタログ

プログラムは、ユーザとの対話のために実行中に様々なメッセージを出力します。従来のソフトウェアの多くは、メッセージ文字列がプログラム中に埋め込まれていたため、実行環境に合わせて出力内容を変えることはできませんでした。

Tru64 UNIX の国際化されたコマンドやライブラリは、メッセージ文字列をプログラムの中に埋め込むのではなく、メッセージ文字列を記録したファイルを実行時に参照し、その内容をメッセージとして出力します²。このように、実行時にファイルを参照して必要なメッセージを出力する機構や、そのファイルのことをメッセージ・カタログと呼びます。

Tru64 UNIX では、インストール時に日本語ロケール用のメッセージ・カタログをロケール名で指定される漢字コードごとに表 2-9 に示すディレクトリに格納します³。

表 2-9: 日本語メッセージ・カタログのあるディレクトリ

漢字コード	メッセージ・カタログが置かれるディレクトリ
日本語 EUC	<code>/usr/lib/nls/msg/ja_JP</code>
DEC 漢字	<code>/usr/lib/nls/msg/ja_JP.deckanji</code>
Super DEC 漢字	<code>/usr/lib/nls/msg/ja_JP.sdeckanji</code>

² メッセージが入っているファイルの内容は、`gencat(1)` コマンドによってバイナリ形式にコンパイルされているため、`cat` コマンドなどで表示することはできません。ファイルの内容を表示する場合は `dspcat(1)` コマンドを使用します。

³ コンポーネントによっては実行時にメッセージ・ファイルをコード変換するものもあります。このため、すべてのコードセットのディレクトリに各コンポーネントのメッセージ・ファイルが存在するとは限りません。

表 2-9: 日本語メッセージ・カタログのあるディレクトリ (続き)

漢字コード	メッセージ・カタログが置かれるディレクトリ
日本語 EUC	/usr/lib/nls/msg/ja_JP.eucJP
シフト JIS	/usr/lib/nls/msg/ja_JP.SJIS
UTF-8	/usr/lib/nls/msg/ja_JP.UTF-8

2.8 日本語リファレンス・ページの設定

Tru64 UNIX では、日本語に対応したコマンド、ライブラリについて日本語リファレンス・ページを提供しています。リファレンス・ページは、`man` コマンドでオンライン上で参照するドキュメントです。コマンド行から次のように `man` コマンドを実行すると、指定したタイトルのドキュメントが端末の画面上に出力されます。

```
% man 1 title
```

ここで第 1 引数の 1 は、リファレンス・ページのセクション番号です。リファレンス・ページは、カテゴリごとに表 2-10 のようなセクションに分類されています。

表 2-10: リファレンス・ページのセクション番号

内容の分類	対応するセクション番号
コマンド	1
システム・コール	2
ライブラリ・ルーチン	3
ファイル・フォーマット	4
nroff のマクロなど	5
その他	6
スペシャル・ファイル	7
システム管理	8

環境変数 `LANG` が日本語用に設定されている場合、`man(1)` コマンドは日本語リファレンス・ページのソース・ファイルが格納されているディレクトリ (

LANG が `ja_JP.deckanji` の場合は `/usr/share/ja_JP.deckanji/man` を最初に検索し、日本語リファレンス・ページを表示します。

ただし、環境変数 LANG が `ja_JP.SJIS` に設定されている場合は、`/usr/share/ja_JP.eucJP/man` ディレクトリを探索し、コードセット・コンバータによって SJIS フォーマットに変換されます。

Tru64 UNIX で提供される英語版リファレンス・ページを参照する必要があるときは、`man` コマンドの `-M` オプションを次のように使います。

```
% man -M /usr/share/man タイトル
```

`man` コマンドについての詳細は、リファレンス・ページ `man(1)` を参照してください。

日本語入力システムの設定

Tru64 UNIX では、ウィンドウ環境における標準の日本語入力システムとして日本語入力サーバ dxjim を提供しています。また、オプションの日本語入力システムとして、VJE-Delta および WX3 も提供しています。

3.1 dxjim

日本語入力サーバ dxjim は、X11R6 で採用された国際化入力方式である XIM に対応した日本語入力サーバです。

OSF/Motif V1.2.5 をベースにした Tru64 UNIX の日本語ウィンドウ環境では、XmText および DXmCSText ウィジェットで日本語入力ができます。

日本語入力を行うには、アプリケーションが表示されるディスプレイ・サーバ上にあらかじめ dxjim が起動されている必要があります。

本バージョンの dxjim は、従来の DECXim プロトコルに加えて、X11R6 で業界標準となった R6 XIM プロトコルをサポートしています。

3.1.1 dxjim の起動

セッション・マネージャや各アプリケーションで日本語入力を行うには、それらを表示しているディスプレイ上で日本語入力サーバが動作している必要があります。

1 つの日本語入力サーバが同一ディスプレイ上の複数のアプリケーションに対して日本語入力機能を提供します。この場合、リモート・ノードからネットワークを経由して同一ディスプレイに表示されているアプリケーションも日本語入力サーバによる日本語入力を受けることができます。

dxjim の起動は、CDE のログイン画面から日本語を選択してログインすることにより自動的に行われます。また、以下のコマンドで明示的に起動することもできます。

```
% /usr/bin/X11/dxjim &
```

3.1.2 dxjim による日本語入力の概要

dxjim が動作しているディスプレイでアプリケーションを起動し、文字入力領域にフォーカスをあてると、dxjim のタイトルが入力スタイルを表示するよう変化します。この状態で、日本語入力が可能です。dxjim は on-the-spot、over-the-spot、root-window の 3 つの入力スタイルをサポートしており、省略時のスタイルは over-the-spot です。dxjim の使用方法については第 7 章を参照してください。

3.2 VJE-Delta

VJE-Delta for Tru64 UNIX は、PC 上の FEP として実績のある VJE-Delta¹ のベース機能を X Window System 上の入力サーバとして Tru64 UNIX 上に実装したものです。Tru64 UNIX 上で動作するアプリケーションに、日本語入力変換サービスを提供するための IM (インプット・メソッド)です。

VJE-Delta のキットは、Tru64 UNIX のインストール・キットに含まれている「日本語入力システム VJE-Delta」CD-ROM で提供されています。インストールおよび使用方法については、VJE-Delta のユーザーズ・ガイドを参照してください。

3.3 WX3

WX3 for Tru64 UNIX は、PC 上の FEP として実績のある WX3を X Window System 上の日本語入力サーバとして日本語 Tru64 UNIX 上に実装したものです。

WX3 for Tru64 UNIX をご使用いただくためには、オペレーティング・システムとは別のライセンスが必要になります。

WX3 のキットは「日本語入力システム WX3」CD-ROM で提供されています。インストールおよび使用方法については、『WX3 for DIGITAL UNIX ユーザーズ・ガイド』を参照してください。

¹ VJE-Deltaは株式会社パックスの登録商標です。

日本語プリンタの設定

Tru64 UNIX では、従来から提供している `lpr(1)` コマンドによるプリント機能に加えて、Advanced Printing Software を利用したプリント機能を提供しています。この章では、4.1 節 ~ 4.4 節で `lpr` コマンドのためのプリンタ設定について説明し、4.5 節で Advanced Printing Software について説明します。

4.1 プリンタ・コード

国際化された環境では、様々な漢字コードを使用した日本語テキスト・ファイルがファイル・システム上に多数存在することになるでしょう。ユーザが使用しているプリンタ装置が、ファイルと異なるコードセットを使用している場合、プリンタ装置に合わせて適当に文字コードを変換しなければなりません。以後、本書ではファイルに使用されているコードセットをファイル・コード、プリンタ装置（シリアル・プリンタ装置）が処理することのできるコードセットをプリンタ・コードと呼びます。

ファイルの内容をプリンタで印字するには、`lpr(1)` コマンドを使用します。`lpr(1)` は、出力するファイルのファイル・コードとして現在のユーザのロケールの「文字コードセット」の部分を参照します。またプリンタ・コードとして、`/etc/printcap` の中から使用するプリンタ装置の `ya` というエントリを探し、そこで指定されているコードセットを参照します。ここでファイル・コードと、プリンタ・コードに異なるコードセットが使用されている場合、`iconv` によって出力するファイルの文字がプリンタ・コードに変換され、プリンタ・スプールと呼ばれるディレクトリにファイルとして出力されます。またプリンタ制御のための制御ファイルもプリンタ・スプールに書き込まれます。

次に `lpr(1)` は、プリンタ・スプール・デーモン `lpd(8)` にプリント要求を送ります。`lpd(8)` は、プリント要求を受けるとスプール・ディレクトリに置かれた出力ファイルを出力装置に応じたプリンタ・フィルタに通してからプリンタ装置へ出力します。

4.2 日本語プリンタの設定方法

日本語プリンタの設定は、`/usr/sbin/lprsetup` コマンド、あるいは Printer Configuration ユーティリティを使用して行います。これらの作業にはスーパーユーザ特権が必要です。

`lprsetup` コマンドの起動は、コマンド行から直接起動するか、あるいは SysMan Menu から「Configure line printer」を選択することにより行えます。

Printer Configuration ユーティリティを利用して日本語プリンタの設定を行う場合は、以下の手順に従ってください。

1. CDE のアプリケーション・マネージャから、「システム管理」「システム設定」「プリンタ」の順にアイコンを選択して、Printer Configuration ユーティリティを起動します。
2. 「Available Printer Types」の中から、設定するプリンタの名前をクリックします。日本語プリンタの名前には、最後に "ja" が付いています。

例：DEClaser2300

3. 「Configure...」ボタンをクリックします。
4. Printer Aliases (プリンタの別名)や、プリンタを接続するデバイス名などを 入力します。
5. 「Advanced...」ボタンをクリックすると、フラグページの抑制など、細かな設定を行うことができます。Printer ODL によるユーザ定義文字の印刷機能を使用する場合には、「Advanced...」ボタンをクリックして必要な設定を行ってください。

- a. Printer ODL 機能を設定するには、「SoftODL cache size (ys):」のフィールドを選択して、256 を入力します (256 以外の値に設定することも可能です)。
- b. 「Country-specific params (ya):」のフィールドを選択して、odlstyle の設定を追加します。

たとえば、ln05ja で、フォントサイズ 40x40 のフォントを使用する場合は、次のように設定します。

```
"plocale=ja_JP.sdeckanji odlstyle=normal-40x40"
```

- c. 「Advanced...」画面の入力が終了したら、「OK」ボタンをクリックします。
6. 「Commit」ボタンをクリックして設定内容を保存した後、「Close」ボタンをクリックします。
7. 以上で、プリンタの設定は終了です。設定内容は、`/etc/printcap` ファイルに書き込まれています。

例 4-1: `/etc/printcap` の例

```
lp0|0|ln05:\
:af=/usr/adm/lp0acct:\
:br#9600:\
:ct=dev:\
:fc#0177777:\
:fs#023:\
:if=/usr/lbin/ln05jaof:\
:lf=/usr/adm/lp0err:\
:lp=/dev/tty01:\
:mx#0:\
:of=/usr/lbin/ln05jaof:\
:pl#66:\
:pw#80:\
:sd=/usr/spool/lpd0:\
:xc#0177777:\
:xf=/usr/lbin/xf:\
:xs#044000:\
:ya="plocale=ja_JP.sdeckanji odlstyle=normal-40x40":\
:ys#256:
```

Printer Configuration ユーティリティは、System Setup ユーティリティの Custom Setup サブ・ユーティリティから「Printer Configuration」を選択することによっても起動できます。

Tru64 UNIX が提供するプリンタ・サポートの国際化機能については、`i18n_printing(5)` を参照してください。

4.3 ESC/P プリンタを利用するための設定

Tru64 UNIX では、`pcfof` フィルタ用の構成ファイル `escp_a4_10cpi` および `escp_a4_12cpi` が提供されており、ESC/P プリンタを利用することがで

きます。これらの構成ファイルは、ESC/P プリンタの初期化コードの設定と、10 または 12 CPI (characters per inch) 印刷のための設定を行います。

lprset コマンドで `escp_a4_10cpi` または `escp_a4_12cpi` を選択することにより、`printcap` ファイルに対して ESC/P プリンタの設定が自動的に行なわれます。

`/etc/printcap` を手作業で編集する場合は、`if` および `of` エントリを次のように設定してください。

```
:if=/usr/lbin/pcfof +Cescp_a4_10cpi -Opcode=escp:
:of=/usr/lbin/pcfof +Cescp_a4_10cpi -Opcode=escp:
```

さらに、次のように、`ya` エントリに対してプリンタ・ロケールとして `ja_JP.eucJP` を設定する必要があります。

```
:ya="plocale=ja_JP.eucJP":
```

4.4 ESC/Page プリンタを利用するための設定

「Tru64 UNIX V5.1B Operating System 日本語追加機能」CD-ROM には ESC/Page プリンタ用の汎用フィルタ `escpageof` が提供されており、このフィルタを使用することにより Tru64 UNIX で ESC/Page プリンタを利用することができます。

本フィルタは `lprset` コマンドには対応していませんので `/etc/printcap` を手作業で編集する必要があります。

システムに `escpageof` をインストールしたら、`/etc/printcap` のエントリを次のように設定してください。

```
:if=/usr/lbin/escpageof:
:of=/usr/lbin/escpageof:
```

`if` エントリにオプションを指定することにより、デフォルトの印刷結果を自由に設定することができます。使用可能なオプションは `w`、`l`、`i`、`K`、`I`、`O`、`N` です。

以下に例を示します。

```
:if=/usr/lbin/escpageof -N2 -K2:
```

この例の場合、デフォルトで印刷用紙の各面に 2 ページ分が印刷されます。

`escpageof` のインストレーション方法、設定方法、および使用方法については、「Tru64 UNIX V5.1B Operating System 日本語追加機能」CD-ROM

の `j_lprfilter` ディレクトリに含まれている README ファイルをを参照してください。

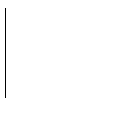
4.5 Advanced Printing Software のインストールと設定

Advanced Printing Software は、Xerox 社の PrintXchange 技術をベースに開発された分散クライアント/サーバ型のプリンティング・システムです。Advanced Printing Software は、インバウンドおよびアウトバウンド・ゲートウェイを使用して、従来から提供している lpr/lpd プリント・サブシステムとの間でプリント・ジョブの送受信を行います。

Advanced Printing Software の英語版キットおよび日本語版キットは「Associated Products Volume 1」CD-ROM に含まれています。Advanced Printing Software を利用して日本語プリンタにファイルを出力したい場合は、英語版および日本語版の両方のキットをシステムにインストールする必要があります。

Advanced Printing Software の日本語キットには、日本語プリンタに日本語テキスト・ファイルまたは日本語 PostScript ファイル (プリンタが日本語 PostScript 対応の場合) を出力するための環境が用意されています。

Advanced Printing Software のインストール、設定、および使用方法については、Advanced Printing Software の日本語ドキュメントを参照してください。Advanced Printing Software の日本語ドキュメントは「日本語追加機能」CD-ROM に含まれています。



日本語キーボードの設定

Tru64 UNIX システムでは、以下の日本語キーボードが使用できます。

キーボード	配列	変換キー	カナLED	説明
LK411-AJ/LK41W-AJ	ANSI	あり	あり	日本語LK411キーボード
LK411-JJ/LK41W-JJ	JIS	あり	あり	日本語LK411キーボード
LK421-AJ	JIS	なし	なし	日本語LK421キーボード
LK421-JJ	JIS	あり	あり	日本語LK421キーボード
PCXAJ-AA	JIS	あり	なし	日本語PC 106キーボード
LK97W-AJ	JIS	あり	なし	日本語PC 109キーボード

- LK41W-AJ, LK41W-JJ は、それぞれ LK411-AJ, LK411-JJ と色が異なるだけで機能的には同じものです。ご利用にあたっては対応するキーボードと同じ設定でご利用ください。
- LK97W-AJ は PCXAJ-AA に Windows 用の 3 つのキーが追加されたものです。ご利用にあたっては PCXAJ-AA と同じ設定でご利用ください。追加されたキーは本バージョンでは意味を持ちません。

