



# 国際化対応言語環境の利用ガイド

---

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

Part No: 819-0400-10  
2005 年 1 月

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

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

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

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

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

Sun、Sun Microsystems、docs.sun.com、AnswerBook、AnswerBook2、および SunOS は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標、登録商標もしくは、サービスマークです。

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

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。Netscape Navigator は、Netscape Communications Corporation の商標です。PostScript は、米国 Adobe Systems, Inc. の商標であり、国によっては登録されていることがあります。

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

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

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

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

「ATOK Server/ATOK12」に含まれる郵便番号辞書 (7 桁/5 桁) は日本郵政公社が公開したデータを元に制作された物です (一部データの加工を行っています)。

「ATOK Server/ATOK12」に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド』に添付のものを使用しています。

Unicode は、Unicode, Inc. の商標です。

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

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

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

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

原典: *International Language Environments Guide*

Part No: 817-2521-10

Revision A



050106@10536



# 目次

---

はじめに	15
<b>1 Solaris 国際化の概要</b>	<b>21</b>
Solaris 国際化のアーキテクチャについて	21
新しい国際化機能と地域対応機能	22
国際化と地域対応の概要	25
国際化の基本的な手順	25
Solaris インタフェースの地域対応機能	27
ロケールとは	27
C ロケール – デフォルトのロケール	28
完全ロケールと部分ロケール	29
ロケールによって影響を受ける動作	29
ロケールのカテゴリ	30
ロケールのカテゴリを使用した地域対応	30
時間の形式	31
日付の形式	31
数値の形式	32
国際的な通貨の形式	33
語と文字の相違点	35
語の区切り	35
ソートの順序	35
文字セット	35
キーボードの相違点	38
用紙サイズの相違点	38

2	一般的な国際化機能	41
	コードセットの独立性のサポート	41
	CSI のアプローチ	42
	CSI 対応のコマンド	42
	CSI 対応のライブラリ	43
	ロケールデータベース	43
	プロセスコードの書式 (ワイド文字表現)	44
	複数バイトサポート環境	44
	動的にリンクされたアプリケーション	45
	変更されたインタフェースについて	45
	ctype マクロ	46
	libc の国際化 API	47
	genmsg ユーティリティ	54
	ユーザーによるコード変換の定義と拡張	55
	国際化ドメイン名 (IDN) のサポート	56
3	<b>Solaris</b> 環境における地域対応	59
	地域対応のためのソフトウェアサポート	59
	Solaris ロケールパッケージの要約	59
	サポートされるロケール	60
	ロケールの複数キーの Compose キーシーケンス	67
	Solaris 環境でのキーボードサポート	68
	SPARC システムでのキーボードの変更	70
	Intel システムでのキーボードの変更	73
	キーボードの配列	73
	新しい Solaris キーボードのソフトウェアサポート	81
	▼ エストニア語 Type 6 USB キーボードサポートの使用法	81
	▼ フランス語系カナダ Type 6 USB キーボードサポートの使用法	81
	▼ ポーランド語プログラマ Type 5 キーボードサポートの使用法	82
4	サポートされるアジアのロケール	83
	日本語の地域対応	83
	日本語ロケール	83
	日本語文字セット	83
	日本語フォント	84
	日本語入力方式	85
	▼ ATOK 入力方式の使用法	85

日本語端末用の端末設定	86
日本語 iconv モジュール	86
ユーザー定義文字のサポート	86
部分ロケールと完全ロケールの違い	86
インド語の地域対応	87
▼インド語入力方式の使用法	87
インド語キーボード	88
マッピングについて	91
連続表音ベース入力方式のマッピング	92
連続表音入力方式の仕組み	113
タイ語の地域対応	114
タイ語入力方式	114
タイ語のキー配列	114
タイ語入力方式の補助ウィンドウ	116
<b>5 UTF-8 ロケールサポートの概要</b>	<b>117</b>
Unicode の概要	117
Unicode ロケール: en_US.UTF-8 サポート	118
デスクトップ入力方式について	120
スクリプトの選択と入力方式	121
入力方式の選択	121
入力モードの切り替えキーシーケンス	122
英語/ヨーロッパ言語入力モード	123
アラビア語入力モード	139
キリル文字入力モード	139
ギリシャ語入力モード	140
ヘブライ語入力モード	149
日本語入力モード	150
韓国語入力モード	151
簡体字中国語入力モード	151
繁体字中国語入力モード	151
繁体字中国語 (香港) 入力モード	152
Unicode 16 進数入力モード	152
表検索入力モード	153
システム環境	153
ロケールの環境変数	153
TTY 環境の設定	154
コード変換	158

DtMail サポート	158
プログラミング環境	161
X アプリケーションで使用されるフォントセット	162
CDE/Motif アプリケーションとしての FontList 定義	162
<b>6 CTL: Complex Text Layout</b>	<b>163</b>
CTL の概要	163
CTL アーキテクチャの概要	164
X ベースのアプリケーションに対する CTL サポート	164
XOC リソース	165
CTL 技術をサポートするための Motif の変更点	165
XmNlayoutDirection リソース	166
XmStringDirection リソース	167
XmRendition リソース	167
XmText と XmTextField リソース	169
XmTextFieldGetLayoutModifier リソース	173
XmTextGetLayoutModifier リソース	173
XmTextFieldSetLayoutModifier リソース	174
XmTextSetLayoutModifier リソース	174
XmStringDirectionCreate リソース	175
UIL 引数	175
CTL アプリケーションの開発	176
レイアウト方向の制御	176
リソースファイルのレンダータブルの作成	179
水平タブ	180
マウスによる選択	181
キーボードによる選択	182
テキストのリソースおよびジオメトリ	182
移植に関する注意事項	183
<b>7 mp による印刷フィルタの拡張</b>	<b>185</b>
UTF-8 用の印刷	185
mp 印刷フィルタの拡張概要	186
mp でロケール固有のフォント構成ファイル mp.conf を使用する場合	187
mp でロケール固有の PostScript プロログファイルを使用する場合	187
mp が Xprt (X Print Server) クライアントとして動作する場合	187
mp.conf 構成ファイルを使用した地域対応	188

▼ プリンタ常駐フォントの追加方法	192
▼ 共有オブジェクトファイルの作成方法	194
prolog ファイルの追加とカスタマイズ	194
PostScript ファイルのカスタマイズ	195
.xpr ファイル	198

**A iconv コード変換 203**

索引	231
----	-----



# 表目次

---

表 1-1	正当な UTF-8 バイトシーケンス	24
表 1-2	国際的な時間の表記形式	31
表 1-3	国際的な日付の表記形式	31
表 1-4	国際的な数値の表記形式	32
表 1-5	国際的な通貨の表記規則	33
表 1-6	ユーロ通貨をサポートするユーザーロケール	34
表 1-7	ドイツ語のロケールとその LC_MONETARY オペランド	34
表 1-8	一般的な国際ページサイズ	38
表 2-1	libc のメッセージング関数	47
表 2-2	libc のコード変換関数	48
表 2-3	libc の正規表現関数	48
表 2-4	libc のワイド文字クラス	48
表 2-5	libc のロケールの変更および照会関数	48
表 2-6	libc のロケールデータ照会関数	49
表 2-7	libc の文字分類および変換関数	49
表 2-8	libc の文字照合関数	50
表 2-9	libc の通貨処理関数	50
表 2-10	libc の日付と時間の処理関数	51
表 2-11	libc の複数バイト処理関数	51
表 2-12	libc のワイド文字および文字処理	51
表 2-13	libc のワイド文字入力および出力の書式化関数	52
表 2-14	libc のワイド文字列関数	53
表 2-15	libc のワイド文字入力および出力関数	53
表 2-16	iconv コード変換	57
表 3-1	アジアのロケール	60
表 3-2	オーストラリアのロケール	62
表 3-3	中央アメリカのロケール	62

表 3-4	中央ヨーロッパのロケール	62
表 3-5	東ヨーロッパのロケール	63
表 3-6	中東のロケール	64
表 3-7	北アフリカのロケール	64
表 3-8	北アメリカのロケール	64
表 3-9	北ヨーロッパのロケール	65
表 3-10	南アメリカのロケール	65
表 3-11	南ヨーロッパのロケール	66
表 3-12	西ヨーロッパのロケール	66
表 3-13	Compose キーで入力する発音記号の例	67
表 3-14	地域向けキーボードのサポート	68
表 3-15	Type 4、5、5c キーボードの配列	70
表 4-1	日本語ビットマップフォント	84
表 4-2	日本語 TrueType フォント	85
表 5-1	入力モードの切り替えキーシーケンス	123
表 5-2	Latin-1 の一般的な Compose キーシーケンス	124
表 5-3	Latin-2 の一般的な Compose キーシーケンス	128
表 5-4	Latin-3 の一般的な Compose キーシーケンス	129
表 5-5	Latin-4 の一般的な Compose キーシーケンス	130
表 5-6	Latin-5 の一般的な Compose キーシーケンス	131
表 5-7	Latin-9 の一般的な Compose キーシーケンス	132
表 5-8	アクセントデッドキーに基づく Compose キーシーケンス	132
表 5-9	ギリシャ語入力モードの Compose キーシーケンス	141
表 5-10	3つのキーを使用するギリシャ語入力モードの Compose キーシーケンス	147
表 5-11	4つのキーを使用するギリシャ語入力モードの Compose キーシーケンス	149
表 5-12	en_US.UTF-8 によりサポートされる STREAMS モジュール	154
表 5-13	en_US.UTF-8 によりサポートされる 64 ビット STREAMS モジュール	154
表 6-1	XmRendition の新しいリソース	167
表 6-2	Xm CTL の新しいリソース	169
表 6-3	UIL	175
表 7-1	キーワード/値ペア (オプション)	192
表 7-2	STARTCOMMON/ENDCOMMON キーワード値	201
表 A-1	使用できる Unicode 関連 iconv コード変換モジュール	203
表 A-2	使用できる Unicode、IBM/Microsoft EBCDIC、PC のコードページ関連 iconv コード変換モジュール	225
表 A-3	使用できる iconv コード変換モジュール - IBM および Microsoft	





# 図目次

---

図 1-1	Solaris OS におけるロケールの機能と構造	26
図 2-1	IDN から ACE への変換	56
図 2-2	ACE から IDN への変換	56
図 3-1	アラビア語キーボード	73
図 3-2	ベルギー向けキーボード	73
図 3-3	キリル文字 (ロシア語) キーボード	73
図 3-4	デンマーク語キーボード	74
図 3-5	フィンランド語キーボード	74
図 3-6	フランス語キーボード	74
図 3-7	ドイツ語キーボード	75
図 3-8	イタリア語キーボード	75
図 3-9	日本語キーボード	75
図 3-10	韓国語キーボード	76
図 3-11	オランダ (オランダ語) キーボード	76
図 3-12	ノルウェー語キーボード	76
図 3-13	ポルトガル語キーボード	77
図 3-14	スペイン語キーボード	77
図 3-15	スウェーデン語キーボード	77
図 3-16	スイス (フランス語) キーボード	78
図 3-17	スイス (ドイツ語) キーボード	78
図 3-18	繁体字中国語キーボード	78
図 3-19	トルコ語 F キーボード	79
図 3-20	トルコ語 Q キーボード	79
図 3-21	イギリス向けキーボード	79
図 3-22	米国向けキーボード	80
図 3-23	米国/UNIX キーボード	80
図 4-1	Bengali の子音のマッピング	92

☒ 4-2	Bengali の母音のマップ	92
☒ 4-3	Bengali のその他のマップ	93
☒ 4-4	Gujarati の子音のマップ	94
☒ 4-5	Gujarati の母音のマップ	95
☒ 4-6	Gujarati のその他のマップ	96
☒ 4-7	Gurmukhi の子音のマップ	97
☒ 4-8	Gurmukhi の母音のマップ	98
☒ 4-9	Gurmukhi のその他のマップ	99
☒ 4-10	Hindi の子音のマップ	100
☒ 4-11	Hindi の母音のマップ	100
☒ 4-12	Hindi のその他のマップ	101
☒ 4-13	Kannada の子音のマップ	102
☒ 4-14	Kannada の母音のマップ	103
☒ 4-15	Kannada のその他のマップ	104
☒ 4-16	Malayalam の子音のマップ	105
☒ 4-17	Malayalam の母音のマップ	106
☒ 4-18	Malayalam のその他のマップ	107
☒ 4-19	Tamil の子音のマップ	108
☒ 4-20	Tamil の母音のマップ	109
☒ 4-21	Telugu の子音のマップ	111
☒ 4-22	Telugu の母音のマップ	111
☒ 4-23	Telugu のその他のマップ	112
☒ 5-1	入力モード選択ウィンドウ	121
☒ 5-2	アラビア語 キーボード	139
☒ 5-3	キリル文字 (ロシア語) キーボード	140
☒ 5-4	ギリシャ語キーボード (ヨーロッパキーボード)	140
☒ 5-5	ギリシャ語キーボード (UNIX キーボード)	141
☒ 5-6	ヘブライ語キーボード	149
☒ 5-7	日本語キーボード	150
☒ 5-8	韓国語キーボード	151
☒ 5-9	DtMail 「新規メッセージ」 ウィンドウ	160
☒ 6-1	CTL アーキテクチャ	164
☒ 6-2	レイアウト方向	176
☒ 6-3	タブ動作	181

## はじめに

---

この『国際化対応言語環境の利用ガイド』は、Solaris™ オペレーティングシステム (Solaris OS) の新しい国際化機能について説明します。このマニュアルには、Solaris の最新リリースを使用してさまざまな言語や文化的な慣行をサポートするグローバルソフトウェア製品を作成する方法についての、重要な情報が含まれています。

さらに、このリリースの国際化機能に関するその他の情報が記載されているほかのマニュアルもこのマニュアルで示されています。

---

注 - このマニュアルの中のオペレーティングシステムに関するすべての情報は Solaris OS に基づいています。

---

この章の内容は次のとおりです。

- 16 ページの「対象読者」
- 16 ページの「内容の紹介」
- 16 ページの「関連マニュアル」
- 17 ページの「Sun のオンラインマニュアル」
- 17 ページの「表記上の規則」
- 19 ページの「一般規則」

---

注 - このリリースでは、SPARC® および x86 系列のプロセッサアーキテクチャ (UltraSPARC®, SPARC64, AMD64, Pentium, Xeon EM64T) を使用するシステムをサポートします。サポートされるシステムについては、*Solaris 10 Hardware Compatibility List* (<http://www.sun.com/bigadmin/hcl>) を参照してください。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書では、「x86」という用語は AMD64 あるいは Intel Xeon/Pentium 製品系列と互換性のあるプロセッサを使用して製造された 32 ビットおよび 64 ビットシステムを意味します。サポートされるシステムについては、*Solaris 10 Hardware Compatibility List* を参照してください。

---

---

## 対象読者

このマニュアルは、最新の Solaris OS でグローバルアプリケーションを設計およびサポートするソフトウェア開発者およびシステム管理者を対象にしています。

C プログラミング言語の実践的な知識があることを前提としています。

---

## 内容の紹介

このマニュアルは次のように構成されています。

- 第 1 章では、Solaris の最新リリースで使用できる新しい国際化機能および地域対応機能について説明します。
- 第 2 章では、コードセットの独立性 (CSI)、ロケールデータベース、libc の API、およびその他の国際化機能についての基本について説明します。
- 第 3 章では、最新の Solaris OS での使用がサポートされているロケール、フォント、およびキーボードについて説明します。
- 第 4 章では、Solaris の最新リリースの日本語、ヒンディー語、およびタイ語の地域対応サポートについて説明します。
- 第 5 章では、最新の Solaris OS での使用がサポートされている入力方式およびコード変換機能について説明します。
- 第 6 章では、Complex Text Layout (CTL) 拡張機能について説明します。この機能は、論理テキスト表現と物理テキスト表現の間で複雑な変換が必要な表記体系をサポートする Motif の API を有効にします。複雑な変換が必要な表記体系には、アラビア語、ヘブライ語、およびタイ語があります。
- 第 7 章では、mp 印刷フィルタを中心に印刷サポートについて説明します。
- 付録 A では、iconv 変換モジュールの表を示します。

---

## 関連マニュアル

このマニュアルの内容に関連するその他の情報が、以下のマニュアルで説明されています。

- Solaris 国際化:  
Tuthill, Bill, and David Smallberg 『*Creating Worldwide Software: Solaris International Developer's Guide*』 2nd edition. Mountain View, California, Sun Microsystems Press, 1997 (このマニュアルの入手先は <http://www.sun.com/books/>。Solaris

OS の国際化の処理について全般的に説明している)

- 中国語および韓国語の Solaris ロケール:

『*Korean Solaris User's Guide*』

『*Simplified Chinese Solaris User's Guide*』

『*Traditional Chinese Solaris User's Guide*』

- OSF/Motif アプリケーション開発:

『*OSF/Motif Programmer's Guide, Release 1.2*』 Englewood Cliffs, New Jersey, Prentice-Hall, 1993 (Motif アプリケーションを作成するための OSF/Motif アプリケーションプログラミングインタフェースの使い方についての Open Software Foundation (OSF) のマニュアル)

---

## Sun のオンラインマニュアル

docs.sun.com では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。URL は、<http://docs.sun.com> です。

---

## 表記上の規則

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

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	<code>.login</code> ファイルを編集します。 <code>ls -a</code> を使用してすべてのファイルを表示します。 <code>system%</code>
<b>AaBbCc123</b>	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	<code>system% <b>su</b></code> <code>password:</code>
<i>AaBbCc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、 <code>rm filename</code> と入力します。

表 P-1 表記上の規則 (続き)

字体または記号	意味	例
『』	参照する書名を示します。	『コードマネージャ・ユーザズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第5章「衝突の回避」を参照してください。  この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% <b>grep</b> `^#define \ XV_VERSION_STRING`

コード例は次のように表示されます。

■ C シェル

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

■ C シェルのスーパーユーザー

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

■ Bourne シェルおよび Korn シェル

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

■ Bourne シェルおよび Korn シェルのスーパーユーザー

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

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

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

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

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

---

## 一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が、適宜併記されています。



## 第 1 章

---

# Solaris 国際化の概要

---

この章では、Solaris の国際化と地域対応に関する新機能および重要概念について説明します。この章の内容は次のとおりです。

- 22 ページの「新しい国際化機能と地域対応機能」
- 25 ページの「国際化と地域対応の概要」
- 27 ページの「ロケールとは」
- 30 ページの「ロケールのカテゴリを使用した地域対応」
- 35 ページの「語と文字の相違点」
- 38 ページの「キーボードの相違点」
- 38 ページの「用紙サイズの相違点」

---

## Solaris 国際化のアーキテクチャについて

Solaris の最新リリースには、UTF-8 ロケールに対する Unicode 4.0 サポート、拡張キーボードサポート、mp 印刷フィルタの改良などの多数の新機能が含まれます。

Solaris 国際化アーキテクチャは、全世界におけるアプリケーションや言語サービスの開発、導入、および管理を容易にします。1つの多言語対応製品によって 39 の異なる言語と 162 のロケールがサポートされます。さらに、タイ語やヒンディー語のスク립トに必要な複雑なテキストレイアウトに対するサポートも行われます。アラビア語、ヘブライ語などに対する双方向テキスト機能もサポートされています。

多数の Solaris ロケールに対して、入力方式、文字セット、コードセット変換、およびその他の言語関連機能がサポートされています。標準 API に従うことによって、アプリケーションを複数の言語環境に導入できます。言語属性のカスタマイズ、コンバータテーブルの変更、新しい入力方式エディタの Solaris 環境への追加も可能です。

2000 年秋、Solaris X グローバル化フレームワークのソースコードがオープンソースコミュニティにリリースされました。このリリースに基づいて、共通の参照実装に従うことによって、グローバルアプリケーションの互換性および相互運用性を強化できま

す。グローバル化のためにコードセットを独立させたため、自国語と Unicode ロケールのどちらでも運用が可能です。Solaris フレームワークには、プラットフォームを越えた拡張性があります。豊富なデータコンバータにより、さまざまなエンコーディングやサン以外のプラットフォームにおける相互運用性が実現されます。

さらに Solaris プラットフォームでは、多国籍企業がサーバーの管理を世界レベルで運用できます。他社のプラットフォームとは異なり、Solaris プラットフォームでは、サービスベースのアプローチを使って言語サービスを管理します。サーバー管理者は、世界的なネットワークからリモートで言語サービスを有効にすることができます。クライアントシステムが何であるかは関係ありません。クライアントから独立したこのアプローチでは、クライアントアプリケーションを変更せずにシステムをアップグレードできます。たとえば、パリのインターネットカフェから送信されたアラビア語の電子メールを読むために、ローカルのクライアントアプリケーションを変更する必要がありません。

---

## 新しい国際化機能と地域対応機能

Solaris の最新リリースで使用できる新機能は次のとおりです。

- エンコーディング自動検出

エンコーディング自動検出はグローバルな文字処理のためのユーティリティです。汎用のインタフェースを介して、特定のファイルまたは文字列のエンコーディングを簡単に検出できます。エンコーディングの検出によって、さまざまな言語の文字エンコーディングを簡単に使用できます。詳細については、`auto_ef(1)` または `libauto_ef(3LIB)` のマニュアルページを参照してください。

- ロケール管理

ロケール管理では、コマンド行インタフェースを介して Solaris OS のロケールを照会および設定できます。`localeadm(1M)` ツールを使用すると、システムにインストールされているロケールパッケージおよび特定のデバイスやディレクトリにあるロケールパッケージに関する情報を表示できます。地域単位で、現在のシステムにロケールを追加したり、ロケールを削除したりできます。たとえば、東ヨーロッパ地域のすべてのロケールを現在のシステムに追加できます。

ロケール管理が導入される以前では、システムのインストール後は、マシンのロケールを変更するために個々のパッケージを追加または削除する必要がありました。パッケージを個別に操作するので、パッケージを見つけれないなどの間違いを犯しやすくなっていました。

ロケール管理は、Solaris インストールプログラムのロケール選択ロジックを補完するものです。Solaris のロケールを正しくインストールするための主要アプリケーションは、これまでどおりインストールプログラムです。

- mp の拡張

最初に Solaris 9 OS でリリースされた mp 印刷フィルタは、Solaris の最新リリースでは `xutops` 印刷フィルタを置き換えています。拡張された mp 印刷フィルタの主な改良点は次のとおりです。

- mp.conf ファイルに設定済みフォントがない場合、認められないフォントを使用するグリフが出現するまで mp プログラムは実行を続ける
- TrueType グリフを印刷する場合の mp 出力のサイズを大幅に削減
- mp に使用される辞書メカニズムの微調整により、より高速な印刷が可能
- TrueType エンジンの拡張により、さまざまな幅のすべてのスペース文字の処理が可能

---

注 - xutops 印刷フィルタは Solaris OS ではサポートされません。これまで、国際化テキストを UTF-8 ロケールで印刷するために xutops 印刷フィルタが使用されていました。xutops に代わる mp 印刷フィルタは、xutops 印刷フィルタのサポート機能のスーパーセットです。詳細については、mp (1) のマニュアルページを参照してください。

---

- 新しいヨーロッパ向けキーボードのサポート

ポーランド語プログラマ向け Type 5 キーボードおよび Sun Ray™ USB Type 6 のロシア語、エストニア語、フランス語系カナダ向けキーボードで、サンの入出力キーボードサポートを使用できます。

---

注 - 現在、新しいヨーロッパ向けキーボードタイプ用のハードウェアはありません。新しいキーボードソフトウェアを使用するには、81 ページの「新しい Solaris キーボードのソフトウェアサポート」の手順を参照してください。

---

- Unicode 4.0 のサポート

次の UTF-8 ロケールが更新されて、新しい Unicode 4.0 バージョンの規格をサポートします。

- ar\_EG.UTF-8
- de\_DE.UTF-8
- en\_US.UTF-8
- es\_ES.UTF-8
- fi\_FI.UTF-8
- fr\_BE.UTF-8
- fr\_FR.UTF-8
- he\_IL.UTF-8
- hi\_IN.UTF-8
- it\_IT.UTF-8
- ja\_JP.UTF-8
- ko\_KR.UTF-8
- pl\_PL.UTF-8
- pt\_BR.UTF-8
- ru\_RU.UTF-8
- sv\_SE.UTF-8