キーボード名は、通常、キーボード裏面のラベルに記述されています。

5.1 CDE におけるキーボード設定

日本語キーボードを正しく使用するには、ウィンドウ環境に対するキーボード設定が必要です。CDE セッションにおけるキーボードの設定方法は次のとおりです。

ユーザごとのキーボード設定は「キーボード設定」アクションで行います。「キーボード設定」アクションをアプリケーション・マネージャの「デスク

トップ・アプリケーション」から起動し、キーボード・リストの中からご使用の日本語キーボード名を選択してください¹。

「キーボード設定」に関する詳細は『CDE ガイドブック』を参照してください。

システムの省略時のキーボード設定については 5.2 節 を参照してください。

注意

日本語キーボード LK41W-AJ, LK41W-JJ, LK97W-AJ はそれぞれ LK411-AJ, LK411-JJ, PCXAJ-AA と同じ設定でご使用ください。

5.1.1 カナ・モード

日本語キーボードの設定後、カナ・モードをオンにすることによって日本語キーボードからカナを直接入力できるようになります。カナ・モードでは、キーボードをタイプするとキーに刻印されている ASCII 文字ではなく同じキーに刻印されているカナ文字が入力されます。

カナ・モードをオンにするキーは、PCXAJ-AA および LK97W-AJ では右側の **Ctrl**²、他の日本語キーボードでは **カナ** です。 **カナ** キーはキーボードの左下にあります。同じ操作をもう一度行くと、ASCII モードに戻ります。

なお、日本語環境以外ではカナ文字は正しく入力されません。

5.1.2 変換キー

いくつかの日本語キーボードには、スペース・キーの両側に **無変換**、**変換 / 前候補**、**ひらがな / カタカナ** というキーがあります。これらのキーは変換キーと呼ばれ、日本語入力サーバからのかな漢字変換入力に使用することができます。

¹ キーボード名に esc がついているものを指定すると、左上の ' のキーがエスケープ・キーとして使用できます。この時、" ' " と " ` " は左下の < のキーに移動します。

² PCXAJ-AA および LK97W-AJ には、**Ctrl** キーが2つあります。

5.1.3 カナ LED

カナ LED は、カナ・モードの状態を表わす日本語キーボード上の LED です。カナ・モードがオンの時この LED が点灯します。

Tru64 UNIX の従来のバージョンでは、キーボード・インディケータ (kb_indicator) がカナ LED を制御していました。V4.0 以降のバージョンでは、XKB のサポートによって kb_indicator 無しでもカナ LED が正しく点滅します。

5.2 コンソール・モードおよびシングルユーザ・モードにおける日本語キーボードの設定

V4.0D 以降、日本語キーボードの設定が変更され、使い勝手が向上しています。

従来のバージョンでは、日本語キーボードに関して以下のような問題がありました。

- コンソール・ファームウェア上で JIS 配列日本語キーボードを使用すると、キーの刻印と違った文字が入力されることがありました。
- OS のシングルユーザ・モードで JIS 配列日本語キーボードを使用すると、キーの刻印と違った文字が入力されることがありました。
- 英語キーボード使用時には省略時のキーボード設定は不要なのに、日本語キーボード使用時には必要でした。

コンソール・ファームウェアが変更され、これらの問題は、以下のようにすべてが解決されています。

- 最新のコンソール・ファームウェアをシステムにインストールして環境設定することにより、コンソール・ファームウェア、OS のシングルユーザ・モード、およびウィンドウ環境のすべてのモードで、日本語キーボードが正しく設定され、刻印どおりの文字が入力されます。
- ウィンドウ環境の省略時のキーボード設定は不要です。

最新のコンソール・ファームウェア CD-ROM は、本バージョンのインストール・キットに含まれています。

コンソール・ファームウェアによる日本語キーボードの設定方法は次のとおりです。

- 1. コンソール・ファームウェアをアップデートします。
詳しくはコンソール・ファームウェアのドキュメントを参照してください。
- 2. コンソール・ファームウェア変数 `kbd_hardware_type` および `language` を設定します。

変数 `kbd_hardware_type` はキーボードのタイプを表します。LK41* 日本語キーボードの場合は LK411 を , PCXAJ-AA および LK97W-AJ の場合は PCXAL を指定します。

変数 `language` はキーボードの配列と言語を表します。JIS 配列の日本語キーボードの場合は 50 を , ANSI 配列の日本語キーボードの場合は 52 を指定します。

キーボードごとの変数値は以下のようになります。

キーボード	<code>kbd_hardware_type</code>	<code>language</code>
LK411-AJ/LK41W-AJ	LK411	52
LK411-JJ/LK41W-JJ	LK411	50
PCXAJ-AA	PCXAL	50
LK97W-AJ	PCXAL	50

以下は LK41W-JJ キーボードの設定例です。

```
>>> set language 50
>>> set kbd_hardware_type LK411
```

次の例のように , 1 行に複数のコマンドを入力することもできます。

```
>>> set language 50; set kbd_hardware_type LK411
```

- 3. コンソール・ファームウェアを初期化します。

```
>>> init
```

あるいは , システムの電源を入れ直します。

キーボードの交換が必要な場合はシステムの電源を切ってから行います。

- 4. `show` コマンドで変数の値が変更されていることを確認します。

```
>>> show
:
kbd_hardware_type      LK411
```

language	50
language_name	Japanese (JIS)
:	



日本語ドキュメント全文検索機能のインストールおよび設定

Tru64 UNIX では、日本語ドキュメントおよびリファレンス・ページの全文検索機能として、MitakeSearch for Online Documents (midoc) と MitakeSearch for Man Pages (miman) を提供しています。これらのツールは、コンパックコンピュータが開発した日本語全文検索システム MitakeSearch (ミタケサーチ) を利用しており、任意の文字列でオンライン・ドキュメントおよび man page の全文検索を行うことができます。

- midoc の検索の対象となるオンライン・ドキュメントは日本語 HTML ドキュメントです。midoc には、Web ブラウザを利用した GUI インタフェースが用意されています。
- miman の検索対象は日本語 man page および英語版 man page です。既存の apropos コマンドでは、whatis データベースに登録されたキーワードでの検索しかできませんが、miman を利用することにより man page の全文検索が可能です。miman には、コマンドライン・インタフェースと Web ブラウザを利用した GUI インタフェースが用意されています。

midoc および miman は、CGI 経由で Web ブラウザによる検索を前提としているため、使用するためには、事前にシステムへのインストールと HTTP サーバに対する設定が必要になります。この章では、midoc および miman のインストールと必要な設定について説明します。

なお、midoc および miman のキットは「日本語追加機能」CD-ROM に含まれています。

miman および midoc の GUI インタフェースの使い方については、それぞれの検索画面のヘルプ・アイコンあるいはヘルプ・ボタンをクリックしてオンライン・ヘルプを参照してください。

miman のコマンドライン・インタフェースの使い方については、リファレンス・ページを参照してください。

日本語全文検索システム MitakeSearch の概要については、下記の Web ページを参照してください。

<http://www.compaq.co.jp/products/mitake/>

6.1 midoc のインストール

MitakeSearch for Online Documents (midoc) のインストールは、`setld` コマンドを使用して行います。キットの展開先は、`/var/opt/midoc` ディレクトリに固定されており、およそ 27 MB の空き容量を必要とします。

以下の項で、midoc のインストール手順を説明します。

6.1.1 キットの展開先の確保

`/var/opt` ディレクトリに十分な空き容量 (27 MB 以上) があることを確認してください。もし空き容量が十分でない場合には、別のディスクまたはパーティション上に midoc 用のディレクトリを用意し、`/var/opt/midoc` からシンボリックリンクを張ってください。

たとえば `/usr/local` ディレクトリに十分な空きがあるため `/usr/local/midoc` にキットを展開したい場合は、次のように行います。この操作は root ユーザとして実行してください。

1. `/usr/local/midoc` ディレクトリを作成します。

```
% mkdir -p /usr/local/midoc
```
2. `/usr/local/midoc` ディレクトリの所有者を root に設定します。

```
% chown root /usr/local/midoc
```
3. `/usr/local/midoc` ディレクトリの許可モードを 755 に設定します。

```
% chmod 755 /usr/local/midoc
```
4. `/var/opt/midoc` ディレクトリを `/usr/local/midoc` ディレクトリにリンクさせます。

```
% ln -s /usr/local/midoc /var/opt/midoc
```

なお `/var/opt` ディレクトリに十分な空き容量があり、`/var/opt/midoc` に直接展開する場合には、この手続きは不要です。

6.1.2 キットからのインストール

midoc のキットは、IOSMIDOC540 サブセットとして 1 つにまとめられています。「日本語追加機能」CD-ROM を /mnt ディレクトリにマウントしている場合、以下のコマンドで midoc をインストールできます。

```
% setld -l /mnt/midoc/kit IOSMIDOC540
```

setld コマンドの使用方法については、『インストール・ガイド』および setld(8) リファレンス・ページを参照してください。

6.1.3 インストール後の設定作業

midoc は、CGI 経由で、Web ブラウザによる検索を前提としています。展開後の作業として、midoc に関する設定のほか、HTTP サーバに対する設定が必要です。

6.1.3.1 アイコン・ファイルのコピー

HTTP サーバの icons ディレクトリ (<http://host-name/icons/> でアクセスされるディレクトリ) に、/var/opt/midoc/cgi-bin ディレクトリに格納されている次のファイルをコピーしてください。

- midoc-h.gif
- midoc-next.gif
- midoc-prev.gif
- midoc-t.gif
- midoc-help.gif
- midoc-p.gif
- midoc-search.gif
- midoc-x.gif

icons ディレクトリが存在しない場合は作成してください。ファイルをコピーしたら、ファイル・プロテクションを確認して、一般ユーザがアクセスできるように設定してください。

6.1.3.2 Perl スクリプトのコピー

HTTP サーバの CGI 向けのディレクトリ (`http://host-name/cgi-bin/` 等でアクセスされるディレクトリ) に、`/var/opt/midoc/cgi-bin` ディレクトリに格納されている以下の Perl スクリプトをコピーしてください。スクリプトをコピーしたら、一般ユーザが実行可能になるようにファイル・プロテクションを設定してください。

- `midoc.pl`
- `midoc2html.pl`
- `libjconv.pl`

なお、ご使用のシステムで `MitakeSearch` や `miman` を使用している場合は、既に `libjconv.pl` が存在する場合があります。これらは同じものですので、共通に使用できます。

6.1.3.3 マップ・ファイルの編集

マップ・ファイル (`/var/opt/midoc/etc/urlmap.cgi`) を編集して、検索結果の URL が表示可能な URL にマップされるようにします。`midoc` 用データベースに登録されたドキュメントの URL は、「日本語追加機能」CD-ROM を、`/usr/share/doclib/online` にマウントした状態と同じです。ただし、ホスト名が `localhost` での `file://` プロトコルで登録されています。したがって、全てのドキュメントは先頭に `file://localhost/usr/share/doclib/online/` を含む URL で登録されています。

以下の例を参考に、URL マップの編集を行ってください。なお、区切り文字は `[TAB]` 1 つとし、ブランク等はいれないでください。

- 例 1 ローカルにマウントされた **CD-ROM** をアクセスする場合 -

たとえば「日本語追加機能」CD-ROM をローカル・システムの `/mnt` ディレクトリにマウントし、検索実行者がローカルファイルとしてアクセスする場合は、次のように編集します。

[編集前]

```
file://localhost/usr/share/doclib/online/[TAB]http://hostname/path/
```

[編集後]

```
file://localhost/usr/share/doclib/online/[TAB]file:/mnt/
```


- 例 2 **Web** サーバにマウントされた **CD-ROM** をアクセスする場合 -

「日本語追加機能」CD-ROM が Web サーバにマウントされており、
http://hostname/document/ の URL でアクセスできる場合は、次のよう
に編集します。

[編集前]

```
file://localhost/usr/share/doclib/online/ TAB http://hostname/path/
```

[編集後]

```
file://localhost/usr/share/doclib/online/ TAB http://hostname/document/
```

なお、CD-ROM 上のファイルを同じディレクトリ構成でディスクへコピー
している場合も、上記と同様にマップ・ファイルを編集することにより、
midoc をご利用いただけます。

6.1.4 検索の実行

6.1.3.3 項 で設定した HTTP サーバに、以下のような URL でアクセスす
ると検索用 CGI が実行されます。

```
http://host-name/cgi-bin/midoc.pl
```

検索結果の画面から URL をクリックしても該当するドキュメントが参照でき
ない場合には、URL マップ・ファイル(/var/opt/midoc/etc/urlmap.cgi)
の設定を見直してください。

6.1.5 midoc のアンインストール

アンインストールは、setld コマンドで行ないます。

```
% setld -d IOSMIDOC540
```

6.2 miman のインストール

MitakeSearch for Man Pages (miman) のインストールは、setld コマ
ンドで行います。miman の展開先は、/var/opt/miman ディレクトリに固
定されており、およそ 105 MB の空き容量を必要とします。

以下の項で、miman のインストール手順を説明します。

6.2.1 キットの展開先の確保

/var/opt ディレクトリに十分な空き容量 (105 MB 以上) があることを確認してください。もし空き容量が十分でない場合には、別のディスクまたはパーティション上に miman 用のディレクトリを確保し、/var/opt/miman とシンボリックリンクを張ってください。

たとえば /usr/local ディレクトリに十分な空きがあるため /usr/local/miman にキットを展開したい場合は、次のように行います。この操作は root ユーザとして実行してください。

1. /usr/local/miman ディレクトリを作成します。

```
% mkdir -p /usr/local/miman
```
2. /usr/local/miman ディレクトリの所有者を root に設定します。

```
% chown root /usr/local/miman
```
3. /usr/local/miman ディレクトリの許可モードを 755 に設定します。

```
% chmod 755 /usr/local/miman
```
4. /var/opt/miman ディレクトリを /usr/local/miman ディレクトリにリンクします。

```
% ln -s /usr/local/miman /var/opt/miman
```

なお /var/opt ディレクトリに十分な空き容量があり、直接 /var/opt/miman に展開する場合には、この手続きは不要です。

6.2.2 キットからのインストール

miman のサブセットは、IOSMIMAN540 の 1 つにまとめられています。CD-ROM を /mnt ディレクトリにマウントしている場合、次のコマンドでインストールできます。

```
% setld -l /mnt/miman/kit IOSMIMAN540
```

setld コマンドの使用方法については、『インストール・ガイド』および setld(8) リファレンス・ページを参照してください。

6.2.3 キット展開後の作業

miman には、次の 2 種類のユーザ・インタフェースが用意されています

- console からのコマンド形式で利用するためのコマンド行インタフェース

- CGI 経由で、Web ブラウザによる検索を行うグラフィカル・ユーザ・インタフェース

検索に利用するデータベースは、予め用意されたものがキットのインストール時にインストールされます。データベースの内容を変更 (追加/削除) する方法については、`mimanctl(1)` リファレンス・ページを参照してください。

6.2.3.1 コマンド形式で利用するための設定

コマンド形式での使用に関しては、キット展開後、直ちに使用することが可能です。使用方法については、`miman(1)` リファレンス・ページを参照してください。

6.2.3.2 CGI 経由で利用するための設定

検索 CGI クライアントおよびスケジュール管理 CGI クライアントを利用するためには、MitakeSearch for Man Pages のインストール終了後に以下の作業を行う必要があります。

1. HTTPサーバの `icons` ディレクトリ (`http://host-name/icons/` でアクセスされるディレクトリ) に、`/var/opt/miman/cgi-bin` ディレクトリに格納されている次のファイルをコピーしてください。ファイルをコピーしたら、一般ユーザがそれらのファイルをアクセスできるように、ファイル・プロテクションを設定してください。
 - `miman-h.gif`
 - `miman-go.gif`
 - `miman-help.gif`
 - `miman-m.jpg`
 - `miman-p.gif`
 - `miman-x.gif`
2. HTTP サーバ (フル・キットをインストールしたマシン) の CGI 向けのディレクトリ (`http://host-name/cgi-bin/` 等でアクセスされるディレクトリ) に、`/var/opt/miman/cgi-bin` ディレクトリに格納されている Perl スクリプト (`miman.pl`、`miman2html.pl` および `libjconv.pl`) をコピーしてください。スクリプトをコピーしたら、一般ユーザが実行可能になるようにファイル・プロテクションを設定してください。

検索 CGI クライアントは、WWW ブラウザで次のように URL 指定することにより利用できます。

```
http://host-name/cgi-bin/miman.pl
```

6.2.4 miman のアンインストールとその他の注意事項

- アンインストールは、次のように `setld` コマンドで行ないます。なお、ユーザが作成したファイルや、一部のディレクトリが残ることがあります。それらについては、手作業で削除してください。

```
% setld -d IOSMIMAN540
```

- データベースの更新中にエラーが発生すると、稀にデータベースが壊れることがあります。この場合は、キットを再度インストールしてください。再インストールの前には、必ずアンインストールを行なってください。なお、再インストールを行うと、ユーザ登録のデータは失われます。

dxjim による日本語入力

この章では、Tru64 UNIX で提供する標準の日本語入力機能である dxjim 日本語入力サーバについて説明します。

7.1 dxjim による日本語変換機能の概要

Tru64 UNIX 上のウィンドウ環境では、テキスト入力フィールドに日本語を入力することができます。日本語の入力は、ローマ字あるいは、かなで“読み”を入力してから“変換キー”を使用して漢字、ひらがな、カタカナなどに変換します。日本語入力の変換機能を表 7-1 に示します。

表 7-1: 日本語入力の変換機能

機能	説明
漢字変換， 文節次候補	漢字に変換します。 続けてもう一度押すと、次の候補が表示されます。
ひらがな変換， 文節ひらがな変換	ひらがなに変換します。
カタカナ変換， 文節カタカナ変換	カタカナに変換します。
全角変換	英数記号を全角に変換します。
半角変換	英数記号を半角に変換します。
記号変換， コード入力	1 ~ 3 文字の英数記号を特殊な記号に変換します (7.4.3 項を参照)。あるいは、漢字コードから漢字に変換します (7.4.4 項を参照)。
大文字変換	英字を大文字に変換します。
小文字変換	英字を小文字に変換します。
次文節移動	漢字変換した文字列に複数の文節がある場合に、次の文節に移動します (次の文節を変換対象とする)。
前文節移動	漢字変換した文字列に複数の文節がある場合に、前の文節に移動します (前の文節を変換対象とする)。
文節縮小	変換対象である文節を縮小します。

表 7-1: 日本語入力の変換機能 (続き)

機能	説明
文節拡大	変換対象である文節を拡大します。
文節前候補	一つ前の漢字変換の候補を表示します。
全文確定	入力した読み全体の変換を確定します。
文節確定	現在の変換対象である文節の変換を確定します (文節が複数ある場合は、次の文節が変換対象となります)。
変換取消， 無変換	読みを変換した直後は、変換開始前の読みを入力した状態にもどします (読みを修正することができます)。 読みを入力した直後は、入力した文字のまま確定します。

7.2 dxjim の使用方法

dxjim が動作しているディスプレイでアプリケーションを起動し、文字入力領域にフォーカスをあてると、dxjim のタイトルが入力スタイルを表示するよう変化します。この状態で、日本語入力が可能です。dxjim は on-the-spot、over-the-spot、root-window の 3 つの入力スタイルをサポートしており、省略時のスタイルは over-the-spot です。

over-the-spot のスタイルでは、日本語変換はアプリケーションのウィンドウの上で行われますので、特に dxjim を意識せずに日本語入力を行うことができます。ただし、変換中の文字列は、アプリケーションのウィンドウの上に表示されるので、既に存在する文字列の途中に文字を挿入する場合は変換が確定するまで、既存の文字列の一部が被り隠されることがあります。また、文字入力直後に「了解」ボタン等を押して処理を進めると、変換が確定していない入力文字については取り込まれません。このような場合は、変換を確定 (JVMS 変換キーの場合 `Ctrl/G`/`Ctrl/N`) してから次の処理に進む必要があります。

on-the-spot スタイルでは、日本語変換はアプリケーションのウィンドウ内で行われ、特に dxjim を意識せずに日本語入力を行うことができます。既に存在する文字列の途中に文字を入力する場合でも、既存の文字列の一部が被り隠されることはありません。

root-window スタイルでは、アプリケーションのウィンドウに入力された文字列は、dxjim のウィンドウ内に表示されます。日本語変換が確定する

までは、変換中の文字列はすべて dxjim のウィンドウ内に表示されますので注意が必要です。

入力スタイルの設定方法については 7.2.2 項 を参照してください。

7.2.1 dxjim の設定

dxjim の左下の領域には「漢字変換」あるいは「ASCII」のいずれかの文字列が表示されます。「漢字変換」が表示されている場合、アプリケーションは dxjim を使用して日本語を入力することができます。「ASCII」が表示されている場合は、日本語入力できません。

「ASCII」の状態から「漢字変換」の状態に変更するためのキーを始動キーと呼びます。逆に「漢字変換」から「ASCII」の状態に変更するためのキーを終了キーと呼びます。

始動キーおよび終了キーの標準値は、Shift-Space (**Shift** キーを押しながらスペース・バーを押す) です。アプリケーション起動時には、dxjim の状態は「漢字変換」になっています。始動キーおよび終了キーの設定の変更は「起動キー：設定」ダイアログボックスで行います。「起動キー：設定」ダイアログボックスは、「オプション」メニューから「キー設定」メニューを選択することにより表示されます。「起動キー：設定」ダイアログボックスで行った始動キーと終了キーの変更は、すべてのアプリケーションに対して有効です。

dxjim は、ユーザ・キー定義ライブラリ IMLIB を使用しています。ユーザ・キー定義は、DEC日本語入力サーバ を起動したユーザのキー定義が使用されます。IMLIB を使用した日本語入力の変換キー、ユーザ・キー定義については『ユーザ・キー定義 利用者の手引き』を参照してください。

「オプション」メニューから「キー定義」を選択することにより、変換キーを変更することができます。これにより変更できるのは変換キーの定義のみで、エコーの仕方の変更はできません。変換キーの変更は、変更時に日本語を入力していたウィジェットにのみ適用されます。

アプリケーションの入力スタイルが root-window の場合、dxjim のウィンドウ内にプリエディット用の領域が作成されます。プリエディットを確定しアプリケーションに入力を行うためのキーを確定キーと呼びます。確定キーの標準値は **Return** キーです。

確定キーの変更は「起動キー：設定」ダイアログボックスで行います。変更した確定キーは、root-window 入力スタイルを使用するすべてのアプリケーションに適用されます。

7.2.2 入力スタイル

X11R6.3 ではテキストウィンドウへの入力スタイルとして、on-the-spot、over-the-spot、off-the-spot、root-window を挙げています。dxjim では、on-the-spot、over-the-spot、root-window をサポートしています。また、X11R6.3 ではプリエディットの状態表示としていくつか挙げていますが、dxjim ではいずれもサポートしていません。dxjim では、接続しているクライアントの状態を表示する機能を持っています。

XmText および DXmCSText ウィジェットでは、VendorShell ウィジェットの XmNpreeditType リソースを使用して入力スタイルの設定を行うことができます。XmNpreeditType リソースには次のいずれかの値を指定します。

- OnTheSpot
- OverTheSpot
- Root
- OffTheSpot

また、XmNpreeditType リソースには複数の値を設定することも可能です。複数の値を設定する場合には、それぞれの値をカンマ "," で区切ってください。

省略時の入力スタイルは OverTheSpot です。OverTheSpot の場合、入力交換中の文字列はすでに存在しているテキストの上に上書きされ、変換が確定すると全体のテキストへ挿入されます。

7.2.3 リソース

リソースを使用して、dxjim を好みの設定に変更することができます。dxjim に対して有効なリソースは以下のとおりです。なお dxjim のクラス名は DXjim、省略時のリソースファイルは /usr/i18n/lib/X11/app-defaults/DXjim です。

- x

dxjim ウィンドウの左上の位置の x 座標を指定します。

設定値: 整数値で設定。左隅からの pixel 数を表す。

初期値: 画面左下

- **y**

dxjim ウィンドウの左上の位置の y 座標を指定します。

設定値: 整数値で設定。上隅からの pixel 数を表す。

初期値: 画面左下

- **startupKanjiMode**

テキスト・ウィジェット起動時に日本語入力が可能かどうかを設定します。on の場合は、テキスト・ウィジェットの起動直後からテキスト・ウィジェットへの日本語入力が可能になります。

設定値: on または off

初期値: on

- **startKey**

始動キーを設定します。始動キーの設定は、「オプション」メニューから「キー設定...」メニュー項目を選択することにより、「キー設定」ダイアログボックス内で設定することもできます。

設定値: Translation の event の記述と同じシンタックス。

初期値: Shif<Key>space

- **endKey**

終了キーを設定します。終了キーの設定は、「オプション」メニューから「キー設定...」メニュー項目を選択することにより、「キー設定」ダイアログボックス内で設定することもできます。

設定値: Translation の event の記述と同じシンタックス。

初期値: Shif<Key>space

- **root_commitKey**

確定キーを設定します。確定キーの設定は、「オプション」メニューから「キー設定...」メニュー項目を選択し、「キー設定」ダイアログボックス内で設定することもできます。

設定値: Translation の event の記述と同じシンタックス。
初期値: <Key>Return

- **kp_numlock**

値が False の場合、キーパッドのキーはプリエディット・ストリングの入力には使用できません。プリエディット中にキーパッドのキーを入力するとプリエディットは確定します。True の場合は、キーパッド・キーをプリエディット・ストリングの入力に使用することができます。ただし、PF1 ~ PF4 キーおよび Enter キーは使用できません。

設定値: True または False
初期値: False

- **icon_state**

値が Normal の場合、dxjim ウィンドウをアイコン化してプリエディットを開始すると dxjim ウィンドウは復元されます。Iconic の場合は、dxjim ウィンドウが勝手に復元されることはありません。Withdrawn の場合は、dxjim ウィンドウは表示されません。この時、ルート・プリエディット・スタイルは使用できません。また、dxjim の起動時にアイコンで起動したい場合は、以下のようにリソースの設定を行ってください。

```
DXjim*initialState: IconicState
```

設定値: Iconic, Withdrawn, Normal
初期値: Normal

- **use_root_resources**

ルート・プリエディット・スタイル用のリソース root_fontset, root_background, root_foreground, root_highlight, root_highlight_foreground を使用するかどうかを決定します。False の場合、クライアントの XSetICValues で設定された値が使用されます。

設定値: True または False

初期値: True

- **root_background**

use_root_resource リソースが True の時に、ルート・プリエディット・スタイルのプリエディット領域で使用されるバックグラウンド・カラーを指定します。

型: Pixel

初期値: white

- **root_foreground**

use_root_resource リソースが True の時に、ルート・プリエディット・スタイルのプリエディット・ストリング領域で使用されるフォアグラウンド・カラーを指定します。

設定値: カラーの設定。Named Color も使用できます。

初期値: black

- **root_highlight**

use_root_resource リソースが True の時に、ルート・プリエディット・スタイルのプリエディット・ストリング領域で使用される変換候補文字列のバックグラウンド・カラーを指定します。

設定値: カラーの設定。Named Color も使用できます。

初期値: black

- **root_highlight_fore**

use_root_resource リソースが True の時に、ルート・プリエディット・スタイルのプリエディット・ストリング領域で使用される変換候補文字列のフォアグラウンド・カラーを指定します。

設定値: カラーの設定。Named Color も使用できます。

初期値: white

- **root_fontset**

use_root_resource リソースが True の時に、ルート・プリエディット・スタイルのプリエディット・ストリングに使用されるフォントセットを指定します。

設定値: フォントセットの設定。

初期値: 下記のとおりです。

```
-jdecw-gothic-medium-r-normal---120-*-*m--jisx0208.1983-1,  
-jdecw-gothic-medium-r-normal---120-*-*m--jisx0201.1976-0,  
-jdecw-kmenu-medium-r-normal---120-*-*--iso8859-1
```

- root_width

ルート・プリエディット・スタイルのプリエディット領域の幅を指定します。

設定値: 整数値での設定。pixel 数を表す。

初期値: 動的に決定される。

7.3 日本語入力の変換キー

テキストを入力する際に使用する変換キーには、次の4つの変換キー定義があります。このうちのいずれか1つを使用することができます。特に何も指定しない限り、変換キー定義はJVMSとなります。変換キー定義の変更方法に関しては7.3.1項を参照してください。

- JVMS キー (省略時設定)

日本語VMS標準の変換キー定義です。変換キーは表7-2の通りです。

- EVEJ キー

EVEJ エディタの変換キー定義です。日本語EVEのEVEJキーパッド・モードで使われています。変換キーは表7-3の通りです。

- KEYPAD キー

VTシリーズ端末の数字キーパッドを使用したキー定義です。変換キーは表7-4の通りです。

- TARO キー

ワープロ・ソフトの一太郎¹の変換キー定義に似た変換キー定義です。
変換キーは表 7-5 の通りです。

- CTRL キー

コントロール・キーを使用したキー定義です。変換キーは表 7-6 の通りです。

表 7-2: JVMS 変換キー定義

機能	キー操作 ^a
漢字変換/文節次候補	Ctrl+スペース
漢字候補一覧 ^b	Ctrl+'
ひらがな変換/文節ひらがな変換	Ctrl+L
カタカナ変換/文節カタカナ変換	Ctrl+K
半角カナ変換	Ctrl+G Ctrl+K
全角変換	Ctrl+F
半角変換	Ctrl+G Ctrl+F
記号変換/コード入力	Ctrl+]
大文字変換	
小文字変換	
次文節移動	Ctrl+P
前文節移動	Ctrl+G Ctrl+P
文節縮小	Ctrl+/
文節拡大	Ctrl+G Ctrl+/
文節前候補	Ctrl+G Ctrl+スペース または Ctrl+G Ctrl+L
全文確定	Ctrl+G Ctrl+N
文節確定	
変換取消/無変換	Ctrl+N

^a Ctrl+スペース は Ctrl キーを押しながら スペース キーを押すことを示します。 Ctrl+G Ctrl+K は Ctrl+G を押した後に Ctrl+K を押すことを示します。

^b ' はキーボードの左上のキーです。

¹ 一太郎は株式会社ジャストシステムの登録商標です。

表 7-3: EVEJ 変換キ一定義

機能	キー操作 ^a
漢字変換/文節次候補	Ctrl+スペース
漢字候補一覧 ^b	Ctrl+'
ひらがな変換/文節ひらがな変換	Ctrl+H
カタカナ変換/文節カタカナ変換	Ctrl+K
半角カナ変換	PF1 Ctrl+K
全角変換	Ctrl+F
半角変換	Ctrl+E
記号変換/コード入力	PF1 ZあるいはPF1 z
大文字変換	
小文字変換	
次文節移動	Ctrl+P
前文節移動	PF1 Ctrl+P
文節縮小	Ctrl+A
文節拡大	PF1 Ctrl+A
文節前候補	PF1 Ctrl+スペース
全文確定	PF1 Ctrl+N
文節確定	
変換取消/無変換	Ctrl+N

^aCtrl+スペースはCtrlキーを押しながらスペースキーを押すことを示します。PF1|ZはPF1を押した後にZを押すことを示します。

^b'はキーボードの左上のキーです。

表 7-4: KEYPAD 変換キ一定義

機能	キー操作 ^a
漢字変換/文節次候補	KP7
漢字候補一覧 ^b	Ctrl+'
ひらがな変換/文節ひらがな変換	KP1
カタカナ変換/文節カタカナ変換	KP2
半角カナ変換	KP6
全角変換	KP3
半角変換	KP9
記号変換/コード入力	KP8
大文字変換	
小文字変換	
次文節移動	KP5
前文節移動	PF1 KP5
文節縮小	KP4
文節拡大	PF1 KP4
文節前候補	PF1 KP7
全文確定	
文節確定	
変換取消/無変換	KP0