- th\_TH.UTF-8
- tr\_TR.UTF-8
- zh\_CN.UTF-8
- zh\_HK.UTF-8
- zh\_TW.UTF-8

新しいバージョンの規格では、1,226 文字が新たに追加され、規格および情報に関してさまざまな変更が行われています。

Unicode 3.2 は、次のように「UTF-8 Corrigendum」に従って UTF-8 バイトシーケンスをより厳密に定義します。

表 1-1 正当な UTF-8 バイトシーケンス

コードポイント	1 番目のバイト	2 番目のバイト	3 番目のバイト	4 番目のバイト
U+0000..U+007F	00..7F			
U+0080..U+07FF	C2..DF	80..BF		
U+0800..U+0FFF	E0	A0..BF	80..BF	
U+1000..U+CFFF	E1..EC	80..BF	80..BF	
U+D000..U+D7FF	ED	80..9F	80..BF	
U+D800..U+DFFF	ill-formed			
U+E000..U+FFFF	EE..EF	80..BF	80..BF	
U+10000..U+3FFFF	F0	90..BF	80..BF	80..BF
U+40000..U+FFFFF	F1..F3	80..BF	80..BF	80..BF
U+100000..U+10FFFF	F4	80..8F	80..BF	80..BF

これらのシーケンスでは、U+D800 と U+DFFF 間のサロゲートコードポイントが除外されます。ほかの不正なバイト値も禁止されます。新しい定義に準拠するために、Unicode のロケール方式および UTF-8 の iconv モジュールを拡張したことにより、新たに定義された UTF-8 の不正なバイトシーケンスが検出されます。

■ タイ語キー配列

タイ語の入力方式に次のキー配列がサポートされます。

- Kedmanee (TIS820-2531) キー配列。Kedmanee 配列は、コンピュータキーボードとしてではなくタイプライタ用に設計されました。数に制限があるタイプライタキーボードのキーでは、タイ語のいくつかの特殊文字が配列上にありません。TIS820-2531 は、コンピュータキーボードとして使用するため Kedmanee 配列を採用しました。
- TIS820-2538 キー配列。この拡張 Kedmanee 配列は TIS820-2531 配列の更新バージョンであり、元の Kedmanee 配列にはなかったタイ語の特殊文字が含まれます。現在、TIS820-2538 はタイ工業標準研究所が定める唯一のタイ語キー配列規格です。

- Pattajoti キー配列。Pattajoti 配列もタイプライタ用に設計されたものですが、指をより使いやすいうようにキーが配列されています。Pattajoti はタイ灌漑局の職員によって考案され、現在も同局で広く使用されています。
- タイ語入力方式に合わせて設定可能な、ユーザー定義のキー配列。
- インド語の入力方式のサポート  
このリリースでは、中国語の入力方式に使用されるインタフェースと同様のコードテーブル入力方式インタフェースを使用できます。IIIMF SDK および SunIM 言語インタフェースに基づくインド語入力方式の新しい機能は次のとおりです。
  - 表音、翻字ベースの入力方式とキー配列。サポートされるキー配列は、INSCRIPT キーボードがオーバーレイする ISCI 規格で定義される
  - 標準入力方式の切り替え
  - インド語スクリプト (ヒンディー語、タミル語、カナラ語、マラヤーラム語、テルグ語、グジャラート語、パンジャブ語、ベンガル語)。F5 キーを押して入力スクリプトを変更する
  - 将来の拡張を簡単に行える、新しいキー配列のプラグインメカニズム。プラグイン設定ファイルは言語エンジンモジュールによってロードされる

---

## 国際化と地域対応の概要

国際化と地域対応の手順は異なります。国際化とは異なる言語や地域の間でソフトウェアを移植可能にする処理を指し、地域対応とは特定の言語や地域に合わせてソフトウェアを適応させる処理を指します。国際化されたソフトウェアの開発には、特定の文化的要件に応じて実行時にプログラムの動作を変えるインタフェースを使用します。地域対応には、「ロケール」と呼ばれる言語や地域をサポートするためのオンライン情報を設定する処理が含まれます。

異なる言語や慣習に従って動作させるために完全に書き直す必要のあるソフトウェアとは異なり、国際化されたソフトウェアはそのような作業を必要としません。ソフトウェアが国際化されていれば、ロケールを入れ替える場合でもソフトウェアを変更する必要はありません。Solaris システムは国際化に対応しており、国際化ソフトウェアを作成する際に必要な基本構造やインタフェースを備えています。

### 国際化の基本的な手順

国際化されたアプリケーションの実行可能イメージは、異なる言語や地域の間で共有することができます。ソフトウェアを国際化する手順を次に示します。

- このマニュアルに記載されているインタフェースを使用して、動的な再コンパイルによって変更されるソフトウェアを 1 つの環境で作成します。
- ソフトウェアを、実行可能コードと、ユーザーに表示されるすべてのメッセージに分離します。メッセージ文字列はメッセージカタログに格納します。

メッセージ文字列は、特定の言語や地域向けに翻訳されます。ロケールには、メッセージ文字列やソートで使用する手続きなどが含まれます。

ある製品の地域対応されたバージョンを使用する場合、ユーザーは特定の環境変数を設定します。これにより、メッセージがその言語のロケールに翻訳されて表示されます。日付、時刻、通貨、およびその他の情報は、ロケール固有の規定に従ってフォーマットおよび表示されます。メッセージ翻訳とオンラインヘルプの内容は、次の図で示すさまざまな層を通して提供されます。

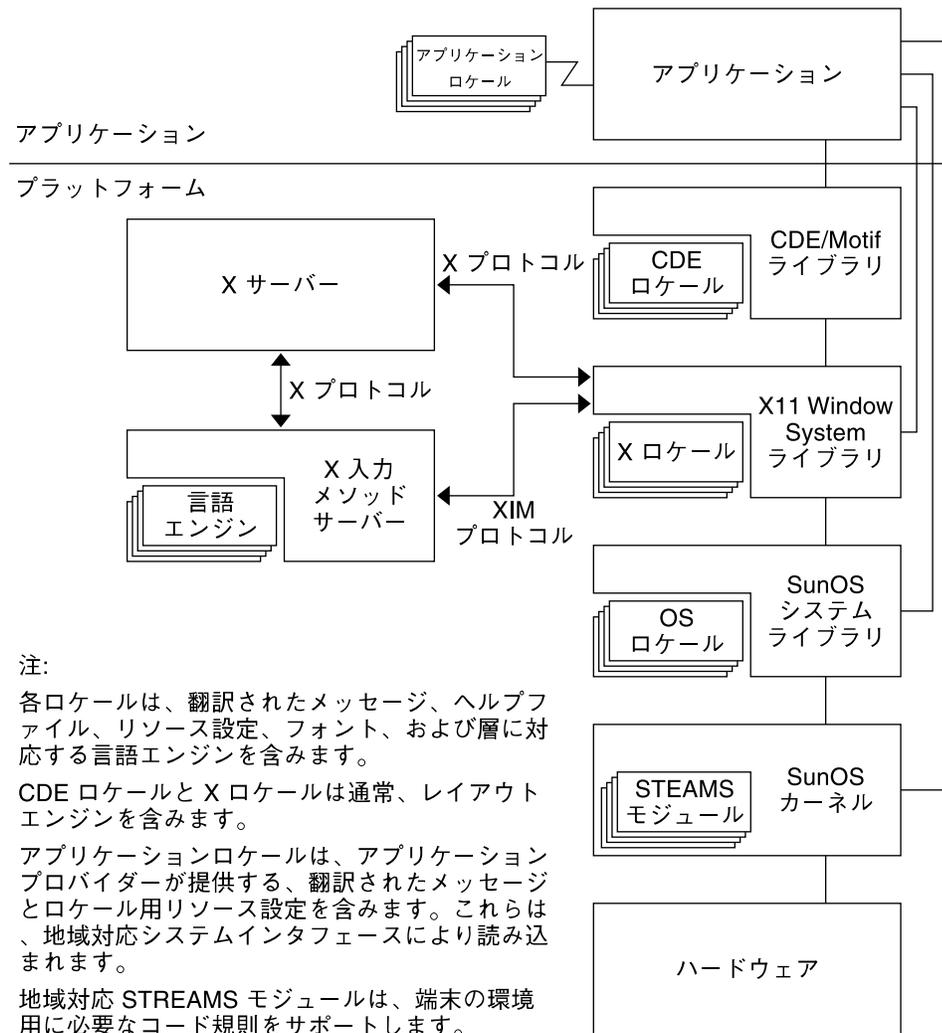


図 1-1 Solaris OS におけるロケールの機能と構造

## Solaris インタフェースの地域対応機能

OS (オペレーティングシステム) のロケール層によって、アプリケーションの実行時に OS のシステムインタフェースに差し込まれる基本的なロケールデータベースと機能が実現します。アプリケーションでは、標準的な API を通じてこれらの OS ロケールモジュールにアクセスします。

X11 ロケール層によって、X11 アプリケーションがローカルのテキスト入力と表示ができるように、X 入力方式と X 出力方式へのインタフェースを実現します。アプリケーションがさまざまな言語の文字を表示できるように各種のフォントが用意されています。

CDE/Motif は、X11 ウィンドウシステムの上に構築されています。したがって、CDE/Motif は、X11 API を使って X11 ロケールの機能を使用できます。目標のロケール内部でデスクトップを機能させるため、Solaris の地域対応には、CDE アプリケーションに関するさまざまなロケール固有の設定があります。翻訳されたメッセージとオンラインヘルプの内容は、さまざまな層で提供されます。

---

## ロケールとは

アプリケーションプログラムにとって主要な概念となるのは、プログラムのロケールの概念です。ロケールとは、本国語環境の明示的なモデルであり、定義です。ロケールの概念は、ANSI C 言語規格のライブラリ定義で明示的に定義され、組み込まれています。

ロケールは、国ごとの書式やその他の指定に関する多くのカテゴリから構成されています。プログラムのロケールは、コードセット、日付と時間の形式の規定、通貨の規定、10 進数の書式の規定、照合 (ソート) 順を定義します。

ロケールは、基本言語、使用国 (地域)、オプションのコードセットから構成されています。コードセットは、通常、言語に応じて自動的に選択されます。たとえばドイツ語は Deutsch の略語である de です。一方、スイス系ドイツ語は de\_CH です。CH は Confederation Helvetica の略語です。この規定によって、通貨単位の指定のような国ごとに固有の差異に対応することができます。

言語によっては複数のロケールを持つものもあり、地域によって異なることがあります。たとえば、アメリカ英語を使う場合は en\_US ロケール (アメリカ英語) を、イギリス英語を使う場合は en\_GB (イギリス英語) を選択できます。

通常、ロケール名は環境変数 LANG で指定します。ロケールのカテゴリは LANG に依存しますが、カテゴリ別に設定することも可能です。この場合、そのロケールのカテゴリは LANG の設定に優先します。LC\_ALL オペランドを設定すると、LANG とすべての個別ロケールカテゴリに優先します。

ロケールの命名規則は次のとおりです。

`language[_territory][.codeset] [@modifier]`

2 文字の *language* コードは ISO 639 に、2 文字の *territory* コードは ISO 3166 にそれぞれ準拠します。*codeset* は、ロケールで使用されているコードセットの名前です。*modifier* は、このロケールと、修飾子のないロケールを区別する特徴を表す名前です。

Solaris 製品の全ロケールには、US-ASCII コード値を持つ移植可能な文字セット (Portable Character Set) が含まれています。

移植可能な文字セットについては、『X/Open CAE Specification: System Interface Definitions, Issue 5 (ISBN 1-85912-186-1)』を参照してください。

単一のロケールが複数のロケール名を持つ場合があります。たとえば、POSIX は C と同じロケールです。

## C ロケール – デフォルトのロケール

C ロケール (POSIX ロケール) は、すべての POSIX 互換システムの POSIX システムデフォルトロケールです。Solaris OS は POSIX システムです。Single UNIX Specification, Version 3 によって C ロケールが定義されています。この仕様を閲覧およびダウンロードするには、<http://www.unix.org/version3/online.html> で登録が必要です。

作成した国際化プログラムを C ロケールで実行するように指定するには、次の 2 つの方法があります。

- すべてのロケール環境変数を設定解除する

```
system% unsetenv LC_ALL LANG LC_CTYPE LC_COLLATE LC_NUMERIC \  
LC_TIME LC_MONETARY LC_MESSAGES
```

これにより、すべてのロケール環境変数が設定解除され、アプリケーションが C ロケールで実行されます。

- ロケールを明示的に C または POSIX に設定する

```
system% setenv LC_ALL C  
system% setenv LANG C
```

アプリケーションによっては、現在のロケールを参照するために実際には `setlocale(3C)` を呼び出さずに `LANG` 環境変数をチェックします。この場合、`setenv` で、`LC_ALL` と `LANG` のロケール環境変数を指定することによって明示的に C ロケールを設定します。ロケール環境変数の優先関係については、`setlocale(3C)` のマニュアルページを参照してください。

端末環境の現在のロケール設定をチェックするには、`locale(1)` コマンドを実行します。

```
system% locale
```

## 完全ロケールと部分ロケール

Solaris の完全ロケールは、ロケールとしてのすべての機能を備え、言語に対応した翻訳メッセージを含んでいます。部分ロケールは、言語に対応した翻訳メッセージを含んでいません。Solaris 環境のすべてのロケールは翻訳メッセージを表示できます。ただし、その言語に対応する翻訳メッセージがインストールされている必要があります。たとえば、次のロケールは部分ロケールにも完全ロケールにもなります。

- de\_DE.ISO8859-1
- de\_DE.ISO8859-15
- de\_DE.UTF-8
- de\_AT.ISO8859-1
- de\_AT.ISO8859-15
- de\_CH.ISO8859-1

LANGUAGES CD からドイツ語のメッセージがインストールされている場合、上記のすべてのロケールは、完全に翻訳されたデスクトップにアクセスするので、完全ロケールになります。この LANGUAGES CD には、以下の言語に対する翻訳メッセージが入っています。

- ドイツ語
- フランス語
- スペイン語
- スウェーデン語
- イタリア語
- 日本語
- 韓国語
- 簡体字中国語
- 繁体字中国語

部分ロケールはすべて SOFTWARE CD に格納されています。翻訳メッセージは LANGUAGES CD に格納されています。

英語ロケールはすべて完全ロケールであり、SOFTWARE CD に格納されています。

## ロケールによって影響を受ける動作

文化が異なると、多くの場合、数、日付と時刻の表記、語句の区切り方、および著作物や話し言葉の引用符の使い方も異なります。ロケールは、以下に示す操作、ファイル、書式、および表現がさまざまな地域に合わせてどのように処理されるかを決定します。

- テキストデータのエンコーディングと処理
- リソースファイルの言語とエンコーディングの識別
- テキスト文字列の描画と配置
- クライアント間のテキストの交換
- 選択したスクリプトのコードセットとテキスト処理の要件に合う入力方式の選択
- 文化的に固有なフォントおよびアイコンのファイル

- アクションとファイルタイプ
- ユーザーインタフェース定義 (UID) ファイル
- 日付と時刻の書式
- 数値形式
- 通貨形式
- 照合順序
- ロケール特有の正規表現処理
- 通知と診断のメッセージと対話型応答の形式

Solaris 環境は、言語と文化に依存する情報をアプリケーションから分離し、アプリケーションとは別に保存します。したがって、異なる市場ごとにアプリケーションの翻訳や手直し、再コンパイルを行う必要はありません。各言語および慣習に合わせて外部情報を言語対応化するだけで新規市場に参入することができます。

## ロケールのカテゴリ

以下にロケールのカテゴリを示します。

LC_CTYPE	文字処理関数の動作を制御します。
LC_TIME	日付と時間の形式を指定します。指定には月の名前、曜日、一般的な完全表記と省略表記も含まれます。
LC_MONETARY	通貨の形式を指定します。これには、そのロケールの通貨記号、千単位の区切り記号、符号の位置、小数位以下の桁数などが含まれます。
LC_NUMERIC	小数位記号 (小数点)、千単位の区切り文字、グループ化を指定します。
LC_COLLATE	そのロケールの照合順序や、正規表現の定義を指定します。
LC_MESSAGES	翻訳メッセージの言語、および肯定と否定の応答ロケール (yes と no の文字列と表現) を指定します。
LO_LTYPE	言語のレンダリングに関する情報を提供する、配置 (レイアウト) エンジン指定します。言語のレンダリング (またはテキストレンダリング) は、スクリプトの形状と方向の属性に依存します。

---

## ロケールのカテゴリを使用した地域対応

製品の地域対応は、対象となる言語や地域に密着したユーザーとともに進める必要があります。情報の特定のスタイルや形式が、開発者にとっては非常に明確であり、普遍的に見える場合でも、ユーザーにとっては扱いにくかったり、正しくなかったり、状況によっては不快感を与えることもあるかもしれません。次の各節では、製品の地域対応の要件に合わせてカスタマイズできる Solaris OS の要素について説明します。

## 時間の形式

次の表に、異なるロケールで午後 11 時 59 分を表すいくつかの方法を示します。

表 1-2 国際的な時間の表記形式

ロケール	表記形式
カナダ (英語またはフランス語)	23:59
フィンランド語	23.59
ドイツ語	23.59 Uhr
ノルウェー語	23.59
タイ	23:59
イギリス英語	23:59

時間は、12 時間単位および 24 時間単位のどちらでも表すことができます。時間と分はコロン (:)またはピリオド (.) で区切ります。

時間帯は国別および同一国内で分かれます。時間帯は通常、協定世界時 (UTC) またはグリニッジ標準時 (GMT) との時差で表現されますが、この時差は必ずしも整数であるとは限りません。たとえば、ニューファンドランド島の時間帯と隣接する時間帯との時差は 0.5 時間です。

夏時間 (DST) の開始および終了の日付は国によって異なります。DST を全く導入していない国もたくさんあります。さらに、同一時間帯でも夏時間が異なることもあります。たとえば、米国では夏時間の実施は各州によって異なります。

## 日付の形式

次の表に、世界中で使用されている日付の形式のいくつかを示します。同じ国内で異なる場合もあります。

表 1-3 国際的な日付の表記形式

ロケール	表記	例
カナダ (英語)	dd/mm/yy	24/08/01
デンマーク語	yyyy-mm-dd	2001-08-24
フィンランド語	dd.mm.yyyy	24.08.2001
フランス語	dd/mm/yyyy	24/08/2001
ドイツ語	yyyy-mm-dd	2001-08-24

表 1-3 国際的な日付の表記形式 (続き)

ロケール	表記	例
イタリア語	dd/mm/yy	24/08/01
ノルウェー語	dd-mm-yy	24-08-01
スペイン語	dd-mm-yy	24-08-01
スウェーデン語	yyyy-mm-dd	2001-08-24
イギリス	dd/mm/yy	24/08/01
米国	mm-dd-yy	08-24-01
タイ	dd/mm/yyyy	24/08/2001

## 数値の形式

イギリスと米国は小数位をピリオドで表す数少ない2つの国です。その他の多くの国ではピリオドの代わりにコンマを使用しています。小数位の切り記号は小数点文字とも呼ばれます。同様に、イギリスと米国では千単位をコンマで区切るのに対し、他の国々では代わりにピリオドを用いたり、狭い空白文字で区切ったりしています。

データファイルにロケール固有の形式で格納されている場合、ロケールの異なるシステムに転送したときに正しく解釈されない可能性があります。たとえば、フランスの形式で数値を格納したファイルは、イギリス向けのプログラムでは使用できません。

次の表に、一般的な数値の表記形式を示します。

表 1-4 国際的な数値の表記形式

ロケール	大きな数値の表記
カナダ (英語)	4,294,967.00
デンマーク語	4.294 967.295,00
フィンランド語	4 294 967 295,00
フランス語	4 294 967 295,00
ドイツ語	4,294,967.00
イタリア語	4.294.967,00
ノルウェー語	4.294.967.295,00
スペイン語	4.294.967.295,00
スウェーデン語	4 294 967 295,00
イギリス	4,294,967,295.00

表 1-4 国際的な数値の表記形式 (続き)

ロケール	大きな数値の表記
米国	4,294,967,295.00
タイ	4,294,967,295.00

注 – 並んでいる数を区切る方法を指定する特別なロケールの規則はありません。

## 国際的な通貨の形式

通貨の単位や表記は世界中で大きく異なっています。国内での通貨記号と国際的に使われる通貨記号が異なっている場合があります。次の表に、いくつかの国の通貨形式を示します。

表 1-5 国際的な通貨の表記規則

ロケール	通貨	例
カナダ (英語)	ドル (\$)	\$1,234.56
カナダ (フランス語)	ドル (\$)	1 234,56\$
デンマーク語	クローネ (kr)	Kr 1.234,56
フィンランド語	ユーロ (€)	€ 1 234,56
フランス語	ユーロ (€)	€ 1,234
日本語	円 (¥)	¥ 1,234
ノルウェー語	クローネ (kr)	kr 1.234,56
スウェーデン語	クローナ (Kr)	1 234,56 Kr
イギリス	ポンド (£)	£1,234.56
米国	ドル (\$)	\$1,234.56
タイ	バーツ	2539 Baht
ユーロ	ユーロ (€)	€ 5,000

最新リリースでは、ユーロ通貨をサポートします。現地通貨記号は、下位互換としてまだ使用できます。

表 1-6 ユーロ通貨をサポートするユーザーロケール

地域	ロケール名	ISO コードセット
オーストリア	de_AT.ISO8859-15	8859-15
ベルギー (フランス語)	fr_BE.ISO8859-15	8859-15
ベルギー (フラマン語)	nl_BE.ISO8859-15	8859-15
デンマーク	da_DK.ISO8859-15	8859-15
エストニア	et_EE.ISO8859-15	8859-15
フィンランド	fi_FI.ISO8859-15	8859-15
フランス	fr_FR.ISO8859-15	8859-15
ドイツ	de_DE.ISO8859-15	8859-15
イギリス	en_GB.ISO8859-15	8859-15
アイルランド	en_IE.ISO8859-15	8859-15
イタリア	it_IT.ISO8859-15	8859-15
オランダ	nl_NL.ISO8859-15	8859-15
ポルトガル	pt_PT.ISO8859-15	8859-15
カタロニア (スペイン語)	ca_ES.ISO8859-15	8859-15
スペイン	es_ES.ISO8859-15	8859-15
スウェーデン	sv_SE.ISO8859-15	8859-15
米国	en_US.ISO8859-15	8859-15

ユーロロケールは、ISO8859-15 コードセットに基づいています。

変換前と変換後の通貨金額の表記に必要なスペースが異なることがあることに注意が必要です。たとえば、\$1,000 は €1.307.000 です。

ユーロ地域内のロケールの場合、ロケールの現在の設定値は locale ユーティリティの LC\_MONETARY オペランドで表されます。たとえば、ドイツ語の場合は、次の表のようになります。

表 1-7 ドイツ語のロケールとその LC\_MONETARY オペランド

ロケール	LC_MONETARY
de_DE.ISO8859-1	DM
de_DE.ISO8859-15	Euro
de_DE.UTF-8	Euro

表 1-7 ドイツ語のロケールとその LC\_MONETARY オペランド (続き)

ロケール	LC_MONETARY
de_DE.ISO8859-15@euro	Euro
de_DE.UTF-8@euro	Euro

## 語と文字の相違点

この節では、言語間の重要な相違点について説明します。

### 語の区切り

英語では、通常、語は空白文字で区切ります。それに対して中国語、日本語、タイ語などの言語では多くの場合、語を区切りません。

### ソートの順序

特定の文字をソートする順序は、すべての言語で同じではありません。たとえば、ドイツ語の文字 ö は普通の o と一緒にソートされますが、スウェーデン語ではアルファベットの最後の文字として別にソートされます。言語によっては、文字の優先順位を決めるために重みが定められています。たとえば、タイ語の辞書では、文字に異なる重みを与えることによってソート順序が決定されます。

### 文字セット

アルファベット文字や特殊文字の数は文字セットによって異なります。英語のアルファベットは 26 文字しかありませんが、それよりもずっと多くの文字を使用する言語もあります。たとえば、日本語には 20,000 以上の文字があり、中国語にはそれ以上の文字があります。

### 西ヨーロッパのアルファベット

ほとんどの西ヨーロッパ諸国のアルファベットは、英語圏の国で使用されている 26 文字の標準アルファベットに類似しています。ただし、基本的な文字が追加されていたり、記号 (またはアクセント) の付いた文字や合字が使用される場合があります。

### 日本語のテキスト

日本語のテキストには次の 3 種類の文字が混在しています。

- 中国の漢字に由来する表意文字である漢字

## ■ ひらがなとカタカナという2つの表音文字(音節文字)

ひらがなとカタカナはそれぞれ対になっていますが、ひらがなの方がより一般的で、丸みのある形をしています。漢字は語幹となる語を書くのに使用され、カタカナは主に外来語を書き記すのに使用されます。

漢字の数は数万もありますが、使用される漢字の数は徐々に減ってきています。現在、頻繁に使用される漢字の数は3,500字程度ですが、一般的な日本人の語彙に使用される漢字はおよそ2,000字ほどです。ただし、コンピュータシステムは日本工業規格(JIS)で定められた7,000字以上の漢字に対応しなければなりません。また、170文字ほどのひらがなとカタカナも必要です。日本語の文章の平均的な文字別構成は、ひらがな55%、漢字35%、カタカナ10%となっています。また日本語の文章にはアラビア数字やローマ字も使用されます。

文章に漢字を全く使用しないこともできますが、ほとんどの日本人にとって、そのような文章は理解が困難です。

## 韓国語のテキスト

韓国語テキストは、ハングルと呼ばれる表音文字で表されます。ハングルには、jamosと呼ばれる母音と子音から成る11,000以上の文字があります。通常、韓国語のコンピュータシステムでは、全ハングル文字の内約3,000文字が使用されています。韓国語にはまた、ハンジャと呼ばれる中国の漢字から派生した表意文字があります。韓国語では6,000以上のハンジャ文字が必要です。ハンジャは主に、ハングルだけでは意味があいまいになる場合に使用されます。ハングル文字では子音と母音を組み合わせることで1つの音節を構成します。結合された子音と母音は1つの音節としてハングル文字になります。ハングル文字は通常、四角く配列されており、ハンジャ文字と同じスペースを必要とします。韓国語では、アラビア数字、ローマ字、および特殊記号も表示されます。

## タイ語のテキスト

タイ語の文字は、4つの表示セルを持つディスプレイ上のカラム位置として定義されます。各カラム位置に表示できるのは最大3文字です。表示セルの構成内容はタイ文字の分類に基づいています。タイ文字の一部は、別の分類の文字と組み合わせることができます。組み合わせられた文字は、同じセルに入ります。それ以外の場合は、別のセルに入ります。

## 中国語のテキスト

中国語は通常、表意文字である漢字ですべてを表記します。

- 中華人民共和国(PRC)の場合、GB2312(zh ロケール)には一般によく使用されている約7,000の漢字があり、GBK文字セット(zh.GBK ロケール)には20,000以上の文字があり、GB18030-2000文字セット(zh\_CN.GB18030 ロケール)には約30,000の文字があります(Unicode 3.0で定義されているすべてのCJK拡張A文字)

を含む)。

- 台湾では、CNS11643-1992 (zh\_TW ロケール) と Big5 (zh\_TW.BIG5 ロケール) 文字セットが最もよく使用されています。どちらのロケールにも約 13,000 の漢字が定義されています。
- 香港では、Big5 文字セットに 4702 文字が追加された Big5-HKSCS 文字セット (zh\_HK.BIG5HK) が使用されています。

語幹文字でない限り、通常 1 つの文字は 2 つ以上の部分で形成されます。最も一般的なのは 2 つの部分で形成されるものです。2 つの部分で構成される文字では、1 つが意味を表し、もう 1 つは発音を表すのが普通です。しかし、両方とも意味を表すこともあります。漢字では部首が最も重要な要素です。漢字は伝統的に部首によって分類され、部首の数は数百にもなります。異なる文字で同じ読み方をするものも多数ありますが、使用する際は明確に区別されます。同じ文字で異なる読み方をするものもあります。

中国語には四声と呼ばれる声調があり、文脈の中で適切な文字を音声によって区別します。一方、日本語と韓国語には声調はありません。

中国語の発音を表す表音体系にはいくつかの種類があります。中華人民共和国で最も普及しているのはピンイン方式です。これはローマ字を使用するもので、広く西部で採用されており、たとえば北京を **Beijing** と表します。ウェードガイル方式は旧式のもので、北京を **Peking** と表すなど地名の表現に以前は使用されていました。台湾では、独特の字体を持った表音用のアルファベットである *zhuyin (bopomofo)* がよく使用されます。

## ヘブライ語のテキスト

スクリプトをヘブライ語やイディッシュ語で作成するときにはヘブライ語のテキストが使用されます。ヘブライ語は双方向スクリプトを使用します。ヘブライ文字は右から左に読み書きされますが、数字は左から右に読まれます。さらに、ヘブライ語テキストに組み込まれた英語テキストも左から右に読まれます。

ヘブライ語では、27 文字のアルファベットのほかに、標準的なラテン (英語) アルファベットの句読点と数字が使用されます。ヘブライ語テキストには、さらに、母音と発音記号が付きます。これらの記号は、基本文字内のドット (*dagesh*) や、文字の下の母音記号、文字の左上のアクセント記号として使用されます。これらの記号は、通常、典礼のテキストとして使用されるだけで日常生活ではほとんど使用されません。ヘブライ語には大文字はありません。

## ヒンディー語のテキスト

ヒンディー語のテキストは、神の書き物を意味するデーバナーガリーというスクリプトで使用されます。ヒンディー語は表音言語であり、一連の音節として書かれます。各音節は、3 種類の子音文字、独立母音、従属母音の記号であるアルファベット部分 (デーバナーガリー文字) から構成されます。音節自体は子音と母音のコア部分から構成され、その前に子音が付くこともあります。基線から始まる英語とは異なり、デー

バナーガリー文字は、文字の上部に引かれた水平線 (頭なで線) から垂れ下がります。これらの文字は、内容に応じて結合したり、形を変えたりします。ヘブライ語と同じように、ヒンディー語テキストには、大文字と小文字の区別はありません。

---

## キーボードの相違点

U.S. キーボードにあるすべての文字がほかのキーボードにあるとは限りません。同様に、U.S. キーボードにはない文字のキーがほかのキーボードに付いている場合もあります。

入力は Solaris OS によって処理されるため、任意のロケールの文字を任意のキーボードから入力できます。

---

注 - SPARC® マシンでは、Compose キーを使用すれば、区別的発音符が付いた任意のラテン文字を、サポートされる任意の ISO8859 文字セットの中に生成できます。Compose キーは、ラテンベースのロケールで使用できますが、韓国語、中国語、日本語ロケールでは UTF-8 ロケールを除き使用できません。

---

---

## 用紙サイズの相違点

それぞれの国で一般的に使用される用紙サイズの数多くはありません。しかも、そのうちの 1 つが他よりも頻繁に使用されます。ほとんどの国は、ISO 216 規格の「筆記用紙および印刷物の分類 - A 版と B 版の定型サイズ」に従っています。

国際化アプリケーションでは、使用可能なページサイズをあらかじめ想定するべきではありません。Solaris には、出力ページのサイズを管理する機能はありません。したがって、この機能は、アプリケーションプログラムで行う必要があります。次の表に、各国で一般に使用されているページサイズを示します。

表 1-8 一般的な国際ページサイズ

用紙の種類	寸法	採用している国
ISO A4	21.0 cm x 29.7 cm	米国以外
ISO A5	14.8 cm x 21.0 cm	米国以外
JIS B4	25.9 cm x 36.65 cm	日本

表 1-8 一般的な国際ページサイズ (続き)

用紙の種類	寸法	採用している国
JIS B5	18.36 cm x 25.9 cm	日本
米国 レター	8.5 インチ x 11 インチ	米国、カナダ
US リーガル	8.5 インチ x 14 インチ	米国、カナダ



## 第 2 章

---

### 一般的な国際化機能

---

この章では、Solaris OS に含まれているいくつかの国際化機能について説明します。この章の内容は次のとおりです。

- 41 ページの「コードセットの独立性のサポート」
- 43 ページの「ロケールデータベース」
- 44 ページの「プロセスコードの書式 (ワイド文字表現)」
- 44 ページの「複数バイトサポート環境」
- 45 ページの「動的にリンクされたアプリケーション」
- 45 ページの「変更されたインタフェースについて」
- 46 ページの「ctype マクロ」
- 47 ページの「libc の国際化 API」
- 54 ページの「genmsg ユーティリティ」
- 55 ページの「ユーザーによるコード変換の定義と拡張」
- 56 ページの「国際化ドメイン名 (IDN) のサポート」

---

### コードセットの独立性のサポート

EUC は Extended UNIX® Code の省略形です。Solaris OS は、日本の PC-Kanji (Shift\_JIS)、台湾の Big5、中華人民共和国の GBK のような非 EUC エンコーディングをサポートしています。コンピュータ市場の大半では EUC 以外のコードセットのサポートが要求されるため、最新の Solaris 環境では EUC と非 EUC コードセットのサポートを可能にするフレームワークを提供します。このサポートをコードセットの独立性 (CSI) と呼びます。

CSI の目標は、Solaris OS のライブラリやコマンドから、特定のコードセットやエンコーディング方法への依存性を除去することです。CSI アーキテクチャにより、Solaris OS において UNIX ファイルシステムに安全なエンコーディングをサポートできます。CSI は UTF-8、PC-Kanji、Big5 など、多くの非 EUC コードセットをサポートします。

## CSI のアプローチ

CSI により、アプリケーションおよびプラットフォームソフトウェア開発者は、UTF-8 など、エンコーディングからコードを独立させたり、ソースコードを変更せずに新しいエンコーディングを採用したりすることができます。アプリケーションは UTF-16 に依存する必要がないので、このアーキテクチャのアプローチは Java™ の国際化とも異なります。

既存の多くの国際化アプリケーション (たとえば Motif など) は、基礎となるシステムから自動的に CSI のサポートを継承します。これらのアプリケーションは、変更しなくても新しいロケールで動作します。

CSI は本来どのコードセットにも依存しませんが、最新の Solaris システムではファイルコードのエンコーディング (コードセット) には、次のような前提条件があります。

- ファイルコードは ASCII のスーパーセットである。
- NULL バイトで終わる複数バイトの文字列をサポートする必要があるため、複数バイト文字の一部に NULL バイト値 (0x00) を使用することはできない。
- UNIX のパス名をサポートする必要があるため、複数バイト文字の一部に ASCII スラッシュ文字のバイト値 (0x2f) を使用することはできない。

## CSI 対応のコマンド

この節では、最新の Solaris 環境で使用できる CSI 対応のコマンドを示します。各コマンドのマニュアルページでは、属性の節で、そのコマンドが CSI 対応かどうかを説明しています。

特に断りがない限り、これらのコマンドは /usr/bin ディレクトリに格納されています。

/usr/lib/diffh	/usr/xpg4/bin/more	bdiff
/usr/sbin/accept	/usr/xpg4/bin/mv	cancel
/usr/sbin/reject	/usr/xpg4/bin/nice	cat
/usr/ucb/lpr	/usr/xpg4/bin/nohup	catman
/usr/xpg4/bin/awk	/usr/xpg4/bin/od	chgrp
/usr/xpg4/bin/cp	/usr/xpg4/bin/pr	chmod
/usr/xpg4/bin/date	/usr/xpg4/bin/rm	chown
/usr/xpg4/bin/du	/usr/xpg4/bin/sed	cmp
/usr/xpg4/bin/ed	/usr/xpg4/bin/sort	col
/usr/xpg4/bin/edit	/usr/xpg4/bin/tail	comm
/usr/xpg4/bin/egrep	/usr/xpg4/bin/tr	compress
/usr/xpg4/bin/env	/usr/xpg4/bin/vedit	cpio
/usr/xpg4/bin/ex	/usr/xpg4/bin/vi	csch
/usr/xpg4/bin/expr	/usr/xpg4/bin/view	csplit
/usr/xpg4/bin/fgrep	acctcom	cut
/usr/xpg4/bin/lp	apropos	diff
/usr/xpg4/bin/ls	batch	diff3

disable	news	sh
echo	nroff	split
expand	pack	strconf
file	paste	strings
find	pcat	sum
fold	pg	tabs
ftp	printf	tar
gencat	priocntl	tee
geteopt	ps	touch
getoptcv	pwd	tty
head	rcp	uncompress
join	red	unexpand
jsh	remsh	uniq
kill	rksh	unpack
ksh	rsh	wc
lp	rsmdir	whatis
man	script	write
mkdir	sdiff	xargs
msgfmt	settime	zcat

## CSI 対応のライブラリ

libc (/usr/lib/libc.so) のほとんどすべての関数は CSI に対応しています。ただし、libc の以下の関数は EUC 依存の関数であるため、CSI に対応していません。

- csetcol()
- csetlen()
- csetno()
- euccol()
- euclen()
- eucscol()
- getwidth()
- wcsetno()

最新の Solaris 環境の libgen (/usr/ccs/lib/libgen.a) と libcurses (/usr/ccs/lib/libcurses.a) は国際化されていますが、CSI に対応していません。

---

## ロケールデータベース

ロケールデータベースの形式および構造は非公開であり、将来のリリースで変更される可能性があります。国際化アプリケーションを作成するときには、libc の国際化 API を使用します。この API については、ロケールデータベースとのリンクに関連してではなく [47 ページの「libc の国際化 API」](#) で説明されています。

---

注 – Solaris 環境を使用する場合は、Solaris の最新リリースに含まれているロケールデータベースを使用してください。以前のバージョンの Solaris のロケールデータベースを使用しないでください。

---

---

## プロセスコードの書式 (ワイド文字表現)

プロセスコードの書式 (Solaris OS ではワイド文字コード形式ともいう) は公開されていません。そのため、将来のリリースで変更されることがあります。したがって、国際化アプリケーションを作成するときには、このワイド文字の形式が将来変更される可能性があることに注意してください。その代わりに libc の国際化 API を使用することをお勧めします (47 ページの「libc の国際化 API」を参照)。

---

注 – Unicode ロケールのプロセスコードは UTF 32 で表現されています。UTF 32 の詳細については、Unicode Consortium (<http://www.unicode.org/>) の「Unicode Standard Annex #19: UTF 32」と「Unicode Standard Annex #27: Unicode 3.1」を参照してください。

---

---

## 複数バイトサポート環境

複数バイト文字はシングルバイトとして保存できない文字の総称で、中国語、日本語、韓国語などの文字を含みます。複数バイト文字を格納するには、2、3、または4バイトが必要です。より詳細な定義については、ISO/IEC 9899:1990 サブクラス 3.13 の項目を参照してください。

ANSI C の Amendment 1 (ISO/IEC 9899:1990) では、Multibyte Support Environment (MSE) と呼ぶ新しい国際化機能が追加されました。Amendment 1 には、状態属性を持つ複数バイトコードセットやワイド文字の処理サポートを強化するための新しい国際化 API が定義されています。

プログラミングモデルでは、これらの複数バイト文字を論理ユニットとして読み込み、ワイド文字として内部的に保持できます。プログラム内において、ワイド文字は論理エンティティとして取り扱うことができます。また、適切な変換処理を行なったのち、これらワイド文字を論理ユニットとして出力できます。

この手順は、シングルバイト文字を読み込んで加工したのち出力する場合と似ています。このため MSE によって、シングルバイト文字の場合と同じプログラミングモデルを用いて複数バイト文字をプログラムで取り扱うことができます。

---

## 動的にリンクされたアプリケーション

アプリケーションを libc などのシステムライブラリにリンクする方法として、動的リンクと静的リンクを選択できます。システムライブラリの国際化機能を必要とするアプリケーションは動的にリンクしなければなりません。このようなアプリケーションが静的にリンクされている場合は、setlocale 関数を使用してロケールを C または POSIX 以外のものに設定する操作は失敗します。静的にリンクされたアプリケーションは、C および POSIX ロケールでのみ動作します。

デフォルトでは、リンカープログラムはアプリケーションを動的にリンクしようとします。リンカーやコンパイラのコマンド行のオプションとして、-Bstatic や -dn を指定すると、アプリケーションは静的にリンクされる場合があります。既存のアプリケーションが動的にリンクされているかどうかを調べるには、/usr/bin/ldd コマンドを使用します。

たとえば、次のコマンドに対する応答によって、/sbin/sh コマンドが動的にリンクされたプログラムではないことがわかります。

```
% /usr/bin/ldd /sbin/sh
ldd: /sbin/sh: file is not a dynamic executable or shared object
```

次のコマンドに対する応答は、/usr/bin/ls コマンドは 2 つのライブラリ libc.so と libdl.so.1 に動的にリンクされていることを示します。

```
% /usr/bin/ldd /usr/bin/ls
libc.so.1 => /usr/lib/libc.so.1
libdl.so.1 => /usr/lib/libdl.so.1
```

---

## 変更されたインタフェースについて

libw と libintl は libc に移動されました。したがって、libw や libintl はありません。

共有オブジェクトは、既存のアプリケーションの実行の互換性を保証し、アーカイブとともにアプリケーションのビルド (コンパイル) 環境の互換性を提供します。ただし、libw や libintl 付きでアプリケーションをビルドする必要はなくなりました。

次のリストは、libw のスタブ (Stub) エントリポイントを示しています。

fgetwc	getwc	isideogram	iswalnum
fgetws	getwchar	isnumber	iswalpha
fputwc	getws	isphonogram	iswcntrl
fputws	isenglish	isspecial	iswctype

iswdigit	wscat	wcstol	wslen
iswgraph	wcschr	wcstoul	wsncasecmp
iswlower	wcsclen	wcswcs	wsncat
iswprint	wcscmp	wcswidth	wsncmp
iswpunct	wscoll	wcsxfrm	wsncpy
iswspace	wcscopy	wctype	wsprbrk
iswupper	wcscspn	wcwidth	wsprintf
iswxdigit	wcsftime	wscasecmp	wsrchr
putwc	wscncat	wscat	wsscancf
putwchar	wcsncmp	wchr	wsspn
putws	wcsncpy	wscmp	wstod
strtows	wcspbrk	wscol	wstok
towlower	wcsrchr	wscoll	wstol
towupper	wcsspn	wscopy	wstoll
ungetwc	wcstod	wscspn	wstostr
watoll	wcstok	wsdup	wsxfrm

次のリストは、libintl のスタブ (Stub) エントリポイントを示しています。

```
bindtextdomain
dcgettext
dgettext
gettext
textdomain
```

---

## ctype マクロ

文字の分類や文字の変換マクロは、`/usr/include/ctype.h` で定義されています。最新の Solaris 環境では、XPG4 で定義されている文字の分類および変換セマンティクスをサポートする一連の `ctype` マクロが提供されています。XPG4 や XPG4.2 アプリケーションから新しいマクロに自動的にアクセスするためには、次の条件のどれかが満たされていなければなりません。

- `_XPG4_CHAR_CLASS` が定義されている
- `_XOPEN_SOURCE` および `_XOPEN_VERSION=4` が定義されている
- `_XOPEN_SOURCE` および `_XOPEN_SOURCE_EXTENDED=1` が定義されている

`_XOPEN_SOURCE`、`_XOPEN_VERSION`、`_XOPEN_SOURCE_EXTENDED` は、新しい `ctype` マクロのほかに XPG4 関連機能をもたらすので、XPG4 または XPG4.2 以外のアプリケーションは `_XPG4_CHAR_CLASS` を使用しなければなりません。

これに対応する `ctype` 関数もあります。最新の Solaris 環境は XPG4 セマンティクスもサポートします。

---

## libc の国際化 API

最新の Solaris 環境は 2 種類の API を提供します。

- 複数バイト (ファイルコード)
- ワイド文字 (プロセスコード)

ワイド文字コードは、一定の幅を持つ論理エンティティです。したがって、複数バイトを使用する時のように適切な文字境界を管理する必要はありません。

プログラムでファイルから入力を受け取る際には、`fscanf` や `fwscanf` などの入力関数を使ってファイルの複数バイトデータを直接ワイド文字のプロセスコードに変換できます。あるいは、データの入力後に `mbtowc` や `mbsrtowcs` などの変換関数を使ってデータを変換することもできます。出力データのワイド文字形式を複数バイト文字形式に変換する場合は、`fwprintf` や `fprintf` などの出力関数を使って直接変換するか、出力前に `wctomb` や `wcsrtombs` などの変換関数を使って変換します。

以下の各表に、最新の Solaris 製品の国際化 API を示します。

次の表は libc のメッセージング関数 API です。

表 2-1 libc のメッセージング関数

ライブラリルーチン	説明
<code>bindtextdomain()</code>	メッセージドメインパスをバインドする
<code>catclose()</code>	メッセージカタログを閉じる
<code>catgets()</code>	プログラムメッセージを読み取る
<code>catopen()</code>	メッセージカタログを開く
<code>dcgettext()</code>	指定したドメインとカテゴリのメッセージカタログからメッセージを取得する
<code>dgettext()</code>	指定したドメインのメッセージカタログからメッセージを取得する
<code>gettext()</code>	メッセージデータベースからテキスト文字列を取得する
<code>textdomain()</code>	現在のドメインを設定および照会する

次の表に libc のコード変換関数 API を示します。

表 2-2 libc のコード変換関数

ライブラリルーチン	説明
<code>iconv()</code>	コードを変換する
<code>iconv_close()</code>	変換記述子の割り当てを解除する
<code>iconv_open()</code>	変換記述子を割り当てる

次の表に libc の正規表現 API を示します。

表 2-3 libc の正規表現関数

ライブラリルーチン	説明
<code>fnmatch()</code>	ファイル名またはパス名を照合する
<code>regcomp()</code>	正規表現をコンパイルする
<code>regerror()</code>	エラーコードとエラーメッセージのマッピングを行う
<code>regexexec()</code>	正規表現の照合を実行する
<code>regfree()</code>	<code>regcomp()</code> により割り当てられたメモリーを解放する

次の表に libc のワイド文字関数 API を示します。

表 2-4 libc のワイド文字クラス

ライブラリルーチン	説明
<code>wctrans()</code>	文字のマッピングを定義する
<code>wctype()</code>	文字クラスを定義する

次の表に libc のロケールの変更および照会を行う API を示します。

表 2-5 libc のロケールの変更および照会関数

ライブラリルーチン	説明
<code>setlocale()</code>	プログラムのロケールを変更および照会を行う

次の表に libc のロケールデータを照会する API を示します。

表 2-6 libc のロケールデータ照会関数

ライブラリルーチン	説明
localeconv()	現在のロケールの通貨および数値形式の情報を取得する
nl_langinfo()	現在のロケールの言語および文化情報を取得する

次の表に libc の文字分類関数 API を示します。

表 2-7 libc の文字分類および変換関数

ライブラリルーチン	説明
isalnum()	文字は英数字か
isalpha()	文字はアルファベットか
isascii()	文字は ASCII 文字か
iscntrl()	文字は制御文字か
isdigit()	文字は数か
isenglish()	ワイド文字は補助コードセットの英語のアルファベットか
isgraph()	文字は表示可能な文字か
isideogram()	ワイド文字は表意文字か
islower()	文字は小文字か
isnumber()	ワイド文字は補助コードセットの数か
isphonogram()	ワイド文字は表音文字か
isprint()	文字は印刷可能か
ispunct()	文字は句読点か
isspace()	文字は空白か
isspecial()	ワイド文字は補助コードセットの文字か
isupper()	文字は大文字か
iswalnum()	ワイド文字は英数字か
iswalpha()	ワイド文字はアルファベットか
iswascii()	ワイド文字は ASCII 文字か
iswcntrl()	ワイド文字は制御文字か
iswdigit()	ワイド文字は数か