^a KP0 ~ KP9 は数字キーパッドの 0 ~ 9 を表します。PF1|KP6 は PF1 を押した後に KP6 を押すことを示します。

^b ' はキーボードの左上のキーです。

表 7-5: TARO 変換キー定義

機能	キー操作 ^a
漢字変換/文節次候補	スペース
漢字候補一覧 ^b	Ctrl+'
ひらがな変換/文節ひらがな変換	F11
カタカナ変換/文節カタカナ変換	F12
全角変換 ^c	F14
半角変換 ^c	F13
記号変換/コード入力	Ctrl+]
大文字変換	
小文字変換	
次文節移動	
前文節移動	
文節縮小	
文節拡大	
文節前候補	
全文確定	Return
文節確定	
変換取消 ^d	DELETE

^aCtrl+]はCtrlキーを押しながら]キーを押すことを示します。

^b'はキーボードの左上のキーです。

^c漢字 / カタカナ / ひらがな変換を行った後に全角 / 半角変換を行うことはできません。この場合は、変換取消を行った後に全角 / 半角変換を行ってください。

^dTARO変換キー定義には無変換キーはありません。

表 7-6: CTRL 変換キー定義

機能	キー操作 ^a
漢字変換/文節次候補	Ctrl+K
漢字候補一覧 ^b	Ctrl+'
ひらがな変換/文節ひらがな変換	Ctrl+G
カタカナ変換/文節カタカナ変換	Ctrl+N
半角カナ変換	Ctrl+H
全角変換	Ctrl+E
半角変換	Ctrl+T
記号変換/コード入力	Ctrl+F
大文字変換	
小文字変換	
次文節移動	Ctrl+P r
前文節移動	Ctrl+P l
文節縮小	Ctrl+A
文節拡大	Ctrl+X
文節前候補	Ctrl+L
全文確定	
文節確定	
変換取消/無変換	Ctrl+B

^aCtrl+K は Ctrl キーを押しながら K キーを押すことを示します。

^b はキーボードの左上のキーです。

7.3.1 変換キ一定義の変更

注意

dxjim では、省略時の設定では、JVMS 変換キ一定義を使用します。

アプリケーションで使用する変換キ一定義はつぎの方法で変更することができます。

ここでは、EVEJ キーを使うように設定してみます。

- 環境変数 IM_PROFILE をつぎのように定義する

```
% setenv IM_PROFILE im_profile_evej  (csh を使用している場合)

$ IM_PROFILE=im_profile_evej
$ export IM_PROFILE                  (shまたはkshを使用している場合)
```

JVMS キー、TARO キー、CTRL キー、KEYPAD キーを使う場合は、上の例の im_profile_evej をそれぞれ次のように変更します。

- JVMS キー: im_profile_jvms
- TARO キー: im_profile_taro
- CTRL キー: im_profile_ctrl
- KEYPAD キー: im_profile_keypad

この定義をユーザの .cshrc (sh を使用している場合は .profile) に記述しておく、以後のセッションで選択したキー定義を使うことができます。

新しいキー定義を使用するには、アプリケーションを再起動してください。

詳しくは『ユーザ・キー定義 利用者の手引き』を参照してください。

7.4 変換規則

本節では、変換に関する次の規則について説明します。

- ローマ字かな変換 (7.4.1 項)
読みをローマ字で入力する際のローマ字とかなの対応を示します。
- 特殊文字変換 (7.4.2 項)

漢字変換によって、1文字の記号を特殊文字に変換する際の記号と特殊文字の対応を示します。

- 記号変換（7.4.3 項）

記号変換によって、1～3文字の英数記号を特殊文字に変換する際の英数記号と特殊文字の対応を示します。

- コード入力（7.4.4 項）

記号変換によって、漢字コードから漢字に変換する方法について説明します。

7.4.1 ローマ字かな変換対応表

ローマ字 ^a					ひらがな				
a	i	u	e	o	あ	い	う	え	お
ka	ki	ku	ke	ko	か	き	く	け	こ
qa	qi	qu	qe	qo	くぁ	くぃ	く	くえ	くぉ
sa	si	su	se	so	さ	し	す	せ	そ
ta	ti	tu	te	to	た	ち	つ	て	と
na	ni	nu	ne	no	な	に	ぬ	ね	の
ha	hi	hu	he	ho	は	ひ	ふ	へ	ほ
fa	fi	fu	fe	fo	ふぁ	ふぃ	ふ	ふえ	ふぉ
ma	mi	mu	me	mo	ま	み	む	め	も
ya	yi	yu	ye	yo	や	い	ゆ	え	よ
ra	ri	ru	re	ro	ら	り	る	れ	ろ
la	li	lu	le	lo	ら	り	る	れ	ろ
wa	wi	wu	we	wo	わ	ゐ	う	ゑ	を
n					ん				
xa	xi	xu	xe	xo	ぁ	ぃ	う	え	ぉ
xka			xke		カ			ケ	
xtu					っ				
xtsu					っ				

ローマ字 ^a					ひらがな				
xwa	xwu				わ	う			
xya	xyi	xyu	xye	xyo	や	い	ゆ	え	よ
ga	gi	gu	ge	go	が	ぎ	ぐ	げ	ご
za	zi	zu	ze	zo	ざ	じ	ず	ぜ	ぞ
ja	ji	ju	je	jo	じゃ	じ	じゅ	じえ	じょ
da	di	du	de	do	だ	ぢ	づ	で	ど
ba	bi	bu	be	bo	ば	び	ぶ	べ	ぼ
pa	pi	pu	pe	po	ぱ	ぴ	ぷ	ぺ	ぽ
va	vi	vu	ve	vo	ヴァ	ヴィ	ヴ	ヴェ	ヴォ
kya	kyi	kyu	kye	kyo	きゃ	きい	きゅ	きえ	きょ
qya	qyi	qyu	qye	qyo	くや	くい	くゅ	くえ	くょ
kwa	kwi	kwu	kwe	kwo	くわ	くい	く	くえ	くお
sya	syi	syu	sye	syo	しゃ	しい	しゅ	しえ	しよ
sha	shi	shu	she	sho	しゃ	し	しゅ	しえ	しよ
tya	tyi	tyu	tye	tyo	ちゃ	ちい	ちゅ	ちえ	ちよ
cya	cyi	cyu	cye	cyo	ちゃ	ちい	ちゅ	ちえ	ちよ
cha	chi	chu	che	cho	ちゃ	ち	ちゅ	ちえ	ちよ
tsa	tsi	tsu	tse	tso	つあ	つい	つ	つえ	つお
tha	thi	thu	the	tho	てや	てい	てゅ	てえ	てよ
nya	nyi	nyu	nye	nyo	にや	にい	にゅ	にえ	によ
hya	hyi	hyu	hye	hyo	ひや	ひい	ひゅ	ひえ	ひよ
fya	fyi	fyu	fye	fyo	ふや	ふい	ふゅ	ふえ	ふよ
mya	myi	myu	mye	myo	みや	みい	みゅ	みえ	みよ
rya	ryi	ryu	rye	ryo	りや	りい	りゅ	りえ	りよ
lya	lyi	lyu	lye	lyo	りや	りい	りゅ	りえ	りよ

ローマ字 ^a					ひらがな				
gya	gyi	gyu	gye	gyo	ぎゃ	ぎい	ぎゅ	ぎえ	ぎょ
gwa	gwi	gwu	gwe	gwo	ぐわ	ぐい	ぐゅ	ぐえ	ぐょ
zya	zyi	zyu	zye	zyo	じゃ	じい	じゅ	じえ	じょ
jya	jyi	jyu	jye	jyo	じゃ	じい	じゅ	じえ	じょ
dya	dyi	dyu	dye	dyo	ぢゃ	ぢい	ぢゅ	ぢえ	ぢょ
dha	dhi	dhu	dhe	dho	でゃ	でい	でゅ	でえ	でょ
bya	byi	byu	bye	byo	びゃ	びい	びゅ	びえ	びょ
pya	pyi	pyu	pye	pyo	ぴゃ	ぴい	ぴゅ	ぴえ	ぴょ

^a 補足

- ローマ字は大文字でも小文字でもかまいません。
- 促音（"っ"）を入力する場合は、次にくる子音を重ねます。
- "ん"の次に母音がくるときには""（アポストロフィ）を間にいれます。

7.4.2 特殊文字変換対応表

変換前	変換後
;(セミコロン)	, (読点)
.(ピリオド)	。(句点)
+(正符号)	・ (中点)
-(負符号 , ハイフン)	ー (長音記号)
¥ (円記号)	\ (逆斜線)
‘ (左一重引用符)	‘ (左一重引用符)
(オーバーライン)	~ (波ダッシュ)
@ (単価記号)	(二重丸)
* (星印)	(米印)
= (等号)	≡ (げた記号)
[(始め大カッコ)	「 (始めかぎ括弧)

変換前	変換後
)(終り大カッコ)	」(終りかぎ括弧)
<(不等号, より小)	(始め山括弧)
>(不等号, より大)	(終わり山括弧)

7.4.3 記号変換対応表

- 1 文字変換

変換前: 1 2 3 4 5 6 7 8 9 0 -
 変換後:

- 2 文字変換

変換前: F1 F2 F3 F4 F5 F6 F7 F8 F9 F0 F-
 変換後:

変換前: T1 T2 T3 T4 T5 T6 T7 T8 T9 T0 T-
 変換後:

変換前: Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y0 Y-
 変換後:

変換前: ,, ^I /] /^ || ,. .. `` ([]) << >> [[]]
 変換後: " 全 々 〃 ... " [] 《 》 『 』

変換前: [()] +- XX :- =/ <= >= OO .; O> O+ /_ |_
 変換後: 【 】 ± × ÷

変換前: ~~ O) ~V =- =: *< *> V~ CC OC ;. S1 S2 (-
 変換後:

変換前:)- (_)_ (()) CU CA /¥ ¥/ ~| => == V-]-
 変換後: ー

変換前: .1 '1 '2 .C C/ L- .A %% SS >< () <> [] <¥
 変換後: ° ¢ £ % ¤

変換前: </ =| >- ><- |^ |V ## |. .^ +1 +2]| C)
 変換後: 〒 † ‡ ¶

- 3 文字変換

変換前: K>< K() K<> K[] K<¥ K</
 変換後:

7.4.4 コード入力

漢字コードを入力して記号変換を行うことによって、漢字コードで漢字を入力することができます。漢字コードとしては、次の 2 種類を使用することができます。

- DEC 漢字コード
- 区点コード

DEC 漢字コードを入力する場合は、先頭に“X”を付けて指定します。

例) XA1B9 記号変換 ヲ

区点コードを入力する場合は、先頭に“J”を付けて指定します。

例) J125 記号変換 ヲ

DEC 漢字コードおよび区点コードに関しては『漢字コード表』を参照してください。

7.5 キーボードの「かなモード」の使用

Tru64 UNIX では、かなキーボードをサポートしています。システムのキーボード設定で日本語キーボードが設定してある場合、かなキーボードが使えます。

かなキーボードは、かなを入力できる「かなモード」と、アルファベットや数字を入力できる「アスキー・モード」の 2 つのモードを切り替えて使用することができます。省略時の設定はアスキー・モードになっています。

PCXAJ-AA では、右側の Ctrl² キーを押ことによって、アスキー・モードとかなモードが切り替わります。

他の日本語キーボードの場合は、かな を押すことによって、アスキー・モードとかなモードが切り替わります。

北アメリカ・キーボードでは、かな キーは Compose または Comp となっています。

² PCXAJ-AA には Ctrl キーが 2 つあります。

7.6 変換キー付き日本語キーボードのサポート

Tru64 UNIX では、**変換**や**無変換**などの日本語入力専用のキーを持つ次の変換キー付き日本語キーボードをサポートします。

- LK411-JJ/LK41W-JJ (JIS 配列)
- LK411-AJ/LK41W-AJ (ANSI 配列)
- LK421-JJ (JIS 配列)
- PCXAJ-AA/LK97W-AA (JIS 配列)

キーボードの設定については『リリース・ノート』を参照してください。

変換キー付き日本語キーボードでは、変換キー定義 (JVMS/EVEJ/KEYPAD) に応じたキー操作の他に、表 7-7 に示すキー操作を利用できます。

注意

変換キー定義が TARO の場合、表 7-7 に示すキー操作を使用することはできません。

表 7-7: 変換キー付き日本語キーボードの変換キー定義

機能	キー操作 ^a
漢字変換/文節次候補	変換
漢字候補一覧	Alt+変換
ひらがな変換/文節ひらがな変換	ひらがな
全角カタカナ変換/文節全角カタカナ変換	カタカナ^b
全角変換	Alt+ひらがな
半角カタカナ変換/文節半角カタカナ変換	Ctrl+ひらがな または Ctrl+Shift+ひらがな
半角変換	Ctrl+Alt+ひらがな
記号変換/コード入力	Alt+無変換
次文節移動	Shift+
前文節移動	Shift+

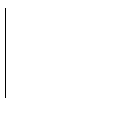
表 7-7: 変換キー付き日本語キーボードの変換キー定義 (続き)

機能	キー操作 ^a
文節縮小	Shift+
文節拡大	Shift+
文節前候補	前候補 ^c
無変換	無変換
変換取消	Shift+無変換

^aCtrl+ひらがなはCtrlキーを押しながらひらがなキーを押すことを示します。

^bカタカナはShiftキーを押しながらカタカナ/ひらがなキーを押すことを示します。

^c前候補はShiftキーを押しながら前候補/変換キーを押すことを示します。



この章では、日本語 Message Handler（以下 MH と略す）で実際に日本語のメールの送受信を行うために必要な設定、操作方法、注意点について説明します。

8.1 メール

メールは日常生活の手紙のやり取りに相当する作業をコンピュータ上で行う機能です。メールを使うと、コンピュータ上で作成されたメッセージをある特定のユーザ間で送受信することができます。

Tru64 UNIX では次の 2 つの部分によってメールの処理を行っています。

- メール配送システム

メールを集めて、宛先のユーザに送ります。ちょうど日常生活における郵便局の仕事に相当します。詳細については『ネットワーク管理ガイド』を参照してください。

- フロント・エンド

送信メッセージを作成してメール配送システムに渡したり、メール配送システムから送って来た受信メッセージの内容を表示したりします。これは、手紙を書いてポストに投函したり、配達された手紙を、郵便受けから取り出して読むことに相当します。

8.2 日本語対応のフロント・エンド

Tru64 UNIX は日本語に対応するフロント・エンド 日本語 MH を提供しています。

`/usr/bin/mh` 以下に用意されている日本語 MH のコマンドは、8 ビット・クリーンです。`/usr/bin/mh` をコマンド・サーチ・パスに追加すると日本語 MH のコマンドが有効になります。

メール配送システムとフロント・エンドの設定が完了すると、日本語 MH を使って、日本語のメールの送受信ができるようになります。

8.3 MH の操作方法

日本語 MH は、かな漢字変換を伴うこと、コード変換を行うことを除いて、英語版の MH を使用するときと同じ方法でメールを扱うことができます。

8.3.1 メールの送信

日本語 MH でメールの作成や転送を行うときは `comp` コマンドを使用します。この場合に日本語のメッセージを送るには、次の 2 つの方法があります。

- システム・レベルかな漢字変換を用いる。
- MH より日本語エディタ (`jvi`, `Mule`) を起動する。

`comp` を実行するときに `prompter` が起動される場合、システム・レベルかな漢字変換を用いて日本語のメッセージを入力します。

```
<endcondition>
% comp
To: yamamoto
cc: tanaka
Subject: Meeting
-----
```