表 2-7 libc の文字分類および変換関数 (続き)

ライブラリルーチン	説明
<code>iswgraph()</code>	ワイド文字は表示可能な文字か
<code>iswlower()</code>	ワイド文字は小文字か
<code>iswprint()</code>	ワイド文字は印刷可能な文字か
<code>iswpunct()</code>	ワイド文字は句読点か
<code>iswspace()</code>	ワイド文字は空白か
<code>iswupper()</code>	ワイド文字は大文字か
<code>iswxdigit()</code>	ワイド文字は 16 進数か
<code>isxdigit()</code>	文字は 16 進数か
<code>tolower()</code>	大文字を小文字に変換する
<code>toupper()</code>	小文字を大文字に変換する
<code>towctrans()</code>	ワイド文字のマッピング
<code>tolower()</code>	ワイド文字の大文字を小文字に変換する
<code>toupper()</code>	ワイド文字の小文字を大文字に変換する

次の表に libc の文字照合関数 API を示します。

表 2-8 libc の文字照合関数

ライブラリルーチン	説明
<code>strcoll()</code>	文字列を照合する
<code>strxfrm()</code>	文字列を照合用に変換する
<code>wscoll()</code>	ワイド文字の文字列を照合する
<code>wcsxfrm()</code>	ワイド文字の文字列を照合用に変換する

次の表に libc の通貨処理関数 API を示します。

表 2-9 libc の通貨処理関数

ライブラリルーチン	説明
<code>localeconv()</code>	現在のロケールの通貨形式情報を取得する
<code>strfmon()</code>	通貨の値を文字列表現に変換する

次の表に libc の日付と時間の処理関数を示します。

表 2-10 libc の日付と時間の処理関数

ライブラリルーチン	説明
getdate ()	ユーザー形式の日付と時間を変換する
strftime ()	日付と時間を文字列表現に変換する。%u の変換機能は、X/Open CAE 仕様、System Interfaces and Headers, Issue 4, Version 2 に準拠しています。この機能は、10 進法の 1 から 7 で週日を表しており、現在では 1 が月曜日を表します。
strptime ()	日付と時間の変換

次の表に libc の複数バイト処理関数 API を示します。

表 2-11 libc の複数バイト処理関数

ライブラリルーチン	説明
btowc ()	シングルバイトからワイド文字に変換する
mblen ()	文字のバイト数を取得する
mbrlen ()	1 つの文字に含まれるバイト数を取得する (再起動可能)
mbrtowc ()	1 つの文字を 1 つのワイド文字に変換する (再起動可能)
mbsinit ()	変換オブジェクトの状態を決定する
mbsrtowcs ()	文字列をワイド文字の文字列に変換する (再起動可能)
mbstowcs ()	文字列をワイド文字の文字列に変換する
mbtowc ()	1 つの文字を 1 つのワイド文字に変換する

次の表に libc のワイド文字および文字列処理関数を示します。

表 2-12 libc のワイド文字および文字列処理

ライブラリルーチン	説明
wcrtomb ()	ワイド文字から文字に変換する (再起動可能)
wscat ()	ワイド文字の文字列を照合する
wcschr ()	ワイド文字の文字列内の文字を検索する
wscmp ()	ワイド文字の文字列を比較する
wscpy ()	ワイド文字の文字列をコピーする
wscspn ()	ワイド文字の文字列が別のワイド文字の文字列に現れない長さを返す

表 2-12 libc のワイド文字および文字処理 (続き)

ライブラリルーチン	説明
wcslen()	ワイド文字の文字列の長さを取得する
wcsncat()	ワイド文字の文字列を長さ $n$ に連結する
wcsncmp()	ワイド文字の文字列を長さ $n$ 分比較する
wcsncpy()	ワイド文字の文字列を長さ $n$ 分コピーする
wcspbrk()	別のワイド文字の文字列に含まれている、ワイド文字の文字列の先頭のポインタを返す
wcsrchr()	ワイド文字の文字列を右から検索する
wcsrtombs()	文字列をワイド文字の文字列に変換する (再起動可能)
wcsspn()	ワイド文字の文字列が別のワイド文字の文字列に現れる長さを返す
wcstod()	ワイド文字の文字列を倍精度に変換する
wcstok()	ワイド文字の文字列の中でトークンを移動する
wcstol()	ワイド文字の文字列をロング整数に変換する
wcstombs()	ワイド文字の文字列を複数バイトの文字列に変換する
wcstoul()	ワイド文字の文字列を符号なしロング整数に変換する
wscwcs()	ワイド文字の文字列内の文字列を検索する
wcswidth()	ワイド文字の文字列のカラム数を調べる
wctob()	ワイド文字からシングルバイト文字に変換する
wctomb()	ワイド文字を複数バイト文字に変換する
wcwidth()	ワイド文字のカラム数を調べる
wscol()	ワイド文字の文字列の表示幅を返す
wsdup()	ワイド文字の文字列を重複させる

次の表に libc のワイド文字入力および出力の書式化関数を示します。

表 2-13 libc のワイド文字入力および出力の書式化関数

ライブラリルーチン	説明
fwprintf()	ワイド文字書式付き出力を印刷する
fscanf()	ワイド文字書式付き入力を変換する
swprintf()	ワイド文字書式付き出力を印刷する

表 2-13 libc のワイド文字入力および出力の書式化関数 (続き)

ライブラリルーチン	説明
swscanf ()	ワイド文字書式付き入力を変換する
vfwprintf ()	stdarg 引数リストをワイド文字書式で出力する
vswprintf ()	stdarg 引数リストをワイド文字書式で出力する
wprintf ()	ワイド文字書式付き出力を印刷する
wscanf ()	ワイド文字書式付き入力を変換する
wsprintf ()	書式に従ってワイド文字の文字列を生成する
wsscanf ()	書式付き入力規約

次の表に libc のワイド文字列関数 API を示します。

表 2-14 libc のワイド文字列関数

ライブラリルーチン	説明
wcsstr ()	ワイド文字の部分文字列を検索する
wmemchr ()	ワイド文字をメモリー内で検索する
wmemcmp ()	メモリー内のワイド文字を比較する
wmemcpy ()	メモリー内のワイド文字をコピーする
wmemmove ()	領域がオーバーラップする状態でメモリー内のワイド文字をコピーする
wmemset ()	メモリー内のワイド文字を設定する
wscasecmp ()	大文字と小文字を区別せずにワイド文字の文字列を比較する
wncasecmp ()	コード文字列動作を処理する

次の表に libc のワイド文字入力および出力関数を示します。

表 2-15 libc のワイド文字入力および出力関数

ライブラリルーチン	説明
fgetwc ()	ストリームから複数バイト文字を取り出し、ワイド文字に変換する
fgetws ()	ストリームから複数バイトの文字列を取り出し、ワイド文字に変換する
fputwc ()	ワイド文字を複数バイト文字に変換し、ストリームに送る

表 2-15 libc のワイド文字入力および出力関数 (続き)

ライブラリルーチン	説明
<code>fputws()</code>	ワイド文字を複数バイトの文字列に変換し、ストリームに送る
<code>fwide()</code>	ストリームの方向を設定する
<code>getwchar()</code>	標準入力から複数バイト文字を取り出し、ワイド文字に変換する
<code>getws()</code>	標準入力から複数バイト文字列を取り出し、ワイド文字に変換する
<code>putwchar()</code>	ワイド文字を複数バイト文字に変換し、標準入力に送る
<code>putws()</code>	ワイド文字を複数バイト文字列に変換し、標準入力に送る
<code>ungetwc()</code>	ワイド文字を入力ストリームに戻す

## genmsg ユーティリティ

新しい `genmsg` ユーティリティは、国際化されたソースメッセージカタログを作成するために、`catgets()` ファミリの関数で使用されます。このユーティリティは、ソースプログラムファイルの `catgets` での関数の呼び出しを調べ、見つかった情報からソースメッセージカタログを作成します。たとえば、次のようにします。

```
% cat example.c
...
/* NOTE: %s is a file name */
printf(catgets(catd, 5, 1, "%s cannot be opened.));
/* NOTE: "Read" is a past participle, not a present
    tense verb */
printf(catgets(catd, 5, 1, "Read"));
...
% genmsg -c NOTE example.c
The following file(s) have been created.
    new msg file = "example.c.msg"
% cat example.c.msg
$quote "
$set 5
1          "%s cannot be opened"
    /* NOTE: %s is a file name */
2          "Read"
    /* NOTE: "Read" is a past participle, not a present
    tense verb */
```

上記の例では、ソースファイル `example.c` について `genmsg` が実行され、`example.c.msg` という名前のソースメッセージカタログが作成されています。`-c` オプションの引数 `NOTE` により、`genmsg` はカタログにコメントを含めます。ソースプログラムのコメントに指定された文字列が含まれている場合、コメントはメッセージカタログの `catgets` の呼び出しから抽出された次の文字列の後に表示されます。

`genmsg` を使用して、メッセージセット内のメッセージの番号を自動的に付けることができます。

詳細については、`genmsg(1)` のマニュアルページを参照してください。

フォーマットされたメッセージカタログファイルを生成する場合は、`gencat(1)` ユーティリティを使用します。

ポータブルメッセージファイル (`.po` ファイル) に対するメッセージ抽出ユーティリティ、および `.po` ファイルからメッセージオブジェクトファイル (`.mo` ファイル) を生成する方法については、それぞれ `xgettext(1)` と `msgfmt(1)` のマニュアルページを参照してください。

---

## ユーザーによるコード変換の定義と拡張

`geniconvtbl` ユーティリティを使ってコードセットコンバータを定義できます。

このユーティリティを使用すると、ユーザー定義の、またユーザーによるカスタマイズが可能なコードセットの変換を `iconv(1)` や `iconv(3C)` などの標準システムユーティリティやインタフェースによって行えます。そのため、互換性のないデータ形式 (特にメーカー独自のアプリケーションや従来のアプリケーションのデータ) をアプリケーションで処理することが容易になります。既存の Solaris コードセット変換を変更することもできます。

このユーティリティで使用する入力ソースファイルの例は `/usr/lib/iconv/geniconvtbl/srcs/` ディレクトリにあります。

ユーザーが定義して適切に配置したコード変換は、32 ビットおよび 64 ビット Solaris OS の `iconv(1)` ユーティリティや `iconv(3C)` 関数から使用できます。

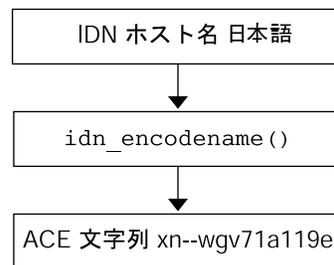
---

## 国際化ドメイン名 (IDN) のサポート

国際化ドメイン名 (IDN) によって、ホスト名およびドメイン名に英語以外の言語を使用できます。英語以外のホスト名およびドメイン名を使用するには、RFC 3490 で指定されるリゾルバルーチンに名前を送信する前に、ASCII 互換エンコーディング (ACE) 名に変換します。システム管理アプリケーションが IDN をサポートしない場合、システム管理者はシステムのファイルおよびアプリケーションにも ACE 名を使用する必要があります。

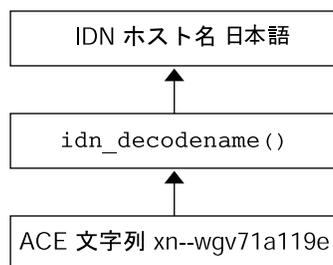
RFC 3490 「アプリケーションにおける国際化ドメイン名 (IDNA)」を参照してください。

libidnkit (3EXT) の国際化ドメイン名の API は、UTF-8 またはアプリケーションロケールのコードセットと ACE の便利な変換機能を提供します。idn\_decodename2 (3EXT) を使用する場合、入力引数のコードセットとして任意のコードセット名を指定することもできます。



getaddrinfo(3SOCKET) など、リゾルバルーチンに入力された ACE 文字列

図 2-1 IDN から ACE への変換



`getnameinfo(3SOCKET)` など、  
リゾルバルーチンから返された ACE 文字列

図 2-2 ACE から IDN への変換

次の表は、使用可能な双方向の `iconv` コード変換です。

表 2-16 `iconv` コード変換

元のコード	変換後のコード
ACE	UTF-8
ACE-ALLOW-UNASSIGNED	UTF-8
UTF-8	ACE
UTF-8	ACE-ALLOW-UNASSIGNED

ACE と ACE-ALLOW-UNASSIGNED の `iconv` コード変換名には次の意味があります。

■ ACE

ACE は `fromcode` または `toencode` 名であり、RFC 3490 で定義されている ASCII 互換エンコーディングを参照するために `iconv` コード変換で使用されます。この変換は STD3 ASCII のルールを使用します。割り当てられていない文字は許可されません。通常、ACE はホスト名またはドメイン名をマシンに格納または提供するために使用します。

■ ACE-ALLOW-UNASSIGNED

ACE-ALLOW-UNASSIGNED は割り当てられていない文字を許可する以外は、ACE と同じ処理を実行します。通常、ACE-ALLOW-UNASSIGNED は照会のために使用します。

次の例は、`hostnames.txt` ファイルからの入力による ACE から UTF-8 への変換です。出力は標準出力に送られます。

```
system% iconv -f ACE -t UTF-8 hostnames.txt
```

専用の IDN 変換ユーティリティ `idnconv(1)` では、さまざまなオプションとともに IDN 変換が可能です。オプションによって変換の詳細を制御できます。

IDN、変換ルーチン、および iconv コード変換については、libidnkit(3LIB)、idn\_decodename(3EXT)、idn\_decodename2(3EXT)、idn\_encodename(3EXT)、および iconv\_en\_US.UTF-8(5) のマニュアルページを参照してください。

## 第 3 章

---

# Solaris 環境における地域対応

---

この章では、最新の Solaris 環境の地域対応機能について説明します。この章の内容は次のとおりです。

- 59 ページの「地域対応のためのソフトウェアサポート」
- 60 ページの「サポートされるロケール」
- 67 ページの「ロケールの複数キーの Compose キーシーケンス」
- 68 ページの「Solaris 環境でのキーボードサポート」
- 81 ページの「新しい Solaris キーボードのソフトウェアサポート」

---

## 地域対応のためのソフトウェアサポート

この節では、Solaris ロケールパッケージ、CD-ROM ディスク、地域対応機能、およびスクリプト機能について説明します。

### Solaris ロケールパッケージの要約

すべての最新の Solaris ロケールパッケージは、完全ロケールまたは部分ロケールに分類されます。

部分ロケールでは、それぞれのロケールが有効になります。部分ロケールがシステムにインストールされている場合、ユーザーは、対象のロケールでテキストの入力、表示、印刷を行ったり、アプリケーションを実行したりできます。ただし、Solaris OS から出される OS/GUI メッセージは英語です。Solaris SOFTWARE CD には、すべての部分ロケールパッケージが格納されています。日本語およびアジア各国語の部分ロケールはその言語に対応してパッケージ化されていますが、その他の部分ロケールは地理的な地域に対応してパッケージ化されています。

完全ロケールパッケージには、ソフトウェアメッセージの翻訳、オンラインヘルプ、オプションフォント、および言語固有の機能が含まれます。完全ロケールパッケージは、多数の言語に対して完全な言語の機能を提供します。次の言語に基づくロケールはすべて完全ロケールです。

- ドイツ語
- フランス語
- スペイン語
- スウェーデン語
- イタリア語
- 日本語
- 韓国語
- 簡体字中国語
- 繁体字中国語

完全ロケールは言語別にパッケージ化され、LANGUAGES CD に格納されています。

---

注 - 完全ロケールを機能させるためには、部分ロケールパッケージ (ロケールを有効にするもの) をインストールする必要があります。

---

Solaris のインストールプロセス中に、サポートが必要な地域の選択を求められます。インストール終了後に利用できるロケールのサポートは、この段階での選択によって決まります。部分ロケールは Solaris OS とともに Solaris SOFTWARE CD からインストールされ、完全ロケールは LANGUAGES CD からインストールされます。完全ロケールのサポートが必要ない場合は、LANGUAGES CD からのインストールを省略できます。なお、英語ロケールはデフォルトでインストールされます。

---

## サポートされるロケール

次の表に Solaris 環境でサポートされるすべてのロケールを示します。ロケール名は国際的な命名規格に従います。

表 3-1 アジアのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
hi_IN.UTF-8	英語	インド	UTF-8	Hindi (UTF-8) Unicode 4.0
ja	日本語	日本	eucJP <sup>1</sup>	日本語 (EUC)

<sup>1</sup> eucJP は日本語 EUC コードセットを示します。ja\_JP.eucJP ロケールの仕様は UI\_OSF 日本語環境実装規約バージョン 1.1 に、ja ロケールは以前の Solaris リリースで採用されてきた仕様にそれぞれ準拠しています。

表 3-1 アジアのロケール (続き)

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
				JIS X 0201-1976
				JIS X 0208-1990
				JIS X 0212-1990
ja_JP.eucJP	日本語	日本	eucJP	日本語 (EUC)
				JIS X 0201-1976
				JIS X 0208-1990
				JIS X 0212-1990
ja_JP.PCK	日本語	日本	PCK <sup>2</sup>	日本語 (PC Kanji)
				JIS X 0201-1976
				JIS X 0208-1990
ja_JP.UTF-8	日本語	日本	UTF-8	日本語 (UTF-8) Unicode 4.0
ko_KR.EUC	韓国語	韓国	1001	韓国語 (EUC) KS X 1001
ko_KR.UTF-8	韓国語	韓国	UTF-8	韓国語 (UTF-8) Unicode 4.0
th_TH.UTF-8	英語	タイ	UTF-8	タイ語 (UTF-8) Unicode 4.0
th_TH.TIS620	英語	タイ	TIS620.2533	タイ TIS620.2533
zh_CN.EUC	簡体字中国語	PRC	gb2312 <sup>3</sup>	簡体字中国語 (EUC) GB2312-1980
zh_CN.GBK	簡体字中国語	PRC	GBK <sup>4</sup>	簡体字中国語 (EUC)
zh_CN.GB18030	簡体字中国語	PRC	GB18030-2000	簡体字中国語 (GB18030-2000) GB18030-2000
zh_CN.UTF-8	簡体字中国語	PRC	UTF-8	簡体字中国語 (UTF-8) Unicode 4.0
zh_HK.BIG5HK	繁体字中国語	香港	Big5+HKSCS	繁体字中国語 (BIG5+HKSCS)
zh_HK.UTF-8	繁体字中国語	香港	UTF-8	繁体字中国語 (UTF-8) Unicode 4.0
zh_TW.EUC	繁体字中国語	台湾	cns11643	繁体字中国語 (EUC) CNS 11643-1992

<sup>2</sup> PCK は Shift\_JIS (SJIS) ともいいます。

<sup>3</sup> gb2312 は簡体字中国語 EUC コードセットを示します。GB 1988-80 および GB 2312-80 を含みます。

<sup>4</sup> GBK は GB 拡張を示します。これらの拡張にはすべての GB 2312-80 文字、ISO/IEC 10646-1 のすべての Unified Han 文字、および日本語のひらがな文字、カタカナ文字が含まれます。また、中国語、日本語、韓国語の文字セット、および ISO/IEC 10646-1 の多くの文字も含みます。

表 3-1 アジアのロケール (続き)

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
zh_TW.BIG5	繁体字中国語	台湾	BIG5	繁体字中国語 (BIG5)
zh_TW.UTF-8	繁体字中国語	台湾	UTF-8	繁体字中国語 (UTF-8) Unicode 4.0

表 3-2 オーストラリアのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
en_AU.ISO8859-1	英語	オーストラリア	ISO8859-1	英語 (オーストラリア)
en_NZ.ISO8859-1	英語	ニュージーランド	ISO8859-1	英語 (ニュージーランド)

表 3-3 中央アメリカのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
es_CR.ISO8859-1	スペイン語	コスタリカ	ISO8859-1	スペイン語 (コスタリカ)
es_GT.ISO8859-1	スペイン語	グアテマラ	ISO8859-1	スペイン語 (グアテマラ)
es_NI.ISO8859-1	スペイン語	ニカラグア	ISO8859-1	スペイン語 (ニカラグア)
es_PA.ISO8859-1	スペイン語	パナマ	ISO8859-1	スペイン語 (パナマ)
es_SV.ISO8859-1	スペイン語	エルサルバドル	ISO8859-1	スペイン語 (エルサルバドル)

表 3-4 中央ヨーロッパのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
cs_CZ.ISO8859-2	英語	チェコ共和国	ISO8859-2	チェコ語 (チェコ)
de_AT.ISO8859-1	ドイツ語	オーストリア	ISO8859-1	ドイツ語 (オーストリア)
de_AT.ISO8859-15	ドイツ語	オーストリア	ISO8859-15	ドイツ語 (オーストリア、ISO8859-15 - Euro)
de_CH.ISO8859-1	ドイツ語	スイス	ISO8859-1	ドイツ語 (スイス)
de_DE.UTF-8	ドイツ語	ドイツ	UTF-8	ドイツ語 (ドイツ、Unicode 4.0)
de_DE.ISO8859-1	ドイツ語	ドイツ	ISO8859-1	ドイツ語 (ドイツ)

表 3-4 中央ヨーロッパのロケール (続き)

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
de_DE.ISO8859-15	ドイツ語	ドイツ	ISO8859-15	ドイツ語 (ドイツ、ISO8859-15 - Euro)
fr_CH.ISO8859-1	フランス語	スイス	ISO8859-1	フランス語 (スイス)
hu_HU.ISO8859-2	英語	ハンガリー	ISO8859-2	ハンガリー語 (ハンガリー)
pl_PL.ISO8859-2	英語	ポーランド	ISO8859-2	ポーランド語 (ポーランド)
pl_PL.UTF-8	英語	ポーランド	UTF-8	ポーランド語 (ポーランド、Unicode 4.0)
sk_SK.ISO8859-2	英語	スロバキア	ISO8859-2	スロバキア語 (スロバキア)

表 3-5 東ヨーロッパのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
bg_BG.ISO8859-5	英語	ブルガリア	ISO8859-5	ブルガリア語 (ブルガリア)
et_EE.ISO8859-15	英語	エストニア	ISO8859-15	エストニア語 (エストニア)
hr_HR.ISO8859-2	英語	クロアチア	ISO8859-2	クロアチア語 (クロアチア)
lt_LT.ISO8859-13	英語	リトアニア	ISO8859-13	リトアニア語 (リトアニア)
lv_LV.ISO8859-13	英語	ラトビア	ISO8859-13	ラトビア語 (ラトビア)
mk_MK.ISO8859-5	英語	マケドニア	ISO8859-5	マケドニア語 (マケドニア)
ro_RO.ISO8859-2	英語	ルーマニア	ISO8859-2	ルーマニア語 (ルーマニア)
ru_RU.KOI8-R	英語	ロシア	KOI8-R	ロシア語 (ロシア、KOI8-R)
ru_RU.ANSI1251	英語	ロシア	ansi-1251	ロシア語 (ロシア、ANSI 1251)
ru_RU.ISO8859-5	英語	ロシア	ISO8859-5	ロシア語 (ロシア)
ru_RU.UTF-8	英語	ロシア	UTF-8	ロシア語 (ロシア、Unicode 4.0)

表 3-5 東ヨーロッパのロケール (続き)

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
sh_BA.ISO8859-2@bosnia	英語	ボスニア	ISO8859-2	ボスニア語 (ボスニア)
sl_SI.ISO8859-2	英語	スロベニア	ISO8859-2	スロベニア語 (スロベニア)
sq_AL.ISO8859-2	英語	アルバニア	ISO8859-2	アルバニア語 (アルバニア)
sr_YU.ISO8859-5	英語	セルビア	ISO8859-5	セルビア語 (セルビア)
tr_TR.ISO8859-9	英語	トルコ	ISO8859-9	トルコ語 (トルコ)
tr_TR.UTF-8	英語	トルコ	UTF-8	トルコ語 (トルコ、Unicode 4.0)

表 3-6 中東のロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
He	英語	イスラエル	ISO8859-8	ヘブライ語 (イスラエル)

表 3-7 北アフリカのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
ar_EG.UTF-8	英語	エジプト	UTF-8	アラビア語 (エジプト)
Ar	英語	エジプト	ISO8859-6	アラビア語 (エジプト)

表 3-8 北アメリカのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
en_CA.ISO8859-1	英語	カナダ	ISO8859-1	英語 (カナダ)
en_US.ISO8859-1	英語	米国	ISO8859-1	英語 (U.S.A.)
en_US.ISO8859-15	英語	米国	ISO8859-15	英語 (U.S.A., ISO8859-15 - Euro)
en_US.UTF-8	英語	米国	UTF-8	英語 (米国、Unicode 4.0)
fr_CA.ISO8859-1	フランス語	カナダ	ISO8859-1	フランス語 (カナダ)
es_MX.ISO8859-1	スペイン語	メキシコ	ISO8859-1	スペイン語 (メキシコ)

表 3-9 北ヨーロッパのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
da_DK.ISO8859-1	英語	デンマーク	ISO8859-1	デンマーク語 (デンマーク)
da_DK.ISO8859-15	英語	デンマーク	ISO8859-15	デンマーク語 (デンマーク、ISO8859-15-Euro)
fi_FI.ISO8859-1	英語	フィンランド	ISO8859-1	フィンランド語、Unicode 4.0
fi_FI.ISO8859-15	英語	フィンランド	ISO8859-15	フィンランド語 (フィンランド、ISO8859-15-Euro)
fi_FI.UTF-8	英語	フィンランド	UTF-8	フィンランド語 (フィンランド)
is_IS.ISO8859-1	英語	アイスランド	ISO8859-1	アイスランド語 (アイスランド)
no_NO.ISO8859-1@bokmal	英語	ノルウェー	ISO8859-1	ノルウェー語 (ノルウェーブークモール語)
no_NO.ISO8859-1@nyorsk	英語	ノルウェー	ISO8859-1	ノルウェー語 (ノルウェーニーノシク語)
sv_SE.ISO8859-1	スウェーデン語	スウェーデン	ISO8859-1	スウェーデン語 (スウェーデン)
sv_SE.ISO8859-15	スウェーデン語	スウェーデン	ISO8859-15	スウェーデン語 (スウェーデン、ISO8859-15-Euro)
sv_SE.UTF-8	スウェーデン語	スウェーデン	UTF-8	スウェーデン語 (スウェーデン、Unicode 4.0)

表 3-10 南アメリカのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
es_AR.ISO8859-1	スペイン語	アルゼンチン	ISO8859-1	スペイン語 (アルゼンチン)
es_BO.ISO8859-1	スペイン語	ボリビア	ISO8859-1	スペイン語 (ボリビア)
es_CL.ISO8859-1	スペイン語	チリ	ISO8859-1	スペイン語 (チリ)
es_CO.ISO8859-1	スペイン語	コロンビア	ISO8859-1	スペイン語 (コロンビア)
es_EC.ISO8859-1	スペイン語	エクアドル	ISO8859-1	スペイン語 (エクアドル)
es_PE.ISO8859-1	スペイン語	ペルー	ISO8859-1	スペイン語 (ペルー)
es_PY.ISO8859-1	スペイン語	パラグアイ	ISO8859-1	スペイン語 (パラグアイ)
es_UY.ISO8859-1	スペイン語	ウルグアイ	ISO8859-1	スペイン語 (ウルグアイ)

表 3-10 南アメリカのロケール (続き)

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
es_VE.ISO8859-1	スペイン語	ベネズエラ	ISO8859-1	スペイン語 (ベネズエラ)
pt_BR.ISO8859-1	英語	ブラジル	ISO8859-1	ポルトガル語 (ブラジル)
pt_BR.UTF-8	英語	ブラジル	UTF-8	ポルトガル語 (ブラジル、Unicode 4.0)

表 3-11 南ヨーロッパのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
ca_ES.ISO8859-1	英語	スペイン	ISO8859-1	カタロニア語 (スペイン)
ca_ES.ISO8859-15	英語	スペイン	ISO8859-15	カタロニア語 (スペイン、ISO8859-15 - Euro)
el_GR.ISO8859-7	英語	ギリシャ	ISO8859-7	ギリシャ語 (ギリシャ)
es_ES.ISO8859-1	スペイン語	スペイン	ISO8859-1	スペイン語 (スペイン)
es_ES.ISO8859-15	スペイン語	スペイン	ISO8859-15	スペイン語 (スペイン、ISO8859-15 - Euro)
es_ES.UTF-8	スペイン語	スペイン	UTF-8	スペイン語 (スペイン、Unicode 4.0)
it_IT.ISO8859-1	イタリア語	イタリア	ISO8859-1	イタリア語 (イタリア)
it_IT.ISO8859-15	イタリア語	イタリア	ISO8859-15	イタリア語 (イタリア、ISO8859-15 - Euro)
it_IT.UTF-8	イタリア語	イタリア	UTF-8	イタリア語 (イタリア、Unicode 4.0)
pt_PT.ISO8859-1	英語	ポルトガル	ISO8859-1	ポルトガル語 (ポルトガル)
pt_PT.ISO8859-15	英語	ポルトガル	ISO8859-15	ポルトガル語 (ポルトガル、ISO8859-15 - Euro)

表 3-12 西ヨーロッパのロケール

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
en_GB.ISO8859-1	英語	イギリス	ISO8859-1	英語 (イギリス)
en_IE.ISO8859-1	英語	アイルランド	ISO8859-1	英語 (アイルランド)
fr_BE.ISO8859-1	フランス語	ベルギーのワロン	ISO8859-1	フランス語 (ベルギーのワロン、Unicode 4.0)

表 3-12 西ヨーロッパのロケール (続き)

ロケール	ユーザーインタ フェース	地域	コードセット	言語サポート
fr_BE.UTF-8	フランス語	ベルギーのワロン	UTF-8	フランス語 (ベルギーのワロン、Unicode 4.0)
fr_FR.ISO8859-1	フランス語	フランス	ISO8859-1	フランス語 (フランス)
fr_FR.UTF-8	フランス語	フランス	UTF-8	フランス語 (フランス、Unicode 4.0)
nl_BE.ISO8859-1	英語	ベルギーのフラマン	ISO8859-1	ドイツ語 (ベルギーのフラマン)
nl_NL.ISO8859-1	英語	オランダ	ISO8859-1	ドイツ語 (オランダ)

## ロケールの複数キーの Compose キーシーケンス

Solaris ロケールの多く、とりわけヨーロッパや Unicode のロケールでは、いわゆる「デッドキーシーケンス」(Compose キーシーケンスともいう)を使ってさまざまな文字を入力できます。

Compose キーシーケンスは、キーボードのキー表面に示されていない発音記号などの文字を入力するときに使用します。

次の表に、Compose キーシーケンスのいくつかの例を示します。Compose キーシーケンスの詳細については、123 ページの「英語/ヨーロッパ言語入力モード」を参照してください。

表 3-13 Compose キーで入力する発音記号の例

記号	Compose キーとの組み合わせ	例
Dieresis		Compose A " → A と diaeresis
Caron	v	Compose Z v → Z と caron
Breve	u	Compose G u → G と breve
Ogonek	a	Compose A a → A と Ogonek
Cedilla	,	Compose K , → K と cedilla
登録標識	R O	Compose R O → 登録商標

表 3-13 Compose キーで入力する発音記号の例 (続き)

記号	Compose キーとの組み合わせ	例
感嘆符の反転	!!	Compose !! → 感嘆符の反転

注 - ロケールのコードセットに対応する文字がない場合は、Compose キーシーケンスを使用できません。たとえば、ISO8859-1 コードセットには caron 付きの Z がないため、en\_US.ISO8859-1 ロケールで caron 付きの Z を入力できません。

## Solaris 環境でのキーボードサポート

特定地域向けのさまざまな配列のキーボードが SPARC プラットフォームおよび x86 プラットフォームでサポートされています。Solaris OS は、次の表に示す地域向けのキーボードをサポートします。

表 3-14 地域向けキーボードのサポート

領域	国	Sun キーボード (Type 4/5/5c)	Sun キーボード (Type 6)	PC キーボード
アジア	日本	X	X	X
	韓国	X	X	X
	台湾	X	X	X
ヨーロッパ	ベルギー	X	X	X
	チェコ共和国	X		X
	デンマーク	X	X	X
	フィンランド		X	
	フランス	X	X	X
	ドイツ	X	X	X
	イギリス	X	X	X
	ギリシャ	X		X
	ハンガリー	X		X
イタリア	X	X	X	

表 3-14 地域向けキーボードのサポート (続き)

領域	国	Sun キーボード (Type 4/5/5c)	Sun キーボード (Type 6)	PC キーボード
	ラトビア	X		X
	リトアニア	X		X
	オランダ	X	X	X
	ノルウェー	X	X	X
	ポーランド	X		X
	ポルトガル	X	X	X
	ロシア	X	X	X
	スペイン	X	X	X
	スウェーデン	X	X	X
	スイス (フランス語)	X	X	X
	スイス (ドイツ語)	X	X	X
	トルコ	X	X	X
アメリカ	カナダ (フランス語)	X	X	X
	ラテンアメリカ (スペイン語)	X		
	米国	X	X	X
中東	アラビア語	X	X	

国際標準に準拠するキー配列を使用する中国などの地域では、米国向けのキー配列サポートに基づいてローカルの文字を入力します。基本的なキーボードマッピングはどちらも同じです。日本、トルコ、スイスなど一部の国では、複数の言語が使用されていたり、複数のキー配列が存在するため、複数のキーボードが使用されます。

Sun の Type 4、5、5c キーボードでは、Mini DIN 8 ピン接続に基づく Sun 入出力インタフェースが使用されます。Sun の Type 6 キーボードには、次の 2 種類のインタフェースがあります。

- Mini DIN 8-ピン接続による Sun 入出力
- USB

キーボードタイプは、Sun のキーボードの裏面に印刷されています。

PC キーボードのインタフェースには、PS/2 や USB など、さまざまなものがあります。

## SPARC システムでのキーボードの変更

Sun のほとんどの Type 4、5、5c キーボードでは、DIP スイッチの設定によって Solaris システムのキー配列を変更できます。DIP スイッチの設定で配列を変更できるキーボードのタイプ、名称、対応する配列 ID は、  
/usr/openwin/share/etc/keytables/keytable.map ファイルに記載されています。

---

注 - Type 6 キーボードの裏面には DIP スイッチがないため、このタイプのキーボードの配列を変えることはできません。米国、米国/UNIX、日本語キーボードなどの一部の Type 5 および 5c キーボードには、DIP スイッチの代わりにジャンパがあります。xmodmap (1) などのユーティリティは別として、SPARC プラットフォームおよび x86 プラットフォームでは、キーボードを変更するユーティリティやツールは提供されていません。

---

次の表に、Type 4、5、および 5c キーボードの配列 ID 値を示します (1 = スイッチ Up、0 = スイッチ Down)。

表 3-15 Type 4、5、5c キーボードの配列

DIP スイッチ	キーボード (キーテーブルファイル)	設定 (2 進数)
0	米国 (US4.kt)	000000
1	米国 (US4.kt)	000001
2	ベルギー (FranceBelg4.kt)	000010
3	カナダ (Canada4.kt)	000011
4	デンマーク (Denmark4.kt)	000100
5	ドイツ (Germany4.kt)	000101
6	イタリア (Italy4.kt)	000110
7	オランダ (Netherland4.kt)	000111
8	ノルウェー (Norway4.kt)	001000
9	ポルトガル (Portugal4.kt)	001001
10 (0x0a)	ラテンアメリカ/スペイン語 (SpainLatAm4.kt)	001010
11 (0x0b)	スウェーデン (SwedenFin4.kt)	001011
12 (0x0c)	スイス/フランス語 (Switzer_Fr4.kt)	001100
13 (0x0d)	スイス/ドイツ語 (Switzer_Ge4.kt)	001101
14 (0x0e)	イギリス (UK4.kt)	001110

表 3-15 Type 4、5、5c キーボードの配列 (続き)

DIP スイッチ	キーボード (キーテーブルファイル)	設定 (2 進数)
16 (0x10)	韓国 (Korea4.kt)	010000
17 (0x11)	台湾 (Taiwan4.kt)	010001
23	ロシア語	100001
33 (0x21)	米国 (US5.kt)	100111
34 (0x22)	米国./UNIX (US_UNIX5.kt)	100010
35 (0x23)	フランス (France5.kt)	100011
36 (0x24)	デンマーク (Denmark5.kt)	100100
37 (0x25)	ドイツ (Germany5.kt)	100101
38 (0x26)	イタリア (Italy5.kt)	100110
39 (0x27)	オランダ (Netherland5.kt)	100111
40 (0x28)	ノルウェー (Norway5.kt)	101000
41 (0x29)	ポルトガル (Portugal5.kt)	101001
42 (0x2a)	スペイン (Spain5.kt)	101010
43 (0x2b)	スウェーデン (Sweden5.kt)	101011
44 (0x2c)	スイス/フランス語 (Switzer_Fr5.kt)	101101
45 (0x2d)	スイス/ドイツ語 (Switzer_Ge5.kt)	101110
46 (0x2e)	イギリス (UK5.kt)	101111
47 (0x2f)	韓国 (Korea5.kt)	101111
48 (0x30)	台湾 (Taiwan5.kt)	110000
49 (0x31)	日本 (Japan5.kt)	110001
50 (0x32), 63 (0x3f)	カナダ/フランス語 (Canada_Fr5.kt)	110010
51 0(x33)	ハンガリー (Hungary5.kt)	110011
52 (0x34)	ポーランド (Poland5.kt)	110100
53 (0x35)	チェコ (Czech5.kt)	110101
54 (0x36)	ロシア (Russia5.kt)	110110
55 (0x37)	ラトビア (Latvia5.kt)	110111
56 (0x38), 62 (0x3e)	トルコ-Q5 (TurkeyQ5.kt)	111000
57 (0x39)	ギリシャ (Greece5.kt)	111001
58 (0x3a)	アラブ (Arabic5.kt)	111011

表 3-15 Type 4、5、5c キーボードの配列 (続き)

DIP スイッチ	キーボード (キーテーブルファイル)	設定 (2 進数)
59 (0x3b)	リトアニア (Lithuania5.kt)	111010
60 (0x3c)	ベルギー (Belgian5.kt)	111100
62 (0x3e)	カナダ/フランス語 (Canada_Fr5_TBITS5.kt)	111111
	フランス語系カナダ	
	ポーランド語プログラマ	
	エストニア語	

4 が付いたキーテーブルファイル名は Type 4 キーボードを、5 が付いたキーテーブルファイル名は Type 5 キーボードをそれぞれ表します。

## ▼ キー配列のチェコ配列への変更方法

- この表または `/usr/openwin/share/etc/keytables/keytable.mp` ファイルで正しい **DIP スイッチ ID (配列 ID)** を確認します。**keytable.mp** ファイルの配列 ID 値は 10 進数で表されています。  
チェコの場合、配列 ID は 10 進数の 53 です (16 進数では 0x35)。
- 配列 ID を 2 進数に変換するか、上の表の設定値を使用します。この変換には、**dtcalc (1)** などの計算ユーティリティが使用できます。  
チェコキーボードの場合、2 進数の値は 110101 です。
- システムをシャットダウンし、電源を切ります。
- キーボード裏面の **DIP スイッチ** の設定を手順 2 の 2 進数に変更します。  
左側が最初の DIP スイッチです。1 の場合はスイッチを Up に、0 の場合はスイッチを Down にします。  
したがって、チェコキーボードの 2 進数 110101 は Up Up Down Up Down Up になります。
- 電源を入れ、システムを起動します。

---

注 - Type 4 キーボードとは異なり、Type 5、5c キーボードには、DIP スイッチが 5 つしかありません。Type 5、5c キーボードの場合は、2 進数の最初の桁を無視してください。たとえば、チェコ Type 5c キーボードの場合は後ろの 5 桁 (10101) だけを使用するため、DIP スイッチの設定は Up Down Up Down Up になります。

---

## Intel システムでのキーボードの変更

Intel アーキテクチャの場合、インストール時の kdmconfig(1M) の箇所ではキーボードを選択します。インストール後にその設定を変更するには、GUI デスクトップ環境を終了してコマンド行モードにします。スーパーユーザーとして kdmconfig を実行します。説明に従って、必要なキー配列を選択します。