明日のミーティングでは、山本さんに簡単な自己紹介をしていただきたいと思います。

よろしくお願いします。

```
-----
What now ?send
%
```

MH より日本語エディタ (`jvi`, `Mule` など) を起動する場合は、`comp` コマンドのオプション `-editor` を用いて日本語エディタをコマンド行で次のように指定するか、

```
% comp -editor jvi
```

`comp` コマンドを実行する前に、あらかじめ次のように日本語エディタの指定を書いた行を、ホーム・ディレクトリの `.mh_profile` に追加しておきます。

```
Editor: jvi
```

このようにしておくと、`comp` を起動したあと自動的に日本語エディタが起動され、日本語のメッセージを入力することができます。

送られて来たメールをメール・ドロップから取り込んで読む場合も、通常の MH の使用方法と同じです。

```

<endcondition>
% inc
Incorporating new mail into inbox...

78+ 05/08 yamamoto          re; Meeting<<自己紹介の件,わかりました。明日
79  05/08 tanaka           re; Meeting<<> いろいろと御気遣いいただきまし
%
% show
(Message inbox:78)
Return-Path: yamamoto
Message-Id: <9308050827.AA02539>
Mime-Version: 1.0
Content-Type: TEXT/PLAIN; charset=X-deckanji
To: suzuki
Cc: tanaka
Subject: re; Meeting
Date: Thu, 05 Aug 93 17:27:47 +0900
From: yamamoto
X-Mts: smtp

```

自己紹介の件,わかりました。明日のミーティングまでに話すことを簡単にまとめておきます。

いろいろと御気遣いいただきまして,本当にありがとう御座います。

山本

8.4 コード変換

メール配送システムとフロント・エンドの設定が完了すると,同じホスト間または同じアプリケーション・コードを使用するホスト間で日本語を使ったメールの送受信はできます。しかし,ネットワークを経由して他のホストにメールを送る場合,自分のホストが8ビット・クリーンになっていても,相手のホストが8ビット目をクリアするように設定していれば,DEC 漢字を含むメッセージを正常に送ることはできません。

そこで,DEC 漢字コードを含むメールを送る場合はDEC 漢字コードを7ビットの語長のJISコード(ISO-2022-JP)に変換してからメールを送信し,メールを受け取る場合は,逆にJISコードをDEC 漢字コードに変換するといった方法が考えられます。

この方法を用いれば,自分のホストの設定を気にする必要も,相手のホストが8ビット目をどのように扱うかを気にする必要もありません。また,日本

語 MH はこの変換を内部的に行うので、一定の設定を行えばユーザは変換処理を気にすることなく、外部とのメールの送受信ができます。

8.4.1 アプリケーション・コードとメール変換コード

メールのコード変換を行うときに変換の対象となる文字コードは次の 2 つです。

- アプリケーション・コード

それぞれのシステムで定義された、現在使っている文字コードセット。

- メール変換コード

メールを送受信するときに便宜的に使う文字コードセット。

アプリケーション・コードはシステムの起動時、またはユーザが login したときに定義される場合が一般的です。MH の中では、次の順番でアプリケーション・コードが認識されますのでアプリケーション・コードの設定が不十分な場合は、次のうちのいずれかに現在のアプリケーション・コードを設定してください。

1. 環境変数 LANG の値
2. \$HOME/.mh_profile の中で lang オプションの値として設定されているロケール情報

これに対して、メール変換コードはメール機能だけに必要なものなのでアプリケーション・コードとちがい特別に設定しなくてはなりません。

8.4.1.1 項 でメール変換コードの設定方法、8.4.1.2 項 でメール変換コードの認識順番について説明します。

8.4.1.1 メール変換コードの設定

メール変換コードを設定する必要があるのは、メール送信時です。日本語 MH は、アプリケーション・コードで書かれたメール・メッセージを次の方法で設定されたメール変換コードに変換してネットワークに送信します。

1. 環境変数 EXCODE にメール変換コードを定義する

メール変換コードを ISO-2022-JP にする場合には、コマンド・ラインで次のように入力します。

```
% setenv EXCODE ISO-2022-JP
```

あるいは、ユーザの .login に上記の設定を追加しておきます。

2. `$HOME/.mh_profile` の `excode` オプションにメール変換コードを記述する。

MH の設定ファイル `$HOME/.mh_profile` に次のように記述をすると MH を扱っている場合にだけ、出力コードが ISO-2022-JP となります。

```
excode: ISO-2022-JP
```

3. `/usr/lib/mail-codesets` にメール変換コードを記述する。

`/usr/lib/mail-codesets` は、システムの指定省略時のメール変換コードを設定するファイルです。システム特権を用いてファイルに次のように記述すると、システム指定省略時のメール変換コードは ISO-2022-JP になります。

```
ISO-2022-JP
```

メール変換コードは、環境変数 `EXCODE` , `.mh_profile` , `/usr/lib/mail-codesets` の順に参照されますので、それぞれに不整合がある場合は、上位の設定が有効になります。

8.4.1.2 メール変換コードの認識

送られてきたメッセージをメール変換コードからアプリケーション・コードに変換するためには、メッセージのメール変換コードを認識する必要があります。Tru64 UNIX では次のような順番で、メール変換コードを認識、決定します。

- ヘッダ部 (8.4.2 項を参照) に、`charset` フィールドがある場合、そこに記されたコードセット
- `/usr/lib/mail-codesets` で設定された、そのシステムにおける指定省略時のメール交換コード

もし、上の 2 つからメール変換コードが認識できない場合、コード変換は行なわれません。

8.4.2 メール送信時のコード変換

アプリケーション・コード、メール変換コードの設定が完了している場合、日本語 MH はメール送信時に `post(1)` コマンドのなかで自動的にコード変換を行い、送信します (変換そのものに関してユーザが設定することはありません) 。

Tru64 UNIX で作成したメール・メッセージを送信すると次の例のように Content-Type ヘッダの charset フィールドにメール交換コードが記されます (例の場合 ISO-2022-JP)。

```
Mime-Version: 1.0
Content-Type: TEXT/PLAIN; charset=ISO-2022-JP
```

charset に書かれたコード名は、メール受信側にどの文字コードセットをメール変換コードとして使ったかを示す大切なデータとなります。

8.4.3 メール受信時のコード変換

メールの受信時のコード変換は、設定により `inc(1)` コマンドで行う場合と、`show(1)` コマンドや `scan(1)` コマンドなどで行う場合の 2 種類があります。

- `inc(1)`で行う場合

メッセージを `spool` ディレクトリから メール・フォルダに取り込むときに (`inc(1)` 実行時に) コード変換します。この方法をとると、すでにコード変換されたファイルがフォルダにできるのでファイルの編集が容易です。その半面、コード変換が失敗した場合、メッセージの内容が失われてしまう危険があります。

`inc(1)` で変換を行う場合は、`.mh_profile` に次のような記述を加えます。

```
inc: -conv inshdr
```

この記述がない場合コード変換は `inc(1)` ではなく次の `show(1)` `scan(1)` などで行なわれます。

- `show(1)` `scan(1)`などで行う場合

メッセージをメール・フォルダから取り出して、実際に読む直前に (`show(1)` `scan(1)` 実行時に) コード変換します。この方法では、メッセージの損失を防ぐことができますが、フォルダ内のファイルを参照する場合、毎回変換しなくてはなりません。

8.5 MH コマンド一覧表

表 8-1 に「MH コマンド一覧」が記載されています。コマンドの機能が確かでないときに参照してください。なお、より詳しい内容については、それぞれのコマンドのリファレンス・ページを参照してください。

表 8-1: MH コマンド一覧

コマンド	機能
alex(1)	メッセージ・ヘッダからアドレスを抽出します。
ali(1)	メールの別名をリストします。
anno(1)	メッセージに注釈をつけます。
burst(1)	要約をメッセージ内にいれます。
comp(1)	メッセージを作成します。
dist(1)	メッセージを追加アドレスに再分配します。
folder(1)	現在のフォルダ/メッセージを設定/リストします。
folders(1)	すべてのフォルダをリストします。
forw(1)	次のメッセージに進みます。
inc(1)	新規のメールをまとめます。
mark(1)	メッセージにマークします。
mhl(1)	MHメッセージのフォーマット済みリストを作成します。
mhmail(1)	メールの送信または読み出しを行います。
mhpath(1)	MHメッセージとフォルダのフル・パス名をプリントします。
msgchk(1)	メッセージをチェックします。
msh(1)	MHシェル
next(1)	次のメッセージを表示します。
packf(1)	フォルダを1つのファイルに圧縮します。
pick(1)	内容に応じてメッセージを選択します。
prev(1)	前のメッセージを表示します。
prompter(1)	エディタ・フロント・エンドのプロンプト機能
rcvstore(1)	新規のメールを非同期で取り込みます。
refile(1)	他のフォルダのファイル・メッセージ
repl(1)	メッセージに返答します。
rmf(1)	フォルダを削除します。
rmm(1)	メッセージを削除します。

表 8-1: MH コマンド一覧 (続き)

コマンド	機能
scan(1)	各メッセージごとに1行のスキャン・リストを作成します。
send(1)	メッセージを送信します。
slocal(1)	メール・フックを受信します。
show(1)	メッセージを表示(リスト)します。
sortm(1)	メッセージをソートします。
whatnow(1)	送信のためのフロント・エンドのプロンプト機能
whom(1)	メッセージの受信者を報告します。
mh-alias(5)	MHメッセージ・システムの別名ファイル
mh-format(5)	MHメッセージ・システムのフォーマット・ファイル
mh-mail(5)	MHメッセージ・システムのメッセージ・フォーマット
mh_profile(5)	MHメッセージ・システムのユーザ・カスタマイズ
mtstailor(5)	MHのためのシステム・カスタマイズ
ap(8)	アドレスを解析してRFC 822形式にします。
conflict(8)	別名/パスワードの矛盾点を検索します。
dp(8)	日付を解析してRFC 822形式にします。
install-mh(8)	MH環境を初期化します。
post(8)	メッセージを配信します。

文字コード

この章では Tru64 UNIX の日本語機能を使用して処理することができる漢字コードについて解説します。また、複数の漢字コードを処理するために導入された、端末コードとアプリケーション・コードの概念についても解説します。

- 文字コード
- 端末コードとアプリケーション・コード
- プリンタ・コード
- ファイル・コードの変換

A.1 文字セット

文字とコードの対応の集合を文字セット(character set)と呼びます。JIS では表 A-1 のように文字セットを規定しています。これらに含まれる文字は 1, あるいは 2 バイトの文字コードで表します。文字の各バイトは 7 ビットだけが使用され、最上位ビット(MSB)は 0 (ゼロ) になっています。

表 A-1: JIS による文字セットの分類

分類	JIS の規格番号	文字コード
JIS ローマ字	JIS X 0201 LH	0xxxxxxx
JIS 片仮名	JIS X 0201 RH	0xxxxxxx
JIS 漢字	JIS X 0208 ^a	0xxxxxxx 0xxxxxxx
JIS 補助漢字	JIS X 0212	0xxxxxxx 0xxxxxxx

^aJIS 漢字にはいくつかの版(規定年度)がありますが、特に記述しない限り Tru64 UNIX では 1990 年版を参照しています。

この他、ANSI(米国規格協会) が定めた 1 バイトの文字コードである *ASCII* 文字セットも日本で広く使用されています。ASCII 文字セットは、JIS X

0201 LH とほぼ同じですが，表 A-2 の 2 つのコードに割り当てられた図形文字が異なります。

表 A-2: ASCII と JIS X 0201 LH

文字コード	ASCII	JIS X 0201 LH
0x5c	バックスラッシュ(\)	円記号(¥)
0x7e	チルダ(`)	オーバーライン()

これらの文字セットを複数まとめたものをコードセット(code set) と呼びます。しかしこのままでは，異なる文字セットに属する文字コードが重なり合うために文字とコードを一意に対応づけることができません。このため，コードセットに含まれる文字セットに対して何らかの符号拡張が必要になります。

ISO(国際標準化機構) は各国に共通な文字コードの体系を定めた ISO 2022 を定めています。JIS(日本工業規格)は ISO 2022 を翻訳し，それを JIS X 0202 としました。JIS X 0202 に基づき符号拡張したコードセットのうち主なものをあげます。

- 7 単位系 JIS 漢字コードセット
- 8 単位系 JIS 漢字コードセット
- 日本語 EUC コードセット

また JIS X 0202 に基づかずに，日本独自の符号拡張を行なったコードセットとして次のものがあります。

- シフト JIS コードセット
- DEC 漢字コードセット
- Super DEC 漢字コードセット

それぞれのコードセットで利用できる文字セットを表 A-3 に示します。

表 A-3: 各コードセットで利用できる文字セット

コードセット	ASCII または JIS X 0201 LH	JIS X 0201 RH	JIS X 0208	JIS X 0212
7 単位系 JIS				
7 単位系 JIS				
8 単位系 JIS				
日本語 EUC コード				
DEC 漢字コード		-		-
Super DEC 漢字コード				
シフト JIS				-

各 JIS 規格の詳細は、それぞれの『JIS 規格票』を参照してください。

また、ISO は世界中の文字を 1 つのコードセットで表現する Universal Character Set (UCS) を ISO/IEC 10646 で定義しています。JIS では、ISO/IEC 10646 を翻訳して JIS X 0221 を定義しています。

ISO/IEC 10646 に基づき、Tru64 UNIX では次のエンコーディングをサポートします。

- UTF-8

A.1.1 7 単位系 JIS 漢字コードセット

JIS X 0201 LH、JIS X 0201 RH、JIS X 0208、JIS X 0212 の各文字セットに含まれるすべての文字を 7 ビットの文字コードに符号化するコードセットです。本書では以後、*7-bit JIS* コードと呼びます。

表 A-4: 7-bit JIS コードセットの文字コード

文字セット	文字コード
JIS X 0201 LH	0xxxxxxx
JIS X 0201 RH	0xxxxxxx
JIS X 0208	0xxxxxxx 0xxxxxxx
JIS X 0212	0xxxxxxx 0xxxxxxx
C0 制御文字	000xxxx

表 A-4 のようにコード化された文字は他の文字のコードと重なるため、一意に識別することができません。そこで 7-bit JIS コードでは、JIS X 0202 に規定されている符号拡張を使用して状態 (state) を持つことでそれぞれの文字を識別します。

漢字イン・シーケンス (指定省略時の設定では "ESC \$ B") から後ろに現れたコードは、JIS X 0208 の漢字文字として、漢字アウト・シーケンス (指定省略時の設定では "ESC (B") から後ろに現れたコードは、ASCII の文字として処理します。補助漢字イン・シーケンス (指定省略時の設定では "ESC \$ (D") から後ろに現れたコードは、JIS X 0212 補助漢字として処理し、漢字アウト・シーケンス (指定省略時の設定では "ESC (B") から後ろに現れたコードは、ASCII の文字として処理します。また、SO (0x0e) から SI (0x0f) までの間のコードは、JIS X 0201 RH のカタカナ文字として処理します。JIS X 0201 RH のカタカナ文字を表現するために、SO (0x0e) および SI (0x0f) を使用する代わりに、カナ・イン・シーケンス ("ECS (I ") を使用する方法もあります。

注意

7-bit JIS コードセットは、端末コードとして使用することはできませんが、ロケール名の指定に使用することはできません。

7-bit JIS コードについての詳細は、日本語リファレンス・ページの JIS7(5) を参照してください。

A.1.2 8 単位系 JIS 漢字コードセット

JIS X 0201 LH, JIS X 0208, JIS X 0212 の各文字セットに含まれるすべての文字を 7 ビットの文字コードに符号化し、JIS X 0201 RH 文字セットに含まれる文字は 8 ビットの文字コードに符号化するコードセットです。本書では以後、*8-bit JIS* コードと呼びます。8-bit JIS コードの文字の符号化は、表 A-5 のようになります。

表 A-5: 8-bit JIS コードセットの文字コード

文字セット	文字コード
JIS X 0201 LH	0xxxxxxx
JIS X 0201 RH	1xxxxxxx
JIS X 0208	0xxxxxxx 0xxxxxxx
JIS X 0212	0xxxxxxx 0xxxxxxx
C0 制御文字	000xxxx