### キーボードの配列

次の図にアラビア語キーボードを示します。

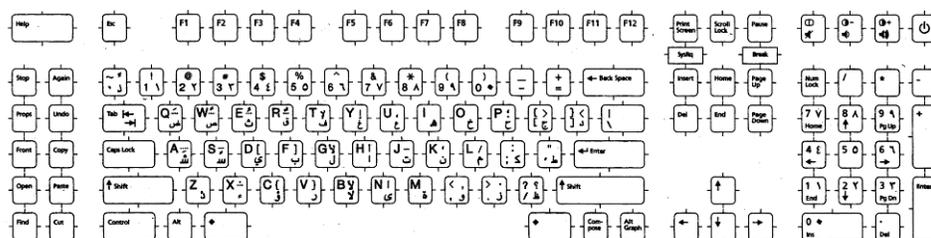


図 3-1 アラビア語キーボード

次の図にベルギー向けキーボードを示します。

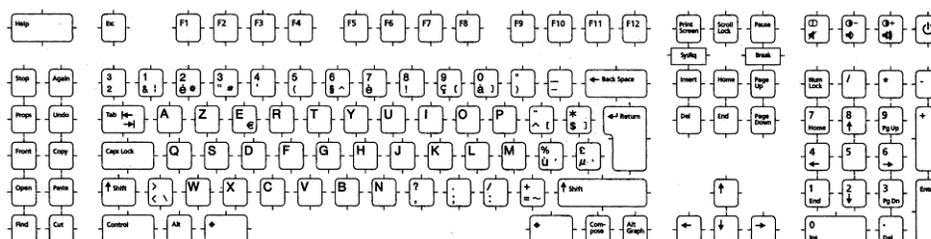


図 3-2 ベルギー向けキーボード

次の図にキリル文字キーボードを示します。

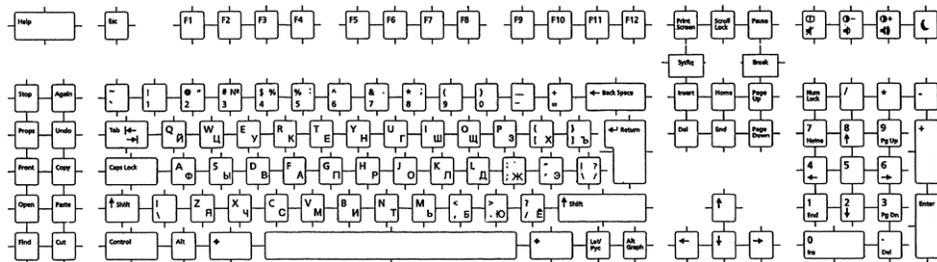


図 3-3 キリル文字 (ロシア語) キーボード

次の図にデンマーク語キーボードを示します。

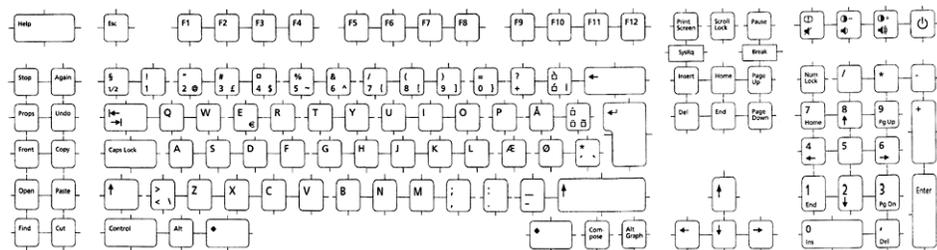


図 3-4 デンマーク語キーボード

次の図にフィンランド語キーボードを示します。

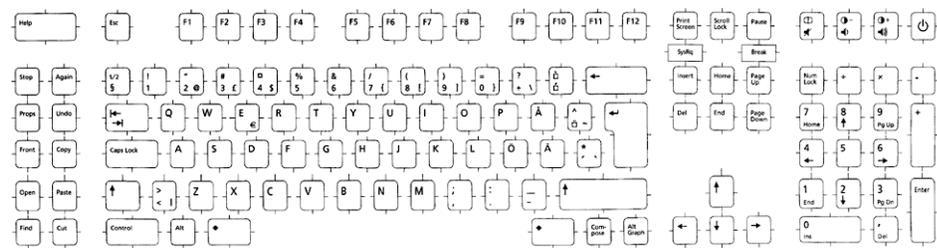


図 3-5 フィンランド語キーボード

次の図にフランス語キーボードを示します。

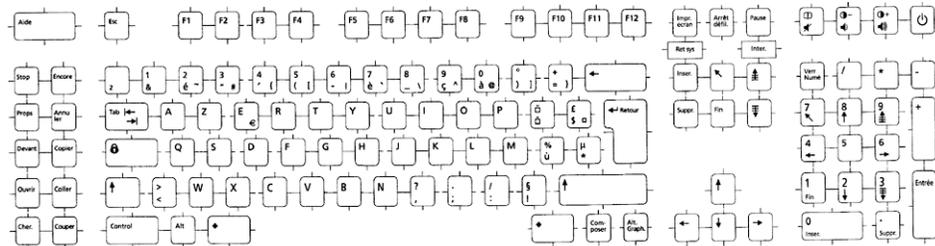


図 3-6 フランス語キーボード

次の図にドイツ語キーボードを示します。

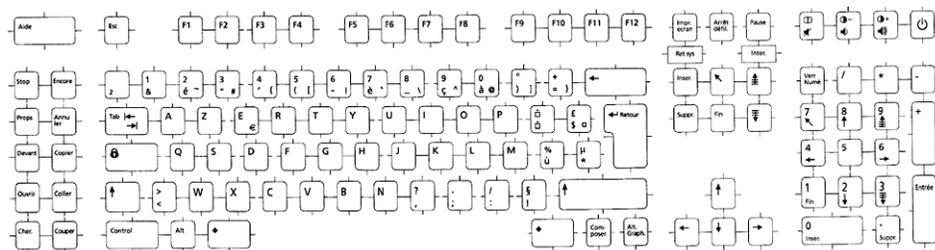


図 3-7 ドイツ語キーボード

次の図にイタリア語キーボードを示します。

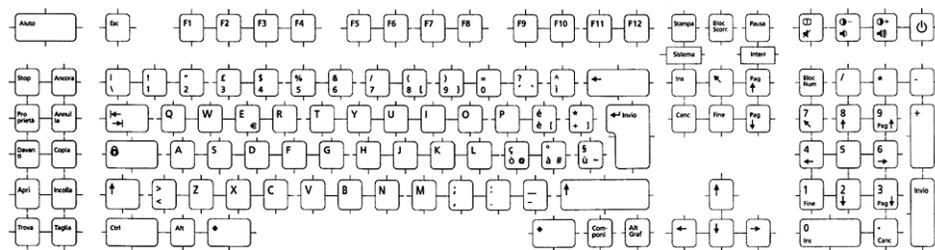


図 3-8 イタリア語キーボード

次の図に日本語キーボードを示します。

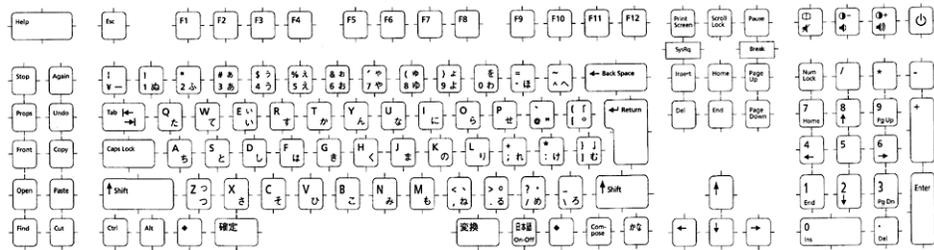


図 3-9 日本語キーボード

次の図に韓国語キーボードを示します。

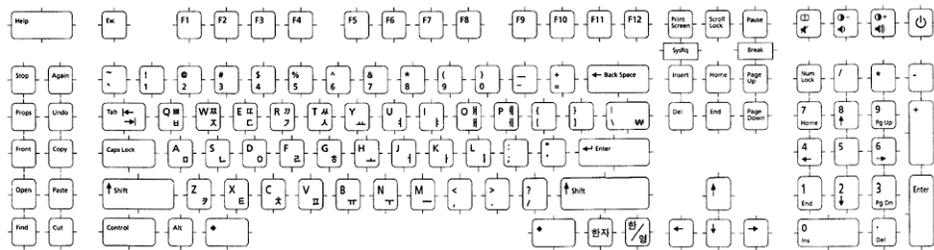


図 3-10 韓国語キーボード

次の図にオランダ (オランダ語) キーボードを示します。

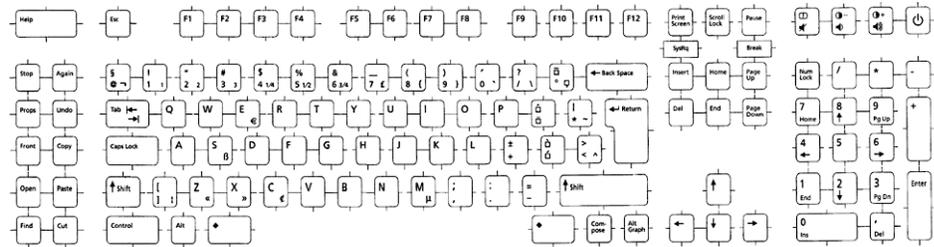


図 3-11 オランダ (オランダ語) キーボード

次の図にノルウェー語キーボードを示します。

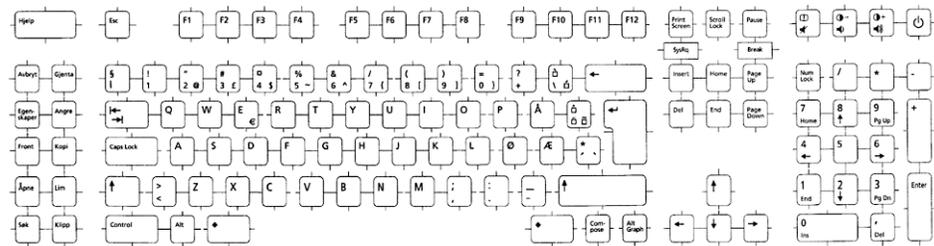


図 3-12 ノルウェー語キーボード

次の図にポルトガル語キーボードを示します。

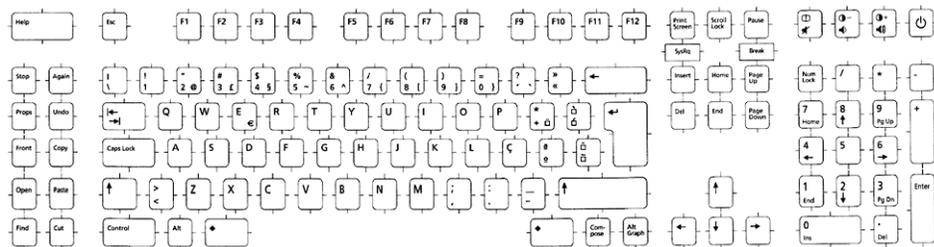


図 3-13 ポルトガル語キーボード

次の図にスペイン語キーボードを示します。

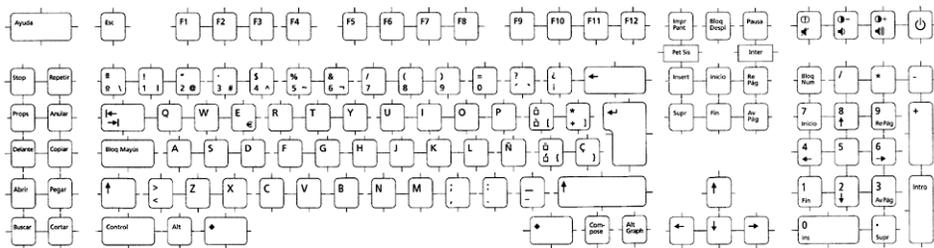


図 3-14 スペイン語キーボード

次の図にスウェーデン語キーボードを示します。

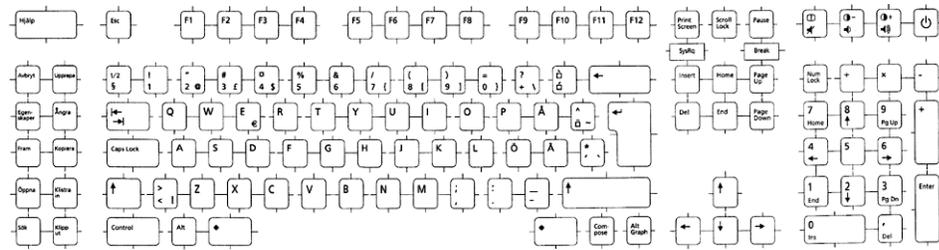


図 3-15 スウェーデン語キーボード

次に図にスイス (フランス語) キーボードを示します。

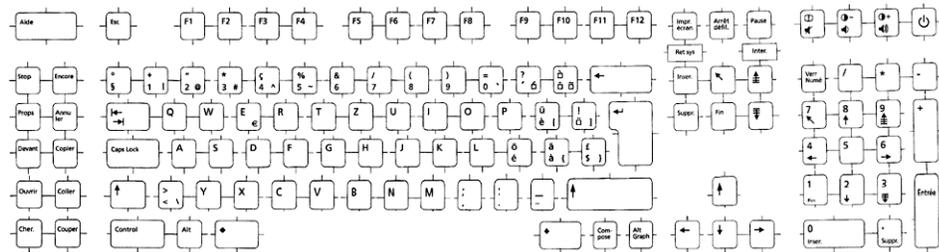


図 3-16 スイス (フランス語) キーボード

次に図にスイス (ドイツ語) キーボードを示します。

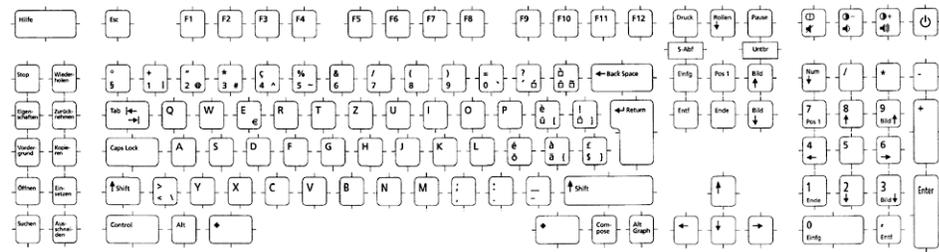


図 3-17 スイス (ドイツ語) キーボード

次に図に繁体字中国語キーボードを示します。

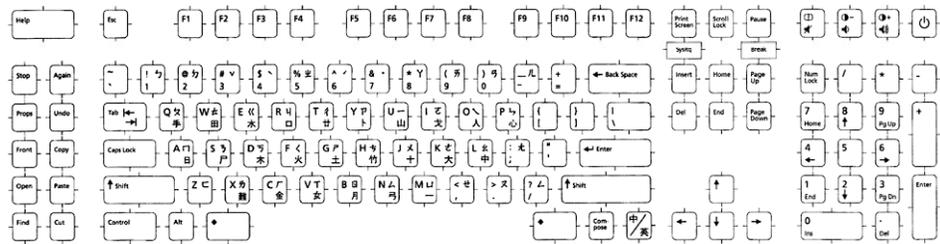


図 3-18 繁体字中国語キーボード

次の図にトルコ語 F キーボードを示します。

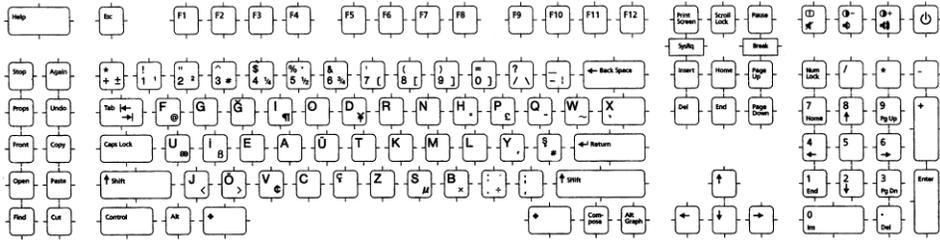


図 3-19 トルコ語 F キーボード

次の図にトルコ語 Q キーボードを示します。

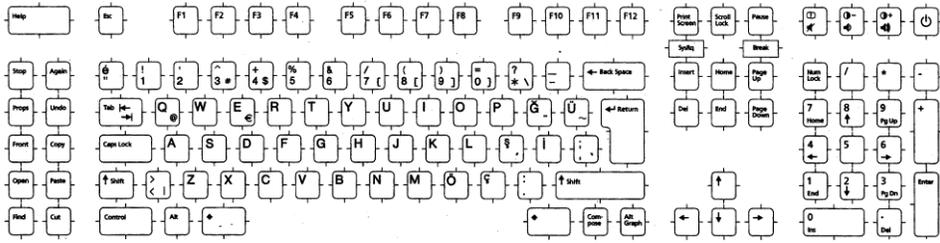


図 3-20 トルコ語 Q キーボード

次の図にイギリス向けキーボードを示します。

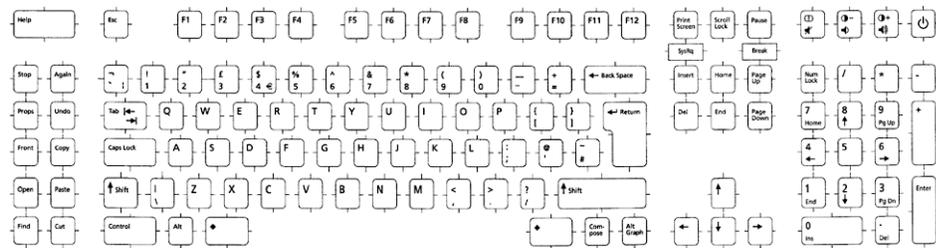


図 3-21 イギリス向けキーボード

次の図に米国向けキーボードを示します。

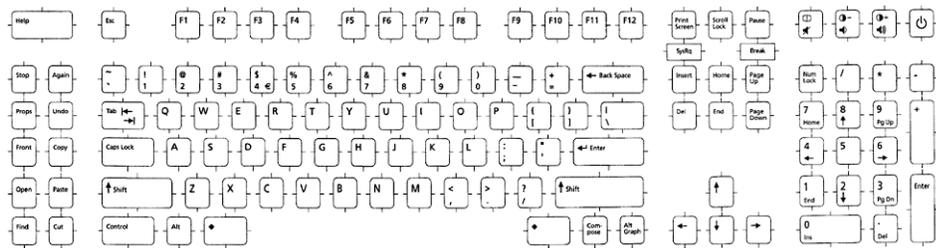


図 3-22 米国向けキーボード

次の図に米国/UNIXキーボードを示します。

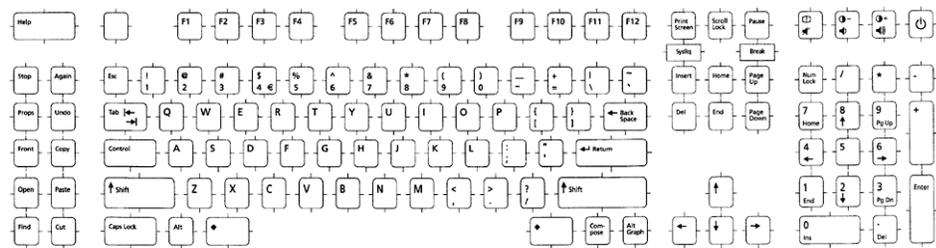


図 3-23 米国/UNIX キーボード

---

## 新しい Solaris キーボードのソフトウェアサポート

このリリースでは、次の追加キーボードのためのソフトウェアサポートがあります。

- ロシア語 Type 6 USB キーボード
- エストニア語 Type 6 USB キーボード
- フランス語系カナダ Type 6 USB キーボード
- ポーランド語プログラマの Type 5 キーボード

ロシア、カナダ、エストニア、およびポーランドのユーザーは、ソフトウェアによって標準の米国向けキー配列を変更してそれぞれの言語の要件に合わせます。現在、追加キーボードタイプ用のハードウェアはありません。新しいキーボードソフトウェアを利用するには、この節に示す手順に従ってください。

### ▼ エストニア語 Type 6 USB キーボードサポートの使用方法

1. `/usr/openwin/share/etc/keytables/keytable.map` ファイルの `US6.kt` のエントリを `Estonia6.kt` に変更します。

変更後のエントリは次のようになります。

```
6          0          Estonia6.kt
```

2. `/usr/openwin/share/lib/locale/iso_8859_15/Compose` ファイルに次のエントリのいずれかを追加します。

変更後のエントリは次のようになります。

```
<scaron>    : "/xa8"    scaron
<scaron>    : "/xa6"    scaron
<scaron>    : "/270"    scaron
<scaron>    : "/264"    scaron
```

3. システムを再起動して変更を適用します。

### ▼ フランス語系カナダ Type 6 USB キーボードサポートの使用方法

1. `/usr/openwin/share/etc/keytables/keytable.map` ファイルの `US6.kt` のエントリを `Canada6.kt` に変更します。

変更後のエントリは次のようになります。

```
6          0      Canada6.kt
```

2. システムを再起動して変更を適用します。

## ▼ ポーランド語プログラマ Type 5 キーボードサポートの使用方法

1. `/usr/openwin/share/etc/keytables/keytable.map` ファイルの `Poland5.kt` のエントリを `Poland5_pr.kt` に変更します。  
変更後のエントリは次のようになります。

```
6          0      Poland5_pr.kt
```

2. システムを再起動して変更を適用します。

## 第 4 章

---

# サポートされるアジアのロケール

---

この章では、日本語、インド語、およびタイ語の地域対応に関連する情報について説明します。この章の内容は次のとおりです。

- 83 ページの「日本語の地域対応」
- 87 ページの「インド語の地域対応」
- 114 ページの「タイ語の地域対応」

---

## 日本語の地域対応

この節では、日本語ロケール固有の情報について説明します。

### 日本語ロケール

最新の Solaris 環境では、異なる文字エンコーディングをサポートする 4 種類の日本語ロケールを使用できます。ja と ja\_JP.eucJP ロケールは、日本語 EUC に基づきます。ja\_JP.eucJP ロケールの仕様は UI-OSF 日本語環境実装規約バージョン 1.1 に、ja ロケールは以前の Solaris リリースで採用されてきた仕様にそれぞれ準拠しています。ja\_JP.PCK ロケールは、Shift\_JIS として知られている PC-Kanji コードに基づいています。ja\_JP.UTF-8 は、UTF-8 に基づいています。

日本語 EUC とその文字セットの対応付けについては、eucJP(5) のマニュアルページを、PC-Kanji コードとその文字セットの対応付けについては、PCK(5) のマニュアルページをそれぞれ参照してください。

### 日本語文字セット

サポートされている日本語文字セットは以下のとおりです。

- JIS X 0201-1976

- JIS X 0208-1990
- JIS X 0212-1990
- JIS X 0213-2000 (Unicode 4.0 で定義されている文字のみ)

JIS X 0212-1990 は ja\_JP.PCK ロケールではサポートされていません。JIS X 0213-2000 は ja\_JP.UTF-8 ロケールでのみサポートされています。JIS X 0213-2000 で定義されているすべての文字が使用可能であるとは限りません。使用できるのは、Unicode 4.0 文字セットに定義されている文字だけです。

ベンダー定義文字 (VDC) とユーザー定義文字 (UDC) もサポートされています。VDC は、JIS X 0208-1990 または JIS X 0212-1990 の使用しない (予約) コードポイントを占めます。UDC は、VDC に対して割り当てられているコードポイントを除いて、VDC と同じコードポイントを占めます。

## 日本語フォント

日本語フォント形式としては、ビットマップ、TrueType、Type1 の3つがサポートされています。日本語 Type1 フォントは印刷用の JIS X 0212 のみです。また、Type1 フォントは、UDC で使用するフォント形式です。

次の表に、日本語ビットマップフォントを示します。

表 4-1 日本語ビットマップフォント

完全なファミリ名	サブファミリ	表記形式	ベンダー	エンコーディング
sun gothic	R、B	PCF(12,14,16,20,24)		JIS X 0208-1983、 JIS X 0201-1976
sun minchou	R	PCF(12,14,16,20,24)		JIS X 0208-1983、 JIS X 0201-1976
ricoh hg gothic b	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0208-1983、JIS X 0201-1976
ricoh hg mincho l	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0208-1983、JIS X 0201-1976
ricoh gothic	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0212-1990、JIS X 0213-2000
ricoh mincho	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0212-1990、JIS X 0213-2000
ricoh heiseimin	R	PCF(12,14,16,18,20,24)	RICOH	JIS X 0212-1990

次の表に、日本語 True Type フォントを示します。

表 4-2 日本語 TrueType フォント

完全なファミリー名	サブファミリー	表記形式	ベンダー	エンコーディング
ricoh hg gothic b	固定幅	TrueType	RICOH	JIS X 0208-1983、JIS X 0201-1976
ricoh hg mincho l	固定幅	TrueType	RICOH	JIS X 0208-1983、JIS X 0201-1976
ricoh hg gothicb sun	固定幅、プロポーションナル	TrueType	RICOH	JIS X 0201-1976、JIS X 0208-1983、JIS X 0213-2000
ricoh hg minchol sun	固定幅、プロポーションナル	TrueType	RICOH	JIS X 0201-1976、JIS X 0208-1983、JIS X 0213-2000
ricoh heiseimin	固定幅	TrueType	RICOH	JIS X 0212-1990

## 日本語入力方式

最新の Solaris 環境のデフォルト日本語入力方式は ATOK12 です。ATOK12 は、日本語ロケールがインストールされていれば、日本語ロケールとすべての UTF-8 ロケールでこの入力方式が使用できます。その他、Wnn6 日本語入力方式がすべての日本語ロケールで使用できます。入力方式の切り替えはデスクトップメニューから行うことができます。日本語 Solaris 1.x BCP サポートでは、kkcv 日本語入力方式を使用できません。

ATOK12 入力方式によって日本語テキストを入力するには次のようにします。

### ▼ ATOK 入力方式の使用法

1. **Control-Space** キーを押して入力変換をオンにします。
2. 変換するテキストのかな文字を入力します。  
たとえば、漢字変換のかな文字を入力します。
3. **Space** キーを押すと、入力したかな文字に対する漢字変換候補が表示されます。
4. 選択する変換候補の番号を入力します。
5. **Return** キーを押すと、かな文字全体が漢字混じりのテキストとして確定されます。  
または、下矢印キーを押すと、選択した文字のみが確定されます。
6. **Control-Space** キーを押して入力変換をオフにします。

## 日本語端末用の端末設定

日本語ロケールを文字ベース端末 (TTY) で使うには、端末設定を使って、行編集作業を適切に実行する必要があります。

- 端末が CDE 端末エミュレータ (dtterm) の場合、すべての日本語ロケール (ja、ja\_JP.PCK、または ja\_JP.UTF-8) で、引数 -defeucw を付けて stty(1) を使用します。たとえば、ja ロケールでは次のように入力します。

```
% setenv LANG ja
% stty defeucw
```

- 端末が CDE 端末エミュレータではないが、端末のコードセットが現在のロケールのものと同じである場合は、stty(1) の引数として -defeucw を指定します。
- 端末のコードセットが現在のロケールと一致しない場合、setterm(1) を使ってコード変換を有効にします。たとえば、ロケール ja を使用しているときに PCK (Shift\_JIS コード) が必要になった場合は、次のように入力します。

```
% setenv LANG ja
% setterm -x PCK
```

詳細については、setterm(3CURSES) のマニュアルページを参照してください。

## 日本語 iconv モジュール

いくつかの日本語コードセット変換が iconv(1) と iconv(3) でサポートされています。詳細については、iconv\_ja(5) のマニュアルページを参照してください。

## ユーザー定義文字のサポート

ユーザー定義文字ユーティリティ sdtudctool は、アウトラインフォント (Type1) とビットマップフォント (PCF) を両方ともサポートします。また、以前のリリースにあった、fontedit、type3creator、fontmanager のような古いユーティリティによって作成された UDC フォントを移行するために、いくつかのユーティリティを利用できます。

## 部分ロケールと完全ロケールの違い

次のコンポーネントは、LANGUAGES CD を使用した日本語完全ロケール環境でのみ使用できます。

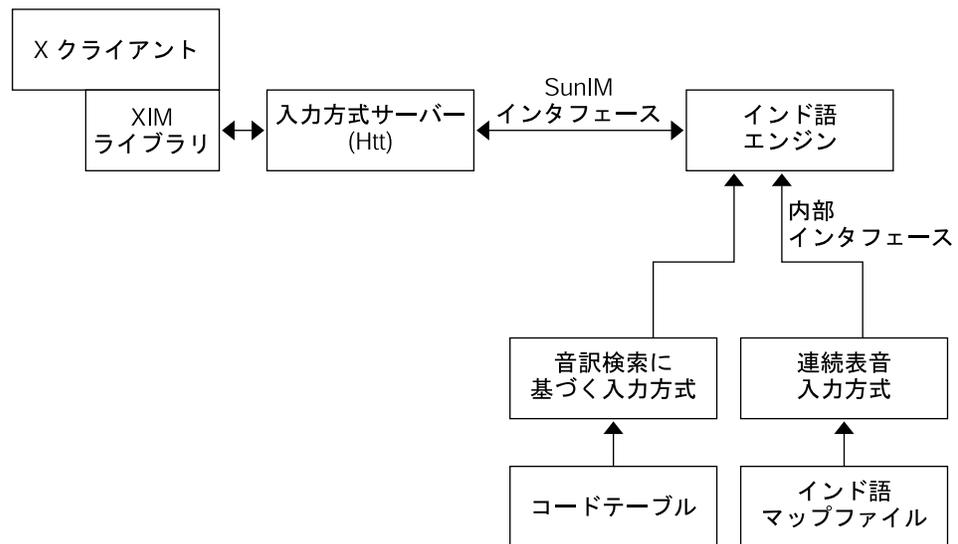
- 翻訳されたメッセージ、ヘルプ、マニュアルページ
- Wnn6 日本語入力方式
- 日本語 Solaris 1.x BCP サポート
- 明朝およびゴシックタイプフェイスのフォント

- 印刷用の JIS X 0212 Type1 フォント
- 日本語固有のダムプリンタおよび postprint サポート
- 古い日本語ユーティリティ (Kanji (1) など)

## インド語の地域対応

UTF-8 ロケールでサポートされているすべてのインド語で、音訳検索に基づく入力方式 (Shabdalipi) および連続表音入力方式が使用できます。この入力方式と仮想キーボードによって、すべての CDE アプリケーションでインド語テキストを入力できます。

次のデータフローは、インド語の入カプロセスの動作を示します。



### ▼ インド語入力方式の使用方法

1. 入力状態領域をクリックして、入力方式の選択メニューを表示します。
2. メニューから入力方式を選択します。  
 または、F6 キーを押すと、使用可能な入力方式の中から選択できます。  
 また、Compose-hi キーシーケンスを入力すると、前に使用した入力方式を選択しません。

3. **F5** キーを押して、使用するインド語スクリプトを選択します。
  - a. キーボードベース (インド語 **INSCRIPT** キーボード) の入力方式の場合、[88 ページの「インド語キーボード」](#) に示すキーボードイメージを使用します。
  - b. 音訳検索に基づく入力方式の場合、対象のスクリプトの文字に対応する最初の英語表音同等文字を入力します。  
 ルックアップウィンドウに表示されるリストから選択します。
  - c. 連続表音入力方式の場合、英語表音同等文字を連続して入力します。  
 対象のスクリプトの対応する文字が編集前テキストとして表示され、編集前テキストをそのままにして入力続けるか、明示的に確定するとその文字が確定されます。連続表音入力方式の場合の英語トークンから UTF-8 コードポイントへのマッピングについては、[92 ページの「連続表音ベース入力方式のマッピング」](#) に示す図を参照してください。
4. **Control-Space** キーを押して、英語/ヨーロッパ語入力モードに戻ります。  
 または、状態領域をクリックし、入力モードの選択ウィンドウから英語/ヨーロッパ語入力モードを選択します。

## インド語キーボード

インド語入力方式で使用可能なキーボードの配列を以下の図に示します。

次の図は Bengali キーボードです。



次の図は Devanagari キーボードです。



次の図は Gujarati キーボードです。



次の図は Gurmukhi キーボードです。





次の図は Teluga キーボードです。



## マッピングについて

92 ページの「連続表音ベース入力方式のマッピング」のイメージは、サポートされる各対象スクリプトにおける英語トークンとその同等コードポイント間のマッピングを示しています。CONSONANT カテゴリは、英語トークンとスクリプトの子音間のマッピングです。VOWEL カテゴリは、英語トークンとスクリプトの母音間のマッピングです。OTHER カテゴリは、子音および母音のプロパティを示さない文字のマッピングです (これらの文字の形は前後の文字によって変化しない)。

CONSONANT、VOWEL、および OTHER のキーワードは、これらの文字が Unicode 規格の一部であることも意味します。セクション SPECIAL CONSONANT、SPECIAL VOWEL、または SPECIAL OTHER は、原則的にそれぞれの文字が子音、母音、またはその他のプロパティを表すが、正式に Unicode 規格の一部であるわけではなく、フォント依存であることを意味します。これらの文字には、Unicode の私用領域 (PUA) のコードポイント値が割り当てられます。Solaris UTF-8 ロケールではサポートされますが、別のプラットフォームではマッピングが機能しない場合もあります。

そのマップファイルは、システムのマップファイルと同じではなく、上記の説明に沿って不必要なキーワードを削除するために少し編集されています。

VOWEL と SPECIAL VOWEL セクションでは、同じ英語トークンに対してコンテキスト依存の独立形と依存形が示されます。113 ページの「連続表音入力方式の仕組み」を参照してください。

malayalam スクリプトには、特別な 'CHILLU' セクションが含まれていますが、実際は SPECIAL OTHER カテゴリです。

## 連続表音ベース入力方式のマッピング

対象のインド語スクリプトにおける英語から表音同等文字への既存のマッピングを以下の図に示します。これらの図を参照しながら、使用するスクリプトのすべてのマッピングを覚えてください。ここではマッピングを直感的に示しているため、ほとんどの文字を図で探すことなく入力できます。

---

注- これらのマッピングでは、マッピングの一部に含まれている「.」、「|」などの特殊文字は「\」文字でエスケープします。エスケープしない場合、「|」文字は複数のトークンが同じ UTF-8 文字を表すときに区切り文字になります。

---

図 4-1、図 4-2、および図 4-3 は、子音、母音、およびその他についての、英語と Bengali のマッピングです。

### CONSONANT

k	ক	Dh	ঢ	r	র
kh	খ	N	ণ	l	ল
g	গ	th	ত	sh	শ
gh	ঘ	thh	থ	S	ষ
nng	ঙ	d	দ	s	স
ch	চ	dh	ধ	h	হ
chh	ছ	n	ন	rr\.	ড়
j	জ	p	প	rh\.	ঢ়
jh	ঝ	f   ph	ফ	y\.	য়
ny   yn	ঞ	b	ব	v	ব
T	ট	bh	ভ	V	ব
Th	ঠ	m	ম		
D	ড	y	য		

図 4-1 Bengali の子音のマッピング

VOWEL	依存形	独立形
a	অ	
aa	আ	আ
i	ই	ই
ee   ii	ঐ	ঐ
u	উ	উ
oo   uu	ঊ	ঊ
r \ ^	ঋ	ঋ
rr \ ^	ঌ	ঌ
n \ ^	৳	৳
nn \ ^	৴	৴
e	এ	এ
ai	আই	আই
o	ও	ও
au	ঔ	ঔ

図 4-2 Bengali の母音のマップ

OTHER	
UM	◌̣
\.N	◌̣̣
M	◌̣̣̣
H	◌̣̣̣̣
OU	◌̣̣̣̣̣
Rs	◌̣̣̣̣̣̣
Rs \.	◌̣̣̣̣̣̣̣

図 4-3 Bengali のその他のマップ

図 4-4、図 4-5、および図 4-6は、子音、母音、およびその他についての、英語と Gujarati のマッピングです。

CONSONANT

k	ક	D	ડ	m	મ
kh	ખ	Dh	ઢ	y	ય
g	ગ	N	ણ	r	ર
gh	ઘ	th	ત	l	લ
ng	ંગ	thh	થ	zh	ઞ
c	ચ	d	દ	w   v	વ
ch	છ	dh	ધ	S	શ
j	જ	n	ન	s	ષ
jh	ઝ	p	પ	sh	સ
ny	ન્ય	ph	ફ	h	હ
T	ટ	b	બ		
Th	ઠ	bh	ભ		

図 4-4 Gujarati の子音のマップ

VOWEL	依存形	独立形
a	અ	
aa	આ	૨
i	ઇ	૩
ee	ઈ	૪
u	ઉ	૫
oo uu	ઊ	૬
r\^	ઋ	૭
e	એ	૮
E	૨ે	૯
ai	૨ૈ	૧૦
o	૨ૌ	૧૧
O	૨૒	૧૨
au	૨૓	૧૩

図 4-5 Gujarati の母音のマップ

OTHER	૨
NG	૩
M	૪
H	૫
OM	૬
RR	૭
kt	૮
av	૯

図 4-6 Gujarati のその他のマップ

図 4-7、図 4-8、および図 4-9 は、子音、母音、およびその他についての、英語と Gurmukhi のマッピングです。

CONSONANT

k	ਕ	Dh	ਢ	r	ਰ
kh	ਖ	N	ਣ	l	ਲ
g	ਗ	t	ਤ	ll	ਲ਼
gh	ਘ	th	ਥ	v	ਵ
ny	ਙ	d	ਦ	sh	ਸ਼
ch	ਚ	dh	ਧ	s	ਸ
chh	ਛ	n	ਨ	h	ਹ
j	ਜ	p	ਪ	khh	ਖ਼
jh	ਝ	ph	ਫ	ghh	ਗ਼
nj	ਞ	b	ਬ	z	ਜ਼
T	ਟ	bh	ਭ	rr	ਰ਼
Th	ਠ	m	ਮ	f	ਫ਼
D	ਡ	y	ਯ		

図 4-7 Gurmukhi の子音のマップ

VOWEL	依存形	独立形
a	ਅ	
aa	ਆ	ੴ
i	ਇ	ਿ
ee   ii	ਈ	ੀ
u	ਉ	ੁ
oo   uu	ਊ	ੂ
E	ਏ	ਏ
ai	ਐ	ਐ
O	ਓ	ਓ
au	ਔ	ਔ

図 4-8 Gurmukhi の母音のマップ

OTHER	
um	ੰ
\.N	ੰ
UH	ੰ
AD	ੰ
IR	ੲ
UR	ੳ
OM	ੴ

図 4-9 Gurmukhi のその他のマップ

図 4-10、図 4-11、および図 4-12 は、子音、母音、およびその他についての、英語と Hindi のマッピングです。

CONSONANT

k	क	t	त	L	ळ
kh	ख	th	थ	\.L	ळ
g	ग	d	द	v	व
gh	घ	dh	ध	S	श
ng	ङ	n	न	sh	ष
c	च	\.n	न	s	स
ch	छ	p	प	h	ह
j	ज	f   ph	फ	q	क्व
jh	झ	b	ब	\.kh	ख
ny	ञ	bh	भ	\.gh	ग
T	ट	m	म	\.j	ज
Th	ठ	y	य	\.D	ड
D	ड	r	र	\.Dh	ड
Dh	ढ	R	ऋ	\.f   \.ph	ऋ
N	ण	l	ल	\.y	य

図 4-10 Hindi の子音のマッピング

VOWEL	依存形	独立形
a	अ	
aa	आ	।
i	इ	ि
ee	ई	ी
u	उ	ु
oo	ऊ	ू
r\^	ऋ	ॠ
rr\^	ॠ	ॡ
l\^	ऌ	ॢ
ll\^	ॡ	ॣ
EE	ए	ै
E	ए	ै
e	ए	ै
ai	ऐ	॥
OO	ऑ	ॠ
O	ओ	ॡ
o	ओ	ॢ
au	औ	ॣ

図 4-11 Hindi の母音のマップ

OTHER			
OM	ॐ		U\~
\.C	.		A\^
M	.		A\~
H	॥		\
\.N	.		\ \
V\^	॥		EH
U\^	॥		o

図 4-12 Hindi のその他のマップ

図 4-13、図 4-14、および図 4-15 は、子音、母音、およびその他についての、英語と Kannada のマッピングです。



VOWEL	依存形	独立形
a	ಅ	
aa	ಆ	ಌ
i	ಇ	ಊ
ee	ಈ	ಋ
u	ಉ	ಋ
U oo	ಊ	ಠ
r\^	ಋ	
R\^	ೠ	ಠ
e	ಏ	ಉ
E	ಏ	ಉ
ai	ಐ	ಊ
o	ಓ	ಋ
O	ಓ	ಠ
au ou	ಔ	ಠ

図 4-14 Kannada の母音のマップ

OTHER	
M	೦
H	ಃ
OU	ೳ
LM	ಲ
RR \ ^	ರೂ
\ ~N	ನ್

図 4-15 Kannada のその他のマップ

図 4-16、図 4-17、および図 4-18 は、子音、母音、およびその他についての、英語と Malayalam のマッピングです。

CONSONANT

k	ക	th	ത	zh	ഴ
kh	ഖ	thh	ഥ	w   v	വ
g	ഗ	d	ദ	s	ശ
gh	ഘ	dh	ധ	sh	ഷ
ng	ങ	n	ന	s	സ
ch	ച	p	പ	h	ഹ
chh	ഛ	f   ph	ഫ		
j	ജ	b	ബ	SPECIAL CONSONANT	
jh	ജ്ഞ	bh	ഭ	nt	ന്ത
nj	ഞ	m	മ	nth	ന്ത
T	ട	y	യ	nnj	ഞ്ഞ
Th	ത	r	ര	nk	ങ്ക
D	ഡ	R	റ	nng	ങ
Dh	ഢ	l	ല	t	റ്റ
N	ണ	L	ള		

図 4-16 Malayalam の子音のマップ

VOWEL	依存形	独立形
a	അ	
A   aa	ആ	ഓ
i	ഇ	ീ
ee	ഇയ്യ	ീ
u	ഉ	ൂ
oo	ഉയ്യ	ൂ
r <sup>^</sup>	ഋ	ൃ
e	എ	ൈ
E	ഏ	േ
ai	ഐ	ൈ
o	ഒ	ൊ
O	ഓ	ോ
au	ഔ	ൌ
SPECIAL VOWEL		
ou	ഔ	ൗ

図 4-17 Malayalam の母音のマップ

OTHER			CHILLU	
M	o		n \ ~	ൻ
H	ഃ		N \ ~	ൺ
rr \ ^	ഋ		l \ ~	ൽ
U	ൊ		L \ ~	ൾ
UU	ോ		r \ ~	ർ

図 4-18 Malayalam のその他のマップ

図 4-19 および 図 4-20 は、子音および母音についての、英語と Tamil のマッピングです。

CONSONANT

k | g | K | G

nG | ng

ch | CH

j | J

gn | Gn

t | d

N

th | dh | TH | DH

n

n \ ^

p | b | P | B

m | M

y | Y

r

க

ங

ச

ஜ

ஞ

ட

ண

த

ந

ன

ப

ம

ய

ர

R

TR

DR

l

L

zh | ZH

w | v | W | V

S

s

h | H

ndh

nth

nj | NJ

f | ph | F | PH

ற

ற்ற

ற

ல

ள

ழ

வ

ஷ

ஸ

ஹ

ந்த

ந்த

ஞ்ச

ஃப

図 4-19 Tamil の子音のマップ

VOWEL	依存形	独立形
a	அ	
A   aa	ஆ	ஈ
i	இ	ஊ
I   ii   ee	ஈ	ஊ
u	உ	ஊ
oo   U	ஊ	ஊ
e	எ	ஊ
E   ae	ஏ	ஊ
ai	ஐ	ஊ
o	ஓ	ஊ
O   oa   oe	ஓ	ஊ
ow   ou   au	ஔ	ஊ
<b>OTHER</b>		
H	ஃ	

図 4-20 Tamil の母音のマップ

図 4-21、図 4-22、および図 4-23 は、子音、母音、およびその他についての、英語と Telugu のマッピングです。

CONSONANT					
k	క	D	డ	m	మ
K Kk kh	క	Dh	ఢ	Y	య
g	గ	N nh	ణ	r	ర
G Gh gh	ఘ	t	త	rr	ఱ
\~m	ఘ	th	థ	l	ల
ch c	చ	d	ద	L	ళ
C Ch	ఛ	dh	ఢ	w W V v	వ
j	జ	n	న	S	ష
J Jh jh	ఝ	p	ప	sh	శ
\~n	ఞ	F f P Ph ph	ఫ	s	స
T	ట	b	బ	h	హ
Th	ఠ	Bh B bh	భ		

図 4-21 Telugu の子音のマップ

VOWEL	依存形	独立形
a	అ	
A   aa	ఆ	౧
i	ఇ	ఀ
I   ia   ee   ii	ఈ	ఁ
u	ఉ	ం
ua   U   oo   uu	ఊ	ః
R	ఋ	ౠ
Ru	ౡ	ౡ
E   ae   ea	ఋ	ౢ
ai	ఌ	ౣ
o	ఋ	౤
oa   oe   O	ఌ	౥
ou   au	ఎ	౦

図 4-22 Telugu の母音のマップ

OTHER	
\ . N	౯
M	౦
H	ః
OU	౯
LM	౯
RR	ఋ
Nu	ౠ

図 4-23 Telugu のその他のマップ

## 連続表音入力方式の仕組み

各インド語スクリプトでは、「virama」(子音と結合される同等記号)が子音の半分になって文字の形を形成します。子音に対応する複数キーの組み合わせを入力すると、常に子音 + virama の形が出力され、文字が結合されることを表します。

最初の入力時の子音は半分の形ですが、後に母音が続くことによって1つの完全な音節になってさまざまに変化します。

2つの連続した子音はこれから結合される半分の形です。半分の形は、レイアウトエンジンによって1つの結合文字に変換されるか、そのまますべての言語で統語的に有効な独立形として残ります。

単語の最初に来る母音または別の母音が続く母音は独立形です。直後に子音が来る母音は依存形です。

いかなるコンテキストでも変化しない文字はその他になります。この文字は子音でも母音でもありません。

文字の一部とはならない数字およびその他の句読点は1対1でマッピングされます。

これらの原則を使用してパーサーが記述されていて、これにより入力がそれぞれのカテゴリに解析され、言語固有の Unicode のコードポイントが出力されます。連続表音入力方式エンジンはレイアウトおよび描画を処理せず、システムのほかのモジュールが行います。

---

## タイ語の地域対応

最新の Solaris 環境は、3 つのタイ語入力レベルおよび 4 つのタイ語キー配列をサポートします。

### タイ語入力方式

次のタイ語入力方式がこのリリースでサポートされています。これらの入力方式は、文字シーケンスチェックのためにタイ語 IT 標準で指定されています。

1. パススルーレベル (入力チェックなし)
2. 基本入力チェックレベル
3. 詳細入力チェックレベル

シーケンスチェックのないパススルーレベルが、前の Solaris リリースと同様にこのリリースでもデフォルトです。

F2 キーを使用して、入力レベルを切り替えることができます。

### タイ語のキー配列

タイ語の入力方式には 4 つのキー配列がサポートされています。

- **Kedmanee (TIS820-2531) キー配列。** Kedmanee 配列はコンピュータキーボード用ではなく、タイプライタ用に設計されました。キーの数に制限があるタイプライタキーボードでは、いくつかのタイ語の特殊文字が配列上にありません。コンピュータキーボードとして使用するために、TIS820-2531 は Kedmanee 配列を採用しました。



- TIS820-2538 キー配列。この拡張 Kedmanee 配列は TIS820-2531 配列の拡張版であり、元の Kedmanee 配列では使用できなかったいくつかのタイ語特殊文字が含まれています。現在、TIS820-2538 はタイ工業標準研究所が定める唯一のタイ語キー配列規格です。



- Pattajoti キー配列。Pattajoti 配列もタイプライタ用に設計されましたが、指をより使いやすいようにキーが配列されています。



- 設定可能キー配列。タイ語入力方式のためのユーザー定義キーボードです。

## タイ語入力方式の補助ウィンドウ

タイ語入力方式の補助ウィンドウでは、次の機能とユーティリティがサポートされています。

- 入力レベルの切り替え補助パレットの入力レベルボタンをクリックして、使用する入力レベル (パススルー、基本、詳細) を選択できます。
- タイ語仮想キーボードキーボードボタンをクリックして、タイ語文字の入力に使用するタイ語仮想キーボードを表示します。

## 第 5 章

---

# UTF-8 ロケールサポートの概要

---

この章では、UTF-8 ロケールサポートの概要について説明します。この章の内容は次のとおりです。

- 117 ページの「Unicode の概要」
- 118 ページの「Unicode ロケール: en\_US.UTF-8 サポート」
- 120 ページの「デスクトップ入力方式について」
- 153 ページの「システム環境」
- 158 ページの「コード変換」
- 158 ページの「DtMail サポート」
- 161 ページの「プログラミング環境」

---

## Unicode の概要

Unicode は、コンピュータ処理用のテキスト表現に使用される汎用文字コード化規格です。Unicode は、国際規格 ISO/IEC 10646-1:2000 および ISO/IEC 10646-2:2001 と完全な互換性があり、ISO/IEC 10646 とまったく同じ文字とエンコーディングポイントを含んでいます。この Unicode 規格によって、文字とその使い方に関する追加情報が提供されます。また、Unicode に準拠した実装は、ISO/IEC 10646 にも準拠します。

Unicode では、多言語のプレーンテキストのエンコーディングが一貫性をもち、国際的なテキストファイルの交換が容易になります。多言語のテキストを取り扱うコンピュータユーザー、オフィスワーカー、言語学者、研究員、科学者などにとっては、Unicode 規格によって仕事が非常に単純化されることがわかります。また、数学記号やその他の技術文字を常に使用する数学者と技術者にとっても、Unicode 規格は価値があります。

Unicode では、17 個の 16 ビットプレーンによって最大 1,114,112 のコードポイントがサポートされます。したがって、各プレーンでは、65,536 のコードポイントがサポートされます。

Unicode がサポートする 100 万以上のコードポイントのうち、バージョン 4.0 では、現在、プレーン 0、1、2、14 を使って 96,382 文字が定義されています。プレーン 15 と 16 は、ユーザー定義文字用に予約されています。プレーン 15 と 16 では、合わせて 131,068 ユーザー定義文字がサポートされます。

Unicode は、次の文字エンコーディングスキーマの 1 つを使ってエンコードされます。

- UTF-8
- UTF-16
- UTF-32

UTF-8 は、ASCII 文字コードの値を透過的に保存する可変長の Unicode エンコーディング形式です。Solaris Unicode ロケールのファイルコードには、この形式が使用されています。

UTF-16 は 16 ビットの Unicode エンコーディング形式です。UTF-16 では、最大で 65,535 文字が 1 つの 16 ビット値としてエンコードされます。65,535 から 1,114,111 の文字は 2 つの 16 ビット値としてエンコードされます (サロゲート)。

UTF-32 は、一般に 32 ビットのコンテナまたはデータタイプとして表される固定長 21 ビットの Unicode エンコーディング形式です。この形式は、Solaris Unicode ロケールの中でプロセスコード (ワイド文字コード) として使用されます。

Unicode 規格や ISO/IEC 10646、およびその表現形式については、次の資料を参照してください。

- Unicode Consortium が提供する *The Unicode Standard, Version 4.0*
- ISO/IEC 10646-1:2000, Information Technology-Universal Multiple-Octet Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane
- ISO/IEC 10646-2: Information Technology-Universal Multiple-Octet Character Set (UCS) - Part 2: Secondary Multilingual Plane for Scripts and Symbols, Supplementary Plane for CJK Ideographs, Special Purpose Plane
- Unicode Consortium の Web サイト <http://www.unicode.org/>

## Unicode ロケール: en\_US.UTF-8 サポート

Unicode/UTF-8 ロケールは Unicode 4.0 をサポートします。en\_US.UTF-8 ロケールは、UTF-8 をコードセットとして使用することによってマルチスクリプト処理をサポートします。このロケールは、複数のスクリプトでテキストを入出力できます。これは、Solaris OS で最初にこの機能を持ったロケールです。他の UTF-8 ロケールの機能も en\_us.UTF-8 の機能と似ています。en\_US.UTF-8 に関する以下の説明はこれらのロケールにも当てはまります。

---

注 - UTF-8 は、1992 年の X/Open-Uniform Joint Internationalization Working Group (XoJIG) によって公式化され、1996 年に ISO と IEC によって ISO/IEC 10646-1:1993 の修正第 2 条として承認された Unicode/ISO/IEC 10646-1 のファイルシステム安全汎用文字セット変換形式です。この規格は、Unicode Consortium、ISO (国際標準化機構)、IEC (国際電気標準会議) によって、Unicode 4.0 および ISO/IEC 10646-1 として採用されました。

---

Solaris 環境の Unicode ロケールは、Unicode 4.0 や ISO/IEC 10646-1 および 10646-2 に定義されているすべてのコードポイント値の処理をサポートしています。サポートされるスクリプトには、全ヨーロッパやアジアのスクリプトだけでなく、アラビア語、ヘブライ語、ヒンディー語、タイ語などの複雑なテキストレイアウトのスクリプトも含まれます。

---

注 - 一部の Unicode ロケール、特にアジアロケールは、追加の漢字および Hanzi グリフを含みます。

---

使用できるフォントに制限があるため、最新の Solaris には次の文字セットの文字グリフのみ含まれています。

- ISO 8859-1 (英語、フランス語、スペイン語、ドイツ語など、ほとんどの西ヨーロッパ諸国の言語)
- ISO 8859-2 (チェコ語、ポーランド語、ハンガリー語など、ほとんどの中央ヨーロッパ諸国の言語)
- ISO 8859-4 (スカンジナビアおよびバルト諸国の言語)
- ISO 8859-5 (ロシア語)
- ISO 8859-6 (さらに多くの表示形式の文字グリフを含むアラビア語)
- ISO 8859-7 (ギリシャ語)
- ISO 8859-8 (ヘブライ語)
- ISO 8859-9 (トルコ語)
- TIS 620.2533 (さらに多くの表示形式の文字グリフを含むタイ語)
- ISO 8859-15 (ユーロ記号を含む多くの西ヨーロッパ言語)
- GB 2312-1980 (簡体字中国語)
- JIS X 0201-1976、JIS X 0208-1990 (日本語)
- KSC 5601-1992 Annex 3 (韓国語)
- GB 18030 (簡体字中国語)
- HKSCS (繁体字中国語、香港)
- Big5 (繁体字中国語、台湾)
- IS 13194.1991 (ISCII ともいう) (ヒンディー語、さらに多くの表示形式の文字グリフを含む)

対応するグリフが en\_US.UTF-8 ロケールに含まれていない文字をユーザーが表示させると、ロケールが、代わりに no-glyph というグリフを以下の例のように表示します。