漢字イン・シーケンス（指定省略時の設定では "ESC (B)" から後ろに現れたコードは，JIS X 0208 の漢字文字として，漢字アウト・シーケンス（指定省略時の設定では "ESC \$ B"）から後ろに現れたコードは，ASCII の文字として処理します。補助漢字イン・シーケンス（指定省略時の設定では "ESC \$ (D)" から後ろに現れたコードは，JIS X 0212 補助漢字として処理し，漢字アウト・シーケンス（指定省略時の設定では "ESC \$ B"）から後ろに現れたコードは，ASCII の文字として処理します。MSB が 1 であるデータは，JIS X 0201 RH のカタカナ文字として処理されます。また 7-bit JIS で使用される SI と SO は無視されます。

注意

8-bit JIS コードセットは，端末コードとして使用することはできませんが，ロケール名の指定に使用することはできません。

8-bit JIS コードについての詳細は，日本語リファレンス・ページの JIS8(5) を参照してください。

A.1.3 日本語 EUC コードセット

EUC(Extended UNIX Code)は，AT&T 日本語UNIXシステム諮問委員会から 1985 年に提案された「UNIX システム日本語機能提案書」の中で定義された日本語用 UNIX 内部コード体系をもとに，AT&T ベル研究所で世界各国で使用可能となるように拡張したコード体系です。この EUC を日本語に適用したコードセットを，日本語 EUC コードセットと呼びます。

EUC は CS0，CS1，CS2，CS3 の 4 つのコードセットを持ちます。

- CS0 に属する文字は、コードの MSB（最上位ビット）が必ず 0（ゼロ）となるように符号化されます。世界各国のどの EUC の適用においても、CS0 は ASCII 文字セットと決められています。
- CS1 に属する文字は、コードの各バイトの MSB が必ず 1 となるように符号化されます。日本語 EUC では、CS1 は JIS X 0208 となっています。
- CS2 に属する文字は SS2コード(0x8e) に続いて、コードの各バイトの MSB が必ず 1 となるように符号化されます。日本語 EUC では、CS2 は JIS X 0201 RH となっています。
- CS3 に属する文字は SS3コード(0x8f) に続いて、コードの各バイトの MSB が必ず 1 となるように符号化されます。日本語 EUC では、CS3 は JIS X 0212 となっています。

日本語 EUC コードセットの文字の符号化は、表 A-6 のようになります。

表 A-6: 日本語 EUC の文字コード

コードセット	対応する文字セット	符号化
CS0	ASCII	0xxxxxxx
CS1	JIS X 0208	1xxxxxxx 1xxxxxxx
CS2	JIS X 0201RH	SS2 1xxxxxxx
CS3	JIS X 0212	SS3 1xxxxxxx 1xxxxxxx
C0 制御文字		000xxxxx
C1 制御文字		100xxxxx

JIS X 0208 と JIS X 0212 の未定義領域のうち、次の領域をベンダやユーザが定義するユーザ定義文字領域として割り当てることができます。

- JIS X 0208 の未定義領域の 85 区 から 94 区 までの領域
- JIS X 0212 の未定義領域の 78 区 から 94 区 までの領域

注意

JIS X 0212 の未定義領域の 78 区から 84 区までは、JIS が将来の拡張のために確保している領域です。したがって将来、JIS がこ

の領域に文字を割り当てることがあれば、ユーザが定義した文字と衝突する可能性があります。

日本語 EUC コードセットについての詳細は、日本語リファレンス・ページの eucJP(5) を参照してください。

A.1.4 DEC 漢字コードセット

DEC 漢字コードは、以下の文字セットから構成されるコードセットです。

- ASCII または JIS X 0201 LH
- DEC 漢字文字セット 1990 年版
- DEC 拡張漢字文字セットのユーザ定義領域

ASCII 文字、あるいは JIS X 0201 LH は 1 バイトの文字コードで表します。バイトの MSB は必ず 0（ゼロ）となっています。

DEC 漢字文字セット 1990 年版は表 A-7 のように、JIS X 0208-1990 の非漢字、第 1 水準漢字、第 2 水準漢字の 6,879 文字から構成されます。1 文字は 2 バイトで表され、各バイトの MSB は必ず 1 となっています。DEC 漢字文字セット 1990 年版の区点番号は、JIS X 0208-1990 の区点番号と同じです。

表 A-7: DEC 漢字文字セット 90 年版

区番号	割り当て	文字数
1 区 から 8 区	非漢字文字 ^a	524 文字
9 区 から 15 区	JIS 保留領域	
16 区 から 47 区	第 1 水準漢字	2,965 文字
48 区 から 84 区	第 2 水準漢字	3,390 文字
85 区 から 94 区	JIS 保留領域	

^a特殊文字、数字、ローマ字、ひらがな、カタカナ、ギリシア文字、ロシア文字、罫線素片

DEC 漢字コードセットでは、DEC 拡張漢字文字セットのユーザ定義領域として、8,836 文字のユーザ定義文字のコードを割り当てることができます。拡張文字セットに属する文字は 1 バイト目の MSB が 1 で、2 バイト目の MSB が 0（ゼロ）である 2 バイトのデータとして表されます。この中で 32 区から 94 区までは、HP が確保している領域です。したがって、

ユーザが定義した外字は DEC 拡張漢字文字セットの 1 区から 31 区までの領域に割り当てることができます。

DEC 漢字コードの文字コードは表 A-8 のようになっています。

表 A-8: DEC 漢字コードの文字コード

文字セット	文字コード
ASCII あるいは JIS X 0201 LH	0xxxxxxx
JIS X 0208	1xxxxxxx 1xxxxxxx
DEC 拡張漢字文字セット	1xxxxxxx 0xxxxxxx
C0 制御文字	000xxxxx
C1 制御文字	100xxxxx

DEC 漢字コードについての詳細は、『漢字コード表』および日本語リファレンス・ページの `deckanji(5)` を参照してください。

A.1.5 Super DEC 漢字コードセット

Super DEC 漢字コードセットは、DEC 漢字コードセットを拡張して、日本語 EUC の CS2 と CS3 を扱えるようにしたコードセットです。

Super DEC 漢字コードセットは、以下の文字コードセットから構成されます。

- ASCII あるいは JIS X 0201 LH
- DEC 漢字文字セット 1990 年版
- DEC 拡張漢字文字セット
- JIS X 0201 RH
- JIS X 0212-1990

ASCII あるいは JIS X 0201 LH、DEC 漢字文字セット 1990 年版、DEC 拡張漢字文字セット、はそれぞれ DEC 漢字コードセットと同じように符号化されます。

JIS X 0201 RH に属する文字は SS2(0x8e) に続く 1 バイトの文字データで表します。このバイトの MSB は必ず 1 になっています。

JIS X 0212-1990 に属する文字は SS3(0x8f) に続く 2 バイトの文字データで表します。各バイトの MSB は必ず 1 になっています。

Super DEC 漢字コードの文字コードは表 A-9 のようになっています。

表 A-9: Super DEC 漢字の文字コード

文字セット	文字コード
ASCII あるいは JIS X0201 LH	0xxxxxxx
JIS X0208	1xxxxxxx 1xxxxxxx
DEC 拡張漢字文字セット	1xxxxxxx 0xxxxxxx
JIS X0201 RH	SS2 1xxxxxxx
JIS X0212	SS3 1xxxxxxx 1xxxxxxx
C0 制御文字	000xxxxx
C1 制御文字	100xxxxx

また Super DEC 漢字コードセットでは表 A-10 のように、DEC 拡張漢字文字セットおよび JIS X 0208、JIS X 0212 の未定義領域にユーザ定義の外字を割り当てることができます。

表 A-10: Super DEC 漢字におけるユーザ定義文字領域

文字セット	割り当て可能な領域
JIS X 0208	85 区から 94 区
JIS X 0212	78 区から 94 区
DEC 拡張漢字文字セット	1 区から 94 区まで

Super DEC 漢字コードについての詳細は、日本語リファレンス・ページの `sdeckanji(5)` を参照してください。

A.1.6 シフト JIS コードセット

シフト JIS コードセットは、パーソナル・コンピュータの世界で広く使用されている漢字コードです。各文字セットは表 A-11 のように符号化されています。

表 A-11: シフト JIS コードの符号化

C0 制御文字	0x00 から 0x1f
Space	0x20
JIS X 0201 LH	0x21 から 0x7e
Del	0x7f
未定義	0x80
JIS X 0208 漢字の 1 バイト目	0x81 から 0x9f
未定義	0xa0
JIS X 0201 RH	0xa1 から 0xdf
JIS X 0208 漢字の 1 バイト目	0xe0 から 0xef
ユーザ定義文字の 1 バイト目	0xf0 から 0xfc
未定義	0xfc から 0xff

コードと漢字およびユーザ定義文字の区点番号との対応は表 A-12 のようになっています。

表 A-12: 漢字，ユーザ定義文字の区点との対応

0x8140 ... 0x819c	JIS X 0208 1 区 1 点 ... 1 区 94 点 (0x817f を除く)
0x819f ... 0x81fc	JIS X 0208 2 区 1 点 ... 2 区 94 点
0x8240 ... 0x829e	JIS X 0208 3 区 1 点 ... 3 区 94 点 (0x827f を除く)
0x8140 ... 0x819c	JIS X 0208 4 区 1 点 ... 4 区 94 点
⋮	⋮
0x9f9f ... 0x9ffc	JIS X 0208 62 区 1 点 ... 62 区 94 点 (0x817f を除く)
0xe040 ... 0xe09e	JIS X 0208 63 区 1 点 ... 63 区 94 点 (0xe07f を除く)
⋮	⋮
0xef40 ... 0xef9e	JIS X 0208 93 区 1 点 ... 93 区 94 点 (0x817f を除く)
0xef9f ... 0xeffc	JIS X 0208 94 区 1 点 ... 94 区 94 点 (0xe07f を除く)
0xf040 ... 0xf0fc	ユーザ定義文字 (0xf07f を除く)

表 A-12: 漢字，ユーザ定義文字の区点との対応 (続き)

⋮	⋮
0xfc40 ... 0xfcfc	ユーザ定義文字 (0xfc7f を除く)

シフト JIS コードでは，次の領域にユーザ定義文字を割り当てることができます。

- 0xf040 から 0xfcfc (ただし第 2 バイトが 0x00 ~ 0x3f, 0x7f, 0xfd ~ 0xff であるものを除く)

シフト JIS コードについての詳細は，日本語リファレンス・ページの `shiftjis(5)` を参照してください。

A.1.7 UTF-8

UTF-8 は，世界中の文字を 1 つのコードセットで表現できるように ISO/IEC 10646 で定義された 4 オクテットの UCS-4 と 2 オクテットの UCS-2 を符合化します。UTF-8 は UCS 交換フォーマットとして可変長コードで表現され，ASCII コードを 1 バイトで表現するとともに，マルチバイトの可変長コード部分に ASCII コードを使用しないように配慮されています。

表 A-13: UCS-4 のエンコーディング範囲と UTF-8 のビット配列

エンコーディング範囲 (16 進数)	UTF-8 オクテット列 (2 進数)
0000 0000-0000 007F	0xxxxxxx
0000 0080-0000 07FF	110xxxxx 10xxxxxx
0000 0800-0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000-001F FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
0020 0000-03FF FFFF	1111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
0400 0000-7FFF FFFF	11111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

UTF-8 については，Unicode(5) リファレンス・ページを参照してください。



B

日本語ロケールの定義内容

ここでは、Tru64 UNIX で提供している 5 つの日本語ロケール (ja_JP.eucJP , ja_JP.SJIS , ja_JP.deckanji , ja_JP.sdeckanji , UTF-8) の定義内容について説明します。4 つの日本語ロケールの定義内容は、コードセットに依存する定義を除いて、共通の定義となっています。Tru64 UNIX で提供している日本語ロケールは、1993 年に発表された「UI-OSF 日本語環境実装規約 Version 1.1」の定義に適合するように実装されています。

以下では、CHARMAP , LC_CTYPE , LC_COLLATE , LC_MESSAGES , LC_MONETARY , LC_NUMERIC , LC_TIME のそれぞれの定義について説明します。ロケールで定義する項目やキーワードの意味などの詳細については、`charmap(4)` および `locale(4)` を参照してください。

B.1 CHARMAP

ロケール・データベースを作成する際に参照される `charmap` ファイルはコードセットごとに存在し、各コードセットの属性やロケール定義に使用されるシンボリック名とコード値との対応を記述します。表 B-1 は、`locale -k CHARMAP` コマンドで表示される各日本語コードセットの属性を `charmap` ファイルのキーワードごとに示しています。プログラム実行時のロケールの `code_set_name` の値は、`nl_langinfo(CODESET)` 呼出しで取得することができます。

表 B-1: 日本語コードセットの属性

	日本語 EUC	シフトJIS	DEC 漢字	Super DEC 漢字	UTF-8
<code>code_set_name</code>	"eucJP"	"SJIS"	"deckanji"	"sdeckanji"	"UTF-8"
<code>mb_cur_max</code>	3	2	2	3	3
<code>mb_cur_min</code>	1	1	1	1	1

B.2 LC_CTYPE

LC_CTYPE には、`isw*` 関数の動作を決める文字クラスの情報と `tow*` 関数の動作を決める文字変換の情報が定義されています。文字クラスには、C 言語規格や POSIX 規格などの標準仕様で規定している標準の文字クラスと、ロケールごとに必要な文字クラスを追加して定義した拡張文字クラスの二種類があります。標準の文字クラスは、XPG4 や POSIX の標準に適合したシステムには必ず存在する文字クラスです。拡張文字クラスは、標準規格には規定されていない日本語ロケールとして必要な文字クラスを追加したものです。表 B-2 に標準の文字クラスの定義内容を、表 B-3 に拡張文字クラスの定義内容を示します (UTF-8 では、拡張文字クラスは未定義です)。プログラムで拡張文字クラスの判定を行う場合は、XPG4 で規定されている `wctype` 関数と `iswctype` 関数を組み合わせて使用します。

LC_CTYPE には、文字クラスの他に、`towupper` 関数および `towlower` 関数から参照される大文字と小文字の対応関係を定義します。表 B-4 は、`toupper` および `tolower` に定義されているアルファベットを示しています。

表 B-2: 標準の文字クラス

文字クラス名	定義内容
upper	ASCII の 'A' ~ 'Z', JIS X0208 のローマ字大文字, JIS X0208 のギリシア文字大文字, JIS X0208 のロシア文字大文字, JIS X0212 のダイアクリティカルマーク付き ギリシアアルファベット文字大文字, JIS X0212 のキリル系アルファベット大文字, JIS X0212 のラテン系アルファベット大文字, JIS X0212 のダイアクリティカルマーク付き ラテンアルファベット大文字
lower	ASCII の 'a' ~ 'z', JIS X0208 のローマ字小文字, JIS X0208 のギリシア文字小文字, JIS X0208 のロシア文字小文字, JIS X0212 のダイアクリティカルマーク付き ギリシアアルファベット文字小文字, JIS X0212 のキリル系アルファベット小文字, JIS X0212 のラテン系アルファベット小文字, JIS X0212 のダイアクリティカルマーク付き ラテンアルファベット小文字
alpha	upper 及び lower に定義されている文字
digit	ASCII の '0' ~ '9'
space	ASCII の ' '(2/0), '\t'(0/9), '\n'(0/10), '\v'(0/11), '\f'(0/12), '\r'(0/13) JIS X0208 の間隔 (1 区 1 点)
cntrl	C0, C1 コントロール文字 (ただし, eucJP 及び sdeckanji については, SS2 及び SS3 を除く)

表 B-2: 標準の文字クラス (続き)

文字クラス名	定義内容
punct	<ul style="list-style-type: none"> • ASCII の '!' (2/1) ~ '/' (2/15) , ':' (3/10) ~ '@' (4/0) , '[' (5/11) ~ '`' (6/0) , '{' (7/11) ~ '~' (7/14) • JIS X0208 の '、' (1 区 2 点) ~ '!' (1 区 10 点) , '”(1 区 13 点) ~ '—' (1 区 18 点) , '”' (1 区 23 点) , '／' (1 区 26 点) , ' ’' (1 区 29 点) ~ ' ’' (1 区 94 点) • JIS X0201 RH の句点 (2/1) , 始かっこ (2/2) , 終かっこ (2/3) , 読点 (2/4) , 中点 (2/5) • JIS X0212 のダイアクリティカルマーク (2 区 15 点 ~ 25 点) , 記述記号 (2 区 34 点 , 2 区 36 点) , 単位記号 (2 区 80 点) , 一般記号 (2 区 35 点 , 75 点 ~ 79 点 , 81 点)
graph	<ul style="list-style-type: none"> • ASCII の upper , lower , digit , punct に含まれる文字 • JIS X0208 の間隔を除くすべての文字 (JIS 未定義部分を含む) • JIS X0201 RH のすべての文字 (2/1 ~ 5/15) • JIS X0212 のすべての文字 (JIS 未定義部分を含む) • ユーザ定義文字領域のすべての文字
print	<ul style="list-style-type: none"> • graph に含まれるすべての文字 • ASCII の ' ' (2/0) • JIS X 02 区 08 の間隔 (1 区 1 点)
xdigit	ASCII の '0' ~ '9' , 'a' ~ 'f' , 'A' ~ 'F'
blank	ASCII の ' ' (2/0) , '\t' (0/9)

表 B-3: 拡張文字クラス

文字クラス名	定義内容
ascii	ASCII のすべての文字
line	JIS X0208 の罫線素片 (8 区 1 点 ~ 8 区 32 点)
jdigit	JIS X0208 の数字 ('0' ~ '9')

表 B-3: 拡張文字クラス (続き)

文字クラス名	定義内容
paren	<ul style="list-style-type: none"> • ASCII の '(', ')', '[', ']', '{', '}' • JIS X0208 の括弧記号 (1 区 38 点 ~ 1 区 59 点) • JIS X0201 RH の始かっこ (2/2), 終かっこ (2/3)
jparen	<ul style="list-style-type: none"> • JIS X0208 の括弧記号 (1 区 38 点 ~ 1 区 59 点) • JIS X0201 RH の始かっこ (2/2), 終かっこ (2/3)
jisx0201	JIS X0201 のすべての文字
jisx0201r	JIS X0201 RH のすべての文字
jisx0208	JIS X0208 のすべての文字 (JIS 定義部分のみ)
jisx0212	JIS X0212 のすべての文字 (JIS 定義部分のみ)
udc	ユーザ定義文字
vdc	ベンダ定義文字 (Tru64 UNIX にはないため、未定義)
gaiji	ユーザ定義文字およびベンダ定義文字
jhira	JIS X0208 の 'ぁ'(4 区 1 点) ~ 'ん'(4 区 83 点), 'ゝ'(1 区 11 点), 'ゞ'(1 区 12 点), 'ゑ'(1 区 21 点), 'ゐ'(1 区 22 点), 'ー'(1 区 28 点)
jkata	JIS X0208 の 'ァ'(5 区 1 点) ~ 'ヶ'(5 区 86 点), 'ヰ'(1 区 11 点), 'ヱ'(1 区 12 点), 'ヰ'(1 区 19 点), 'ヱ'(1 区 20 点), 'ー'(1 区 28 点)
jhankana	JIS X0201 RH のカタカナ (2/6 ~ 5/13), 濁点 (5/14), 半濁点 (5/15)
jkanji	<ul style="list-style-type: none"> • JIS X0208 の漢字 (16 ~ 84 区の JIS 定義文字), '仝' (1 区 24 点), '々' (1 区 25 点), '〇' (1 区 27 点) • JIS X0212 の漢字 (16 ~ 77 区の JIS 定義文字)
jspace	JIS X0208 の間隔 (1 区 1 点)
english	JIS X0208 の 'A' ~ 'Z', 'a' ~ 'z'
number	JIS X0208 の '0' ~ '9'
special	JIS X0208 の特殊文字 (1 ~ 2 区の JIS 定義文字)
phonogram	JIS X0208 の 'ぁ'(4 区 1 点) ~ 'ん'(4 区 83 点), 'ァ'(5 区 1 点) ~ 'ヶ'(5 区 86 点)
ideogram	JIS X0208 のすべての文字 (JIS 未定義部分を含む)

表 B-4: **toupper** および **tolower** に定義されているアルファベット

ASCII のローマ字 (小文字 'a' ~ 'z' 大文字 'A' ~ 'Z')
JIS X0208 のローマ字 (小文字 'a' ~ 'z' 大文字 'A' ~ 'Z')
JIS X0208 のギリシア文字 (小文字 ' ' ~ ' ' 大文字 ' ' ~ ' ')
JIS X0208 のロシア文字 (小文字 ' ' ~ ' ' 大文字 ' ' ~ ' ')
JIS X0212 のダイアクリティカルマーク付きギリシアアルファベット
JIS X0212 のキリル系アルファベット
JIS X0212 のラテン系アルファベット
JIS X0212 のダイアクリティカルマーク付きラテンアルファベット

B.3 LC_COLLATE

LC_COLLATE には、文字の照合順序を定義します。日本語ロケールで定義されている文字の照合順序は、次の番号の小さい順に定義されています。

1. ASCII の文字
2. JIS X0201 RH の文字
3. JIS X0208 の文字 (JIS 未定義部分を含む)
4. JIS X0212 の文字 (JIS 未定義部分を含む)
5. DEC 拡張漢字領域のユーザ定義文字

それぞれの文字セット内での順序は、符号の小さい順になります。

B.4 LC_MESSAGES

LC_MESSAGES には、YES(肯定)/NO(否定) で答える質問に対する入力文字列を YES あるいは NO と判定するためのルールを正規表現で定義します。日本語ロケールでは、ASCII あるいは JIS X0208 のアルファベットで 'y' または 'Y' で始まる文字列は YES、ASCII あるいは JIS X0208 のアルファベットで 'n' または 'N' で始まる文字列は NO と判定するように定義しています。

表 B-5 に LC_MESSAGES の定義内容を示します。

表 B-5: LC_MESSAGES の定義

キーワード	定義内容
yesexpr	"^[yYy Y]"
noexpr	"^[nN n N]"

B.5 LC_MONETARY

LC_MONETARY には、金額の表現に使用される文字列やルールを定義します。表 B-6 に LC_MONETARY の定義内容を示します。

表 B-6: LC_MONETARY の定義

キーワード	定義内容
int_curr_symbol	"JPY "
currency_symbol	"¥"
mon_decimal_point	"."
mon_thousands_sep	","
mon_grouping	3
positive_sign	""
negative_sign	"_"
int_frac_digits	0
frac_digits	0
p_cs_precedes	1
p_sep_by_space	0
n_cs_precedes	1
n_sep_by_space	0
p_sign_posn	1
n_sign_posn	4

B.6 LC_NUMERIC

LC_NUMERIC には、金額以外の数値の表現に使用される文字列やルールを定義します。表 B-7 に LC_NUMERIC の定義内容を示します。

表 B-7: LC_NUMERIC の定義

キーワード	定義内容
decimal_point	"."
thousands_sep	","
grouping	3

B.7 LC_TIME

LC_TIME には、日付および時間の表現に使用される文字列やルールを定義します。表 B-8に LC_TIME の定義内容を示します。それぞれのキーワードの定義内容は、表中に示した strftime 関数の変換仕様から参照されます。

表 B-8: LC_TIME の定義

キーワード	strftime 関数の 変換仕様	定義内容
abday	%a	"日","月","火","水","木","金","土"
day	%A	"日曜日","月曜日","火曜日","水曜日","木曜日","金曜日","土曜日"
abmon	%b	"1月","2月","3月","4月","5月","6月","7月","8月","9月","10月","11月","12月"
mon	%B	"1月","2月","3月","4月","5月","6月","7月","8月","9月","10月","11月","12月"
d_t_fmt	%c	"%Y年%m月%d日 %H時%M分%S秒"
d_fmt	%x	"%Y年%m月%d日"
t_fmt	%X	"%H時%M分%S秒"
am_pm	%p	"午前","午後"
t_fmt_ampm	%r	"%p%I時%M分%S秒"

表 B-8: LC_TIME の定義 (続き)

キーワード	strftime 関数の 変換仕様	定義内容
era	-	<ul style="list-style-type: none">• "+:2:1990/01/01:+*:平成:%EC%Ey年";\• "+:1:1989/01/08:1989/12/31:平成:%EC元年";\• "+:2:1927/01/01:1989/01/07:昭和:%EC%Ey年";\• "+:1:1926/12/25:1926/12/31:昭和:%EC元年";\• "+:2:1913/01/01:1926/12/24:大正:%EC%Ey年";\• "+:1:1912/07/30:1912/12/31:大正:%EC元年";\• "+:2:1869/01/01:1912/07/29:明治:%EC%Ey年";\• "+:1:1868/09/08:1868/12/31:明治:%EC元年";\• "+:1:1/1/1:1868/09/07:西暦:%EC%Ey年";\• "-:1:-1/12/31:-*:紀元前:%EC%Ey年"
era_year	%EY	"%EC%Ey年"
era_d_fmt	%Ex	"%EY%m月%d日"
era_d_t_fmt	%Ec	"%EY%m月%d日 %H時%M分%S秒"

Motif の日本語機能

この章では、Motif を使用して日本語アプリケーションの開発を行うユーザーを対象に、Motif の日本語機能について説明します。

C.1 X Window System

C.1.1 X Locale

Tru64 UNIX では、X11R6.3 の国際化機能を使用して日本語のアプリケーションが作成できます。XmbDrawString() 等を用いた日本語の表示や、XmbLookupString() 等を使用した日本語の入力が可能です。

X library は以下の日本語ロケールをサポートします。

- ja_JP
- ja_JP.eucJP
- ja_JP.deckanji
- ja_JP.sdeckanji
- ja_JP.SJIS
- ja_JP.UTF-8

C.1.2 入力サーバ

Tru64 UNIX は DEC 日本語入力サーバ (dxjim) を提供します。dxjim を使用する場合は im modifier は "DEC" です。im modifier を省略した場合も dxjim が使用できます。

例

```
setenv XMODIFIER @im=DEC
```

サポートするプロトコルは、R6 XIM と DECXim です。DECXim は Tru64 UNIX の以前のバージョンや、OpenVMS、ULTRIX 上のクライアントの

ために使用されます。プロトコルの選択は自動的に行なわれるので、ユーザが意識する必要はありません。

DEC 日本語入力サーバ (dxjim) は以下の入力スタイルをサポートします。

XIMPreeditPosition,XIMStatusNothing	(OverTheSpot)
XIMPreeditCallbacks,XIMStatusNothing	(OnTheSpot)
XIMPreeditNothing,XIMStatusNothing	(Root)

また、DEC 入力サーバ・ライブラリを使用して、独自の入力サーバを作成することができます。そのような入力サーバでは、サーバで指定した名前を `im modifier` にセットします。DEC 入力サーバ・ライブラリについては『DEC 入力サーバ・ライブラリ』のドキュメントを参照してください。

C.1.3 X Toolkit Intrinsics

X11R6.3 においては、`XtSetLanguageProc()` を使用して Xlib とツールキットの言語の整合性をとることができます。これによってセッションと同じ言語でアプリケーションを動作させることが可能です。

C.1.4 Xlib

Tru64 UNIX は、X11R6.3 の Xlib を提供します。Input Method プロトコルとしては、R6 XIM と Ximp プロトコルをサポートします。入力サーバとの接続時には、R6 XIM、Ximp の順で入力サーバをサーチしますので、従来のような環境変数 `XVENDORLAYER` による国際化ライブラリの指定は不要です。

C.2 Motif ツールキット(Xm) および HP による Motif ツールキットの拡張 (DXm)

C.2.1 日本語の表示

Tru64 UNIX では、OSF/Motif 1.2.5 の国際化機能を使用して、Xm および DXm のウィジェットで日本語の表示、入力ができます。

言語設定が日本語の場合、`XmFileSelectionBox`、`DXmColorMix` 等のウィジェット内の省略時のラベルは自動的に日本語になります。また、ウィジェットの省略時のフォント・リストは、以下のフォントからなる日本語用のフォント・セットを持ちます。


```
-JDECW-Kmenu-Medium-R-Normal--*-120-*--P--ISO8859-1
-JDECW-Gothic-Medium-R-Normal--*-120-*--M--JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--*-120-*--M--JISX0208.1983-1
```

日本語を扱うアプリケーションの開発には、OSF/Motif 1.2.5 の `XmFontListEntryLoad()`、`XmFontListAppendEntry()` 等で作成される、フォント・セットを用いたフォント・リストや、`XmStringCreateLocalized()` 等で作成される、`XmFONTLIST_DEFAULT_TAG` を用いたコンパウンド・ストリングが使用できます。

なお、以前のバージョンとソース互換性を保つため、`DXmCvtFCtoCS()` 等の関数も動作します。これらについては、以前のバージョンの『日本語 Motif 日本語機能説明書』を参照してください。

C.2.2 日本語の入力

OSF/Motif 1.2.5 の `XmIm` の機能を使用して日本語の入力ができます。ウィジェットでは `DXmCSText` に加えて `XmText` でも日本語が入力できます。ただし、DEC 日本語入力サーバ (`dxjim`) または他の入力サーバが動作していることが必要です。`XmText` ウィジェットは `OnTheSpot` スタイルをサポートするように拡張されています。

`VendorShell` の `XmNinputMethod` に "DEC" を指定すると DEC 入力サーバが使用されます。何も指定しない場合も同様に動作します。

`VendorShell` の `XmNpreeditType` リソースは、以下の値を取ることができます。

`XmNpreeditType:` *type* [, *type*...]

type には `OnTheSpot`、`OverTheSpot`、`OffTheSpot`、`Root` のいずれかの文字列を指定します。`XmNpreeditType` の省略時の値は "OverTheSpot,OffTheSpot,Root" です。

C.2.3 ロケールの設定

OSF/Motif 1.2.5 では、フォントセットを用いた日本語の表示や、入力サーバを用いた日本語入力の機能は、`X11R6.3` の国際化機能を利用しています。そのため、それらの機能を実行するプログラムでは、ロケールが正しく設定されている必要があります。

たとえば、以下の 1 行を `XtAppInitialize()` の前に加えることによって、セッションの言語と同じ値でロケールを設定することができます。

```
XtSetLanguageProc (NULL, NULL, NULL) ;
```

C.2.4 日本語 Motif 1.1.3 との互換性

OSF/Motif 1.1.3 をもとにした以前のバージョンのウィンドウ環境では、リソース・ファイル内にコンパウンド・ストリング・リソースの値として書かれた日本語文字列は、各キャラクタセットごとのタグを持ったコンパウンド・ストリングに変換されていました。

本バージョンでは OSF/Motif 1.2.5 に従い、`XmFONTLIST_DEFAULT_TAG` を持ったコンパウンド・ストリングに変換されます。また、`XmCommand`、`XmSelectionBox`、`XmFileSelectionBox` の callback でアプリケーションに渡されるコンパウンド・ストリングも本バージョンからは、`XmFONTLIST_DEFAULT_TAG` を持ったものになります。

これらのコンパウンド・ストリングが以前のバージョンのタイプである必要がある場合には環境変数 `ADECW_MOTIF11` をセットしてください。

```
setenv ADECW_MOTIF11
```

C.2.5 DXmCSText のコンパウンド・ストリング

キーボードから `DXmCSText` に文字列を入力して、その文字列を `DXmCSTextGetString()` 関数等で取り出した場合、文字列はコンパウンド・ストリング内で、キャラクタ・セットごとのセグメントに分割されています。OSF/Motif 1.1.3 をもとにした DECwindows Motif では、そのようなコンパウンド・ストリングで日本語を扱ってきました。

OSF/Motif 1.2.5 をもとにした現在の Tru64 UNIX ウィンドウ環境では、そのようなコンパウンド・ストリングに加えて、複数のキャラクタ・セットを含んだコード・セットの文字列を `XmFONT_LIST_DEFAULT_TAG` のセグメントで取り扱うことができます。キーボードから入力した `CSText` の文字列が `XmFONT_LIST_DEFAULT_TAG` を用いたコンパウンド・ストリングで取り出されるようにするには、環境変数 `ADECW_MOTIF12` をセットしてください。

```
setenv ADECW_MOTIF12
```

C.2.6 DXmCSText の日本語入力の制御

本バージョンでは、DXmCSText の DXmNinputMethod リソースは作用しません。入力サーバの切り換え等の制御には、VendorShell の XmNinputMethod リソースを使用してください。

C.3 クライアント間の通信

日本語ウィンドウのタイトルおよびアイコン名を表示するには、WMShell の XmNtitle, XmNiconName リソースにコンパウンド・テキストで日本語文字列を与え、XmNtitleEncoding, XmNiconNameEncoding リソースに "COMPOUND_TEXT" のアトムをセットします。XmBulletinBoard ウィジェット (およびそのサブクラス) と DXmHelp ウィジェットでは、それぞれ、XmNdialogTile と XmNtitleLabel リソースにコンパウンド・ストリングで日本語文字列を与えることができます。

クリップ・ボードを使用した通信では、コンパウンド・テキストで日本語文字列を通信します。

C.4 日本語フォント

Tru64 UNIX は、以下の XLFD の日本語フォントを提供します。

C.4.1 75 dpi フォント

```
-JDECW-Gothic-Medium-R-Normal--8-80-75-75-M-40-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--10-100-75-75-M-50-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--12-120-75-75-M-60-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--14-140-75-75-M-70-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--18-180-75-75-M-90-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--24-240-75-75-M-120-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--8-80-75-75-M-80-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--10-100-75-75-M-100-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--12-120-75-75-M-120-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--14-140-75-75-M-140-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--18-180-75-75-M-180-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--24-240-75-75-M-240-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--8-80-75-75-M-40-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--10-100-75-75-M-50-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--12-120-75-75-M-60-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--14-140-75-75-M-70-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--18-180-75-75-M-90-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--24-240-75-75-M-120-JISX0201.1976-0
-JDECW-Kmenu-Medium-R-Normal--12-120-75-75-P-70-ISO8859-1
-JDECW-Kmenu-Medium-R-Normal--12-120-75-75-P-70-JISX0201.1976-0
```

-JDECW-Mincho-Medium-R-Normal--8-80-75-75-M-40-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--10-100-75-75-M-50-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--12-120-75-75-M-60-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--14-140-75-75-M-70-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--18-180-75-75-M-90-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--24-240-75-75-M-120-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--8-80-75-75-M-80-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--10-100-75-75-M-100-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--12-120-75-75-M-120-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--14-140-75-75-M-140-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--18-180-75-75-M-180-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--24-240-75-75-M-240-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--8-80-75-75-M-40-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--10-100-75-75-M-50-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--12-120-75-75-M-60-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--14-140-75-75-M-70-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--18-180-75-75-M-90-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--24-240-75-75-M-120-JISX0201.1976-0
-JDECW-Screen-Medium-R-Normal--14-140-75-75-M-70-DEC-DECsuppl
-JDECW-Screen-Medium-R-Normal--18-180-75-75-M-80-DEC-DECsuppl
-JDECW-Screen-Medium-R-Normal--24-240-75-75-M-120-DEC-DECsuppl
-JDECW-Screen-Medium-R-Normal--14-140-75-75-M-70-DEC-DECTech
-JDECW-Screen-Medium-R-Normal--18-180-75-75-M-80-DEC-DECTech
-JDECW-Screen-Medium-R-Normal--24-240-75-75-M-120-DEC-DECTech
-JDECW-Screen-Medium-R-Normal--14-140-75-75-M-70-ISO8859-1
-JDECW-Screen-Medium-R-Normal--18-180-75-75-M-80-ISO8859-1
-JDECW-Screen-Medium-R-Normal--24-240-75-75-M-120-ISO8859-1
-JDECW-Screen-Medium-R-Normal--10-100-75-75-M-100-JISX0208-Kanji00
-JDECW-Screen-Medium-R-Normal--14-140-75-75-M-140-JISX0208-Kanji00
-JDECW-Screen-Medium-R-Normal--16-160-75-75-M-160-JISX0208-Kanji00
-JDECW-Screen-Medium-R-Normal--18-180-75-75-M-160-JISX0208-Kanji00
-JDECW-Screen-Medium-R-Normal--24-240-75-75-M-240-JISX0208-Kanji00
-JDECW-Screen-Medium-R-Normal--10-100-75-75-M-100-JISX0208-Kanji11
-JDECW-Screen-Medium-R-Normal--14-140-75-75-M-140-JISX0208-Kanji11
-JDECW-Screen-Medium-R-Normal--18-180-75-75-M-160-JISX0208-Kanji11
-JDECW-Screen-Medium-R-Normal--24-240-75-75-M-240-JISX0208-Kanji11
-JDECW-Screen-Medium-R-Normal--10-100-75-75-M-50-JISX0201-RomanKana
-JDECW-Screen-Medium-R-Normal--14-140-75-75-M-70-JISX0201-RomanKana
-JDECW-Screen-Medium-R-Normal--16-160-75-75-M-80-JISX0201-RomanKana
-JDECW-Screen-Medium-R-Normal--18-180-75-75-M-80-JISX0201-RomanKana
-JDECW-Screen-Medium-R-Normal--24-240-75-75-M-120-JISX0201-RomanKana

C.4.2 100 dpi フォント

-JDECW-Gothic-Medium-R-Normal--11-80-100-100-M-55-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--14-100-100-100-M-70-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--17-120-100-100-M-85-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--20-140-100-100-M-100-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--25-180-100-100-M-125-ISO8859-1
-JDECW-Gothic-Medium-R-Normal--34-240-100-100-M-170-ISO8859-1

```

-JDECW-Gothic-Medium-R-Normal--11-80-100-100-M-110-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--14-100-100-100-M-140-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--17-120-100-100-M-170-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--20-140-100-100-M-200-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--25-180-100-100-M-250-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--34-240-100-100-M-340-JISX0208.1983-1
-JDECW-Gothic-Medium-R-Normal--11-80-100-100-M-55-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--14-100-100-100-M-70-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--17-120-100-100-M-85-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--20-140-100-100-M-100-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--25-180-100-100-M-125-JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--34-240-100-100-M-170-JISX0201.1976-0
-JDECW-Kmenu-Medium-R-Normal--17-120-100-100-P-85-ISO8859-1
-JDECW-Kmenu-Medium-R-Normal--12-120-100-100-P-85-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--11-80-100-100-M-55-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--14-100-100-100-M-70-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--17-120-100-100-M-85-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--20-140-100-100-M-100-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--25-180-100-100-M-125-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--34-240-100-100-M-170-ISO8859-1
-JDECW-Mincho-Medium-R-Normal--11-80-100-100-M-110-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--14-100-100-100-M-140-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--17-120-100-100-M-170-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--20-140-100-100-M-200-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--25-180-100-100-M-250-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--34-240-100-100-M-340-JISX0208.1983-1
-JDECW-Mincho-Medium-R-Normal--11-80-100-100-M-55-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--14-100-100-100-M-70-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--17-120-100-100-M-85-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--20-140-100-100-M-100-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--25-180-100-100-M-125-JISX0201.1976-0
-JDECW-Mincho-Medium-R-Normal--34-240-100-100-M-170-JISX0201.1976-0

```

C.5 フォント・コンバータ

フォント・コンバータとは、フォント・エディタによって作成したユーザ定義文字フォント・ファイルを **PCF** 形式のフォント・ファイルに変換するものです。これにより、ユーザ定義文字をワークステーションのディスプレイ上に表示することができるようになります。

C.5.1 フォント・ファイル変換の方法

フォント・ファイル変換には、**fontconverter** を使用します。**fontconverter** は、次のようにコマンド行から起動します。

```
% fontconverter [-オプション] -font フォント名 [-preload] プリロード・ファイル名
```

fontconverter に指定できる省略可能なオプションには次のものがあります。

-merge	このオプションを指定すると、ユーザ定義文字を指定フォントのすべての文字とマージします。このオプションを指定しない場合、指定されたフォントの省略時の文字だけをマージします。
-w width	ユーザ定義文字の幅を指定します。
-h height	ユーザ定義文字の高さを指定します。
-udc udc_name	-merge を指定しなかった場合のフォントの registry を指定します。出力されるファイル名は、{udc_name}_{width}_{height}.pcf と {udc_name}_{width}_{height}.bdf です。

次の2つの指定は省略することはできません。

-font フォント名	マージ先のフォントのフォント名を指定します。
-preload プリロード・ファイル名	ユーザ定義文字フォント・ファイル名を指定します。プリロード・ファイル名は省略できませんが、-preload は省略することができます。

C.5.2 フォント・ファイル変換の例

次の例では、プリロード・ファイル test.pre の中のユーザ定義文字を PCF/BDF ファイルに変換します。出力されるファイル名は、TEST_24_24.bdf と TEST_24_24.pcf です。

```
% fontconverter -w 24 -h 24 -udc TEST -font *mincho*24*jisx0208.1983 test.pre
```

次の例では、プリロード・ファイル test.pre の中のユーザ定義文字を "*mincho*24*jisx0208.1983" の中の文字とマージします。出力されるファイル名は、JISX0208.1983_24_24.pcf です。

```
% fontconverter -merge -font *mincho*24*jisx0208.1983 test.pre
```

索引

数字および記号

- 7-bit JIS** コード A-3
- 8-bit JIS** コード A-4
- 8-bit** データの取り扱い 2-11

A

- ASCII** 文字セット A-1

C

- C** ロケール 2-8
- charmap** B-1
- charset** フィールド 8-6
- Content-Type** ヘッダ 8-6
- CTRL** 変換キー定義 7-13

D

- DEC** 漢字コードセット A-7

E

- /etc/i18n_login** 2-3
- /etc/i18n_path** 2-3
- /etc/i18n_profile** 2-3
- EVEJ** キー 7-8, 7-10

I

- iconv** 2-12
- inc** 8-6

J

- JIS** 漢字コードセット
 - 7 単位系 A-3
 - 8 単位系 A-4
- JVMS** キー 7-8, 7-9

K

- KEYPAD** キー 7-8
- KEYPAD** 変換キー定義 7-11

L

- LC_COLLATE** B-5
- LC_CTYPE** B-2
- LC_MESSAGES** B-5
- LC_MONETARY** B-6
- LC_NUMERIC** B-6
- LC_TIME** B-7

M

- .mh_profile** 8-2, 8-4, 8-6
- MH** の操作方法 8-2

S

show 8-6
stty 2-12
 acode 2-10
 imode 2-11
 jdec 2-8
Super DEC 漢字コードセット .. A-8

T

TARO キー 7-8
TARO 変換キー定義 7-12

U

/usr/lib/mail-codesets 8-4
/usr/sbin/i18nsetup 2-4

あ

アプリケーション・コード 2-10,
 8-4, A-1
アプリケーション・コードの設
 定 2-10

か

拡張文字クラス B-2
かなモード 7-19
環境変数 **EXCODE** 8-4
環境変数 **I18NPATH** 2-3
環境変数 **LANG** 2-7
環境変数 **LC_ALL** 2-7
環境変数 **MANPATH** 2-15

環境変数 **PATH** 2-3
漢字コード A-1

き

記号変換対応表 7-18

こ

国際化環境の設定 2-3
コマンド一覧表 - **MH** 8-6
コードセット A-2
コード入力 7-19
コード変換 - メール 8-3

し

シフト **JIS** コードセット A-9
照合順序 B-5

た

端末コード 2-10, A-1
端末コードとアプリケーションコー
 ド 2-9
端末コードとアプリケーション・コー
 ドの設定 2-9
端末コードの設定 2-10
端末ドライバ 2-9
端末の設定状態の確認 2-12

と

特殊文字変換対応表 7-17

に

日本語 EUC コードセット.....	A-5
日本語コードセットの属性	B-1
日本語端末の設定	2-8
日本語入力機能	7-1
日本語入力の変換キー.....	7-8
日本語リファレンス・ページの設 定.....	2-15
日本語ロケール	B-1
日本語ロケールの設定.....	2-6

ひ

標準の文字クラス	B-2
----------------	-----

ふ

ファイル・コードの変換	2-12
プリンタ・コード	4-1
フロント・エンド	8-1

へ

変換規則	7-14
------------	------

変換キー付き日本語キーボード..	7-20
変換キー付き日本語キーボードの変換 キー定義.....	7-20
変換キー定義の変更	7-14

め

メッセージ・カタログ.....	2-14
メール.....	8-1
メール配送システム	8-1
メール変換コード	8-4

も

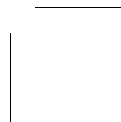
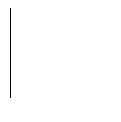
文字コード	A-1
文字セット	A-1

ゆ

有効な日本語ロケール名	2-6
-------------------	-----

ろ

ローマ字かな変換対応表	7-15
-------------------	------



Tru64 UNIX ドキュメントの購入方法

Tru64 UNIX ドキュメントのご購入については、弊社担当営業または日本ヒューレット・パッカートの各営業所/代理店にお問い合わせください。

各ドキュメント・キットの注文番号は以下のとおりです。ドキュメント・キットに含まれるマニュアルの内容については『ドキュメント概要』を参照してください。

キット名	注文番号
Tru64 UNIX Documentation CD-ROM	QA-6ADAA-G8
Tru64 UNIX Documentation Kit	QA-6ADAA-GZ
End User Documentation Kit	QA-6ADAB-GZ
- Startup Documentation Kit	QA-6ADAC-GZ
- General User Documentation Kit	QA-6ADAD-GZ
- System and Network Management Documentation Kit	QA-6ADAE-GZ
Developer's Documentation Kit	QA-6ADAF-GZ
Reference Pages Documentation Kit	QA-6ADAG-GZ
TruCluster Server Documentation Kit	QA-6BRAA-GZ
Tru64 UNIX 日本語ドキュメント・キット	QA-6ADJB-GZ
スタートアップ・ドキュメント・キット	QA-6ADJC-GZ
一般ユーザ・ドキュメント・キット	QA-6ADJD-GZ
システム/ネットワーク管理ドキュメント・キット	QA-6ADJE-GZ
プログラミング・ドキュメント・キット	QA-6ADJF-GZ
CDE 翻訳ドキュメント・キット	QA-6ADJG-GZ
TruCluster Server 日本語ドキュメント・キット	QA-05SJA-GZ
Advanced Server for UNIX 日本語ドキュメント・キット	QA-5U2JA-GZ



マニュアルに対するご意見

Tru64 UNIX
日本語機能ガイドブック
AA-RK3FE-TE

弊社のマニュアルに関して、ご意見、ご要望、または内容の不明確な部分など、お気づきの点がございましたら、下記にご記入の上、弊社社員にお渡しくださるようお願い申し上げます。

マニュアルの採点：

	大変良い	良い	普通	良くない
正確さ(説明どおりに動作するか)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
情報量(十分か)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
分かり易さ	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
マニュアルの構成	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
図(役立つか)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
例(役立つか)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
索引(項目の検索性)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ページ・レイアウト(情報の検索性)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

内容の不明確な部分がありましたら、以下にご記入ください：

ペー ジ

その他お気づきの点がございましたら、以下にご記入ください：

ご使用のソフトウェアのバージョン： _____

貴社名/部課名 _____

御名前 _____

記入日 _____

(注) 当用紙を受け取った弊社社員は、すみやかに下記にお送りください。

ビジネスクリティカルシステム統括本部 **BCS** 技術本部 **Alpha** ソフトウェア技術部