インストール時にこのロケールをシステムのデフォルトロケールとして選択することができます。

同じレベルの en\_US.UTF-8 ロケールのサポートが、64 ビットと 32 ビットの Solaris システムの両方に用意されています。

---

注 – Motif および Solaris CDE のデスクトップアプリケーションとライブラリは、en\_US.UTF-8 ロケールをサポートしています。ただし、XView™ と OLIT ライブラリでは en\_US.UTF-8 ロケールをサポートしていません。

---

## デスクトップ入力方式について

CDE では、Xm ツールキットを使用する国際化アプリケーションで、各ロケールの文字の入力を行う機能を提供しています。XmText [Field] ウィジェットは各ロケールの入力方式のインタフェースとなります。いくつかの言語環境において、テキストは右から左、上から下などの方向に書かれるため、入力方式が国際化されています。同じアプリケーションで、複数のフォントを使用する異なる入力方式の使用が可能です。

プリエディット領域は確定前の文字列が表示されます。テキストの作成は、次の 4 つのモードで行うことができます。

- OffTheSpot
- OverTheSpot (デフォルト)
- Root
- なし

OffTheSpot モードでは、場所はメインウィンドウの下のステータス領域の右側です。OverTheSpot モードでは、プリエディット領域はカーソル位置です。Root モードではプリエディットおよびステータス領域はクライアントのウィンドウと分けられます。

詳細については、VendorShell (3X) マニュアルページの XmNpreeditType リソースの項を参照してください。

---

注 - 最新の Solaris 環境には、簡体/繁体字中国語、日本語、韓国語用の、アジア固有の入力方式があります。これらの方式は、Unicode ロケールに対する現在のマルチスクリプト入力方式の他に提供されるものです。

---

121 ページの「入力方式の選択」では、選択された入力方式、それらの使用方法、それらの切り替え替方法について説明します。

## スクリプトの選択と入力方式

Solaris Unicode ロケールは複数のスクリプトをサポートしています。それぞれの Unicode ロケールには、次の 14 の入力方式が含まれています。

- 英語/ヨーロッパ言語
- キリル文字
- ギリシャ語
- アラビア語
- ヘブライ語
- タイ語
- 日本語
- 韓国語
- 簡体字中国語
- 繁体字中国語
- 繁体字中国語 (香港)
- インド語
- Unicode の 16 進法と 8 進法の入力方式
- 表検索入力モード

---

## 入力方式の選択

特定の入力モードに切り替えるには、Compose キーの組み合わせを使用するか、入力モード選択ウィンドウを使用します。入力モード選択ウィンドウを表示するには、アプリケーションウィンドウ左下隅のステータス領域でクリックします。次の図のような入力モード選択ウィンドウが表示されます。

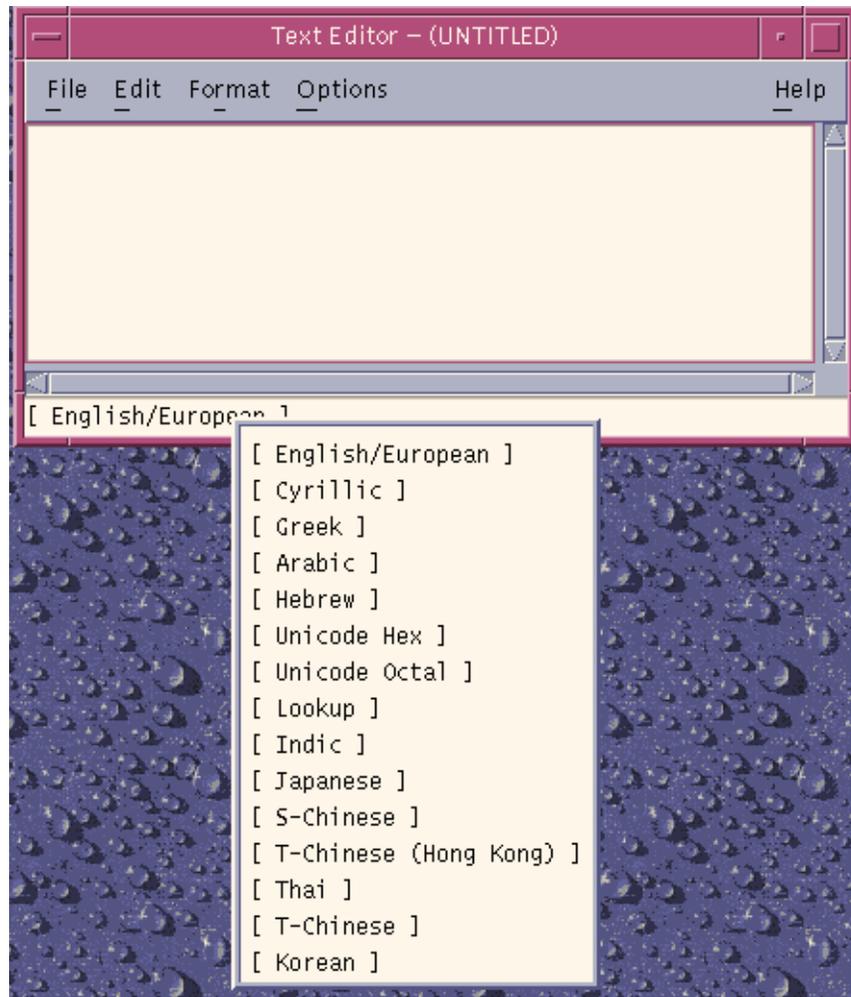


図 5-1 入力モード選択ウィンドウ

## 入力モードの切り替えキーシーケンス

現在の入力モードを新しい入力モードに切り替えるには、表 5-1 に示すキーシーケンスを使用します。ただし、アジアの入力方式を使用している場合は、Control-Space キーを押して英語/ヨーロッパ入力モードに戻る必要があります。英語/ヨーロッパ言語入力モードに戻った後は、該当するキーシーケンスを使って任意の入力モードに自由に切り替えることができます。

たとえば、英語/ヨーロッパ言語入力モードをキリル語入力モードに切り替える場合は、次のキーシーケンスを使用します。

1. Compose キーを押します。
2. c キーを押して離します。
3. c キーを押します。

表 5-1 入力モードの切り替えキーシーケンス

キーシーケンス	入力モード
Control-Space	英語/ヨーロッパ言語
Compose c c	キリル文字
Compose g g	ギリシャ語
Compose a r	アラビア語
Compose h h	ヘブライ語
Compose t t	タイ語
Compose h i	インド語
Compose i n	インド語
Compose j a	日本語
Compose k o	韓国語
Compose s c	簡体字中国語
Compose t c	繁体字中国語
Compose h k	繁体字中国語 (香港)
Compose u o	Unicode 8 進数コード入力モード
Compose u h	Unicode 16 進数コード入力モード
Compose l l	表検索入力モード

## 英語/ヨーロッパ言語入力モード

英語/ヨーロッパ言語入力モードでは、英語のアルファベットだけでなく、ヨーロッパの言語で使用される発音符号(á, è, î, ð, ü など)や特殊文字(j, §, ¿)なども使用できます。

この入力モードはすべてのアプリケーションのデフォルトのモードです。入力モードは、GUI アプリケーションウィンドウの左下隅に表示されます。

Latin-1、Latin-2、Latin-4、Latin-5、Latin-9 の発音符号付きの文字や特殊文字を入力するには、次の例のように Compose シーケンスを入力する必要があります。

Ä を表示するには、次のようにします。

1. Compose キーを押してから離します。
2. Shift-A キーを押します。Shift-A を離します。
3. " キーを押してから離します。

¿ を表示するには、次のようにします。

1. Compose キーを押してから離します。
2. ? キーを押してから離します。
3. ? キーを押してから離します。

キーボードに Compose キーがない場合は、Control-Shift キーを押すことによって Compose キーをエミュレートできます。

ユーロの通貨記号 (Unicode の U+20AC) を入力するには、次のいずれかのシーケンスを使用します。

- AltGraph と E (同時に)
- AltGraph と 4 (同時に)
- AltGraph と 5 (同時に)

上の入力シーケンスでは、両方のキーを同時に押す必要があります。キーボードに AltGraph キーがない場合は、代わりに Compose e = あるいは Compose c = などのユーロ記号の入力シーケンスが使用できます。

次の表に、Solaris OS で Latin-1、Latin-2、Latin-3、Latin-4、Latin-5、Latin-9 の文字の入力によく使用される Compose シーケンスを示します。

次の表に、Latin-1 の一般的な Compose シーケンスを示します。

表 5-2 Latin-1 の一般的な Compose キーシーケンス

Compose の次に押すキー	その次に押すキー	結果
Space	Space	スペース
s	1	上付きの 1
s	2	上付きの 2
s	3	上付きの 3
!	!	逆感嘆符
x	o	通貨記号 (¤)
p	!	パラグラフ記号 (¶)
/	u	ミュー (µ)

表 5-2 Latin-1 の一般的な Compose キーシーケンス (続き)

Compose の次に押すキー	その次に押すキー	結果
'	"	アキュートアクセント記号 (´)
,	, (コンマ)	セディーユ (ç)
	"	ウムラウト (¨)
-	^	長音記号 (˘)
o	o	度 (°)
x	x	乗算記号 (×)
+	-	プラスマイナス (±)
-	-	ソフトハイフン (–)
-	:	除算記号 (÷)
-	a	女性序数識別 (ª)
-	o	男性序数識別 (º)
-	, (コンマ)	否定記号 (¬)
.	.	中点 (·)
1	2	分数 (½)
1	4	分数 (¼)
3	4	分数 (¾)
<	<	左二重引用符 («)
>	>	右二重引用符 (»)
?	?	逆疑問符 (¿)
A	' (逆引用符)	グレーブアクセント付き A (À)
A	' (単一引用符)	アキュートアクセント付き A (Á)
A	*	オングストローム付き A (Å)
A		ウムラウト付き A (Ä)
A	^	サーカムフレクトアクセント付き A (Â)
A	~	チルド付き A (Ã)
A	E	AE リゲチャ (Æ)
C	, (コンマ)	セディーユ付き C (Ç)
C	o	著作権表示記号 (©)

表 5-2 Latin-1 の一般的な Compose キーシーケンス (続き)

Compose の次に押すキー	その次に押すキー	結果
D	-	キャピタルエズ (ð)
E	' (逆引用符)	グレーブアクセント付き E (È)
E	'	アキュートアクセント付き E (É)
E		ウムラウト付き E (Ë)
E	^	サーカムフレクトアクセント付き E (Ê)
I	' (逆引用符)	グレーブアクセント付き I (Ì)
I	'	アキュートアクセント付き I (Í)
I	"	ウムラウト付き I (Ï)
I	^	サーカムフレクトアクセント付き I (Î)
L	-	ポンド記号 (£)
N	~	チルド付き N (Ñ)
O	' (逆引用符)	グレーブアクセント付き O (Ò)
O	'	アキュートアクセント付き O (Ó)
O	/	斜線付き O (Ø)
O	"	ウムラウト付き O (Ö)
O	^	サーカムフレクトアクセント付き O (Ô)
O	~	チルド付き O (Õ)
R	o	登録商標 (®)
T	h	アイスランド語ソーン (þ)
U	' (逆引用符)	グレーブアクセント付き U (Ù)
U	'	アキュートアクセント付き U (Ú)
U		ウムラウト付き U (Û)
U	^	サーカムフレクトアクセント付き U (Û)
Y	'	アキュートアクセント付き Y (ý)
Y	-	円記号 (¥)
a	' (逆引用符)	グレーブアクセント付き a (à)
a	'	アキュートアクセント付き a (á)

表 5-2 Latin-1 の一般的な Compose キーシーケンス (続き)

Compose の次に押すキー	その次に押すキー	結果
a	*	オングストローム付き a (å)
a		ウムラウト付き a (ä)
a	~	チルド付き a (ā)
a	^	サーカムフレクトアクセント付き a (â)
a	e	ae リゲチャ (æ)
c	, (コンマ)	セディーユ付き c (ç)
c	/	セント記号 (¢)
c	o	著作権表示記号 (©)
d	-	エズ (ð)
e	' (逆引用符)	グレーブアクセント付き e (è)
e	'	アキュートアクセント付き e (é)
e		ウムラウト付き e (ë)
e	^	サーカムフレクトアクセント付き e (ê)
i	' (逆引用符)	グレーブアクセント付き i (ì)
i	'	アキュートアクセント付き i (í)
i		ウムラウト付き i (ï)
i	^	サーカムフレクトアクセント付き i (î)
n	~	チルド付き n (ñ)
o	' (逆引用符)	グレーブアクセント付き o (ò)
o	'	アキュートアクセント付き o (ó)
o	/	斜線付き o (ø)
o		ウムラウト付き o (ö)
o	^	サーカムフレクトアクセント付き o (ô)
o	~	チルド付き o (õ)
s	s	ドイツ語の二重 s (シャープ S) (ß)
t	h	ソーン (þ)
u	' (逆引用符)	グレーブアクセント付き u (ù)
u	'	アキュートアクセント付き u (ú)
u		ウムラウト付き u (ü)

表 5-2 Latin-1 の一般的な Compose キーシーケンス (続き)

Compose の次に押すキー	その次に押すキー	結果
u	^	サーカムフレクトアクセント付き u (û)
y	'	アキュートアクセント付き y (y)
y		ウムラウト付き y (ÿ)
l	l	破断線 (¡)

次の表に、Latin-2 の一般的な Compose キーシーケンスを示します。

表 5-3 Latin-2 の一般的な Compose キーシーケンス

Compose の次に押すキー	その次に押すキー	結果
k	k	kra
A	—	長音記号付き A
E	—	長音記号付き E
E	.	ドット付き E
G	,	セディーユ付き G
I	—	長音記号付き I
I	~	チルド付き I
I	a	ogonek 付き I
K	,	セディーユ付き K
L	,	セディーユ付き L
N	,	セディーユ付き N
O	—	長音記号付き O
R	,	セディーユ付き R
T		ストローク付き T
U	~	チルド付き U
U	a	ogonek 付き U
U	—	長音記号付き U
N	N	エング
a	—	長音記号付き a
e	—	長音記号付き e

表 5-3 Latin-2 の一般的な Compose キーシーケンス (続き)

Compose の次に押すキー	その次に押すキー	結果
e	.	ドット付き e
g	,	セディーユ付き g
i	_	長音記号付き i
i	~	チルド付き i
i	a	ogonek 付き i
k	,	セディーユ付き k
l	,	セディーユ付き l
n	,	セディーユ付き n
o	_	長音記号付き o
r	,	セディーユ付き r
t		ストローク付き t
u	~	チルド付き u
u	a	ogonek 付き u
u	_	長音記号付き u
n	n	エンゲ

次の表に、Latin-3 の一般的な Compose キーシーケンスを示します。

表 5-4 Latin-3 の一般的な Compose キーシーケンス

Compose の次に押すキー	その次に押すキー	結果
C	>	サーカムフレクトアクセント付き C
C	.	ドット付き C
G	>	サーカムフレクトアクセント付き G
G	.	ドット付き G
H	>	サーカムフレクトアクセント付き H
J	>	サーカムフレクトアクセント付き j
S	>	サーカムフレクトアクセント付き S
U	u	短音記号付き U

表 5-4 Latin-3 の一般的な Compose キーシーケンス (続き)

Compose の次に押すキー	その次に押すキー	結果
c	>	サーカムフレクトアクセント付き c
c	.	ドット付き c
g	>	サーカムフレクトアクセント付き g
g	.	ドット付き g
h	>	サーカムフレクトアクセント付き h
j	>	サーカムフレクトアクセント付き j
s	>	サーカムフレクトアクセント付き s
u	u	短音記号付き u

次の表に、Latin-4 の一般的な Compose キーシーケンスを示します。

表 5-5 Latin-4 の一般的な Compose キーシーケンス

Compose の次に押すキー	その次に押すキー	結果
k	k	kra
A	_	長音記号付き A
E	_	長音記号付き E
E	.	ドット付き E
G	,	セディーユ付き G
I	_	長音記号付き I
I	~	チルド付き I
I	a	ogonek 付き I
K	,	セディーユ付き K
L	,	セディーユ付き L
N	,	セディーユ付き N
O	_	長音記号付き O
R	,	セディーユ付き R
T	l	ストローク付き T
U	~	チルド付き U
U	a	ogonek 付き U

表 5-5 Latin-4 の一般的な Compose キーシーケンス (続き)

Compose の次に押すキー	その次に押すキー	結果
U	-	長音記号付き U
N	N	エンゲ
a	-	長音記号付き a
e	-	長音記号付き e
e	.	ドット付き e
g	,	セディーユ付き g
i	-	長音記号付き i
i	~	チルド付き i
i	a	ogonek 付き i
k	,	セディーユ付き k
l	,	セディーユ付き l
n	,	セディーユ付き n
o	-	長音記号付き o
r	,	セディーユ付き r
t	l	ストローク付き t
u	~	チルド付き u
u	a	ogonek 付き u
u	-	長音記号付き u
n	n	エンゲ

次の表に、Latin-5 の一般的な Compose キーシーケンスを示します。

表 5-6 Latin-5 の一般的な Compose キーシーケンス

Compose の次に押すキー	その次に押すキー	結果
G	u	短音記号付き G
I	.	ドット付き I
g	u	短音記号付き g
i	.	ドット付き i

次の表に、Latin-9 の一般的な Compose キーシーケンスを示します。

表 5-7 Latin-9 の一般的な Compose キーシーケンス

Compose の次に押すキー	その次に押すキー	結果
o	e	oe の合字
O	E	OE の合字
Y	“	ウムラウト付き Y

アクセントデッドキーがあるキーボードを使用しているときは、以下の Compose キーシーケンスを使用します。dead\_acute などのキー名は、  
/usr/openwin/include/X11/keysymdef.h に見られる、X\_dead\_acute などの X11 登録 keysym 名に由来します。SunFA\_Circum などのキー名は、  
/usr/openwin/include/X11/Sunkeysym.h に見られる、SunXK\_FA\_Circum などの Sun により定義された X11 keysym 名に由来します。

表 5-8 アクセントデッドキーに基づく Compose キーシーケンス

その次に押すキー	その次に押すキー	結果
dead_grave	Space	グレーブアクセント記号
dead_acute	アポストロフィ	アキュートアクセント記号
dead_acute	Space	アポストロフィ
dead_diaeresis	二重引用符	ウムラウト
dead_diaeresis	Space	ウムラウト
dead_circumflex	Space	サーカムフレクトアクセント記号
dead_circumflex	スラッシュ	垂直線
dead_circumflex	0	度記号
dead_circumflex	1	上付きの 1
dead_circumflex	2	上付きの 2
dead_circumflex	3	上付きの 3
dead_circumflex	ピリオド	中点
dead_circumflex	感嘆符	破断線
dead_circumflex	マイナス	長音記号
dead_circumflex	アンダースコア	長音記号
dead_cedilla	コンマ	セディーユ
dead_cedilla	マイナス	否定記号

表 5-8 アクセントデッドキーに基づく Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
dead_tilde	Space	チルド
dead_grave	A	グレーブアクセント付き A
dead_acute	A	アキュートアクセント付き A
dead_circumflex	A	サーカムフレクトアクセント付き A
dead_tilde	A	チルド付き A
dead_diaeresis	A	ウムラウト付き A
dead_grave	a	グレーブアクセント付き a
dead_acute	a	アキュートアクセント付き a
dead_circumflex	a	サーカムフレクトアクセント付き a
dead_tilde	a	チルド付き a
dead_diaeresis	a	ウムラウト付き a
dead_cedilla	C	セディーユ付き C
dead_cedilla	c	セディーユ付き c
dead_grave	E	グレーブアクセント付き E
dead_acute	E	アキュートアクセント付き E
dead_circumflex	E	サーカムフレクトアクセント付き E
dead_diaeresis	E	ウムラウト付き E
dead_grave	e	グレーブアクセント付き e
dead_acute	e	アキュートアクセント付き e
dead_circumflex	e	サーカムフレクトアクセント付き e
dead_diaeresis	e	ウムラウト付き e
dead_grave	I	グレーブアクセント付き I
dead_acute	I	アキュートアクセント付き I
dead_circumflex	I	サーカムフレクトアクセント付き I
dead_diaeresis	I	ウムラウト付き I
dead_grave	i	グレーブアクセント付き i

表 5-8 アクセントデッドキーに基づく Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
dead_acute	i	アキュートアクセント付き i
dead_circumflex	i	サーカムフレクトアクセント付き i
dead_diaeresis	i	ウムラウト付き i
dead_tilde	N	チルド付き N
dead_tilde	n	チルド付き n
dead_grave	O	グレーブアクセント付き O
dead_acute	O	アキュートアクセント付き O
dead_circumflex	O	サーカムフレクトアクセント付き O
dead_tilde	O	チルド付き O
dead_diaeresis	O	ウムラウト付き O
dead_grave	o	グレーブアクセント付き o
dead_acute	o	アキュートアクセント付き o
dead_circumflex	o	サーカムフレクトアクセント付き o
dead_tilde	o	チルド付き o
dead_diaeresis	o	ウムラウト付き o
dead_cedilla	S	セディーユ付き S
dead_cedilla	s	セディーユ付き s
dead_grave	U	グレーブアクセント付き U
dead_acute	U	アキュートアクセント付き U
dead_circumflex	U	サーカムフレクトアクセント付き U
dead_diaeresis	U	ウムラウト付き U
dead_grave	u	グレーブアクセント付き u
dead_acute	u	アキュートアクセント付き u
dead_circumflex	u	サーカムフレクトアクセント付き u
dead_diaeresis	u	ウムラウト付き u
dead_acute	Y	アキュートアクセント付き Y

表 5-8 アクセントデッドキーに基づく Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
dead_acute	y	アキュートアクセント付き y
dead_diaeresis	y	ウムラウト付き y
SunFA_Grave	Space	グレーブアクセント
SunFA_Grave	A	グレーブアクセント付き A
SunFA_Grave	a	グレーブアクセント付き a
SunFA_Grave	E	グレーブアクセント付き E
SunFA_Grave	e	グレーブアクセント付き e
SunFA_Grave	I	グレーブアクセント付き I
SunFA_Grave	i	グレーブアクセント付き i
SunFA_Grave	O	グレーブアクセント付き O
SunFA_Grave	o	グレーブアクセント付き o
SunFA_Grave	U	グレーブアクセント付き U
SunFA_Grave	u	グレーブアクセント付き u
SunFA_Acute	アポストロフィ	アキュートアクセント記号
SunFA_Acute	Space	アポストロフィ
SunFA_Acute	A	アキュートアクセント付き A
SunFA_Acute	a	アキュートアクセント付き a
SunFA_Acute	C	アキュートアクセント付き C
SunFA_Acute	c	アキュートアクセント付き c
SunFA_Acute	E	アキュートアクセント付き E
SunFA_Acute	e	アキュートアクセント付き e
SunFA_Acute	I	アキュートアクセント付き I
SunFA_Acute	i	アキュートアクセント付き i
SunFA_Acute	L	アキュートアクセント付き L
SunFA_Acute	l	アキュートアクセント付き l
SunFA_Acute	N	アキュートアクセント付き N
SunFA_Acute	n	アキュートアクセント付き n
SunFA_Acute	O	アキュートアクセント付き O
SunFA_Acute	o	アキュートアクセント付き o

表 5-8 アクセントデッドキーに基づく Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
SunFA_Acute	R	アキュートアクセント付き R
SunFA_Acute	r	アキュートアクセント付き r
SunFA_Acute	S	アキュートアクセント付き S
SunFA_Acute	s	アキュートアクセント付き s
SunFA_Acute	U	アキュートアクセント付き U
SunFA_Acute	u	アキュートアクセント付き u
SunFA_Acute	Y	アキュートアクセント付き Y
SunFA_Acute	y	アキュートアクセント付き y
SunFA_Acute	Z	アキュートアクセント付き Z
SunFA_Acute	z	アキュートアクセント付き z
SunFA_Cedilla	コンマ	セディーユ
SunFA_Cedilla	マイナス	否定
SunFA_Cedilla	C	セディーユ付き C
SunFA_Cedilla	c	セディーユ付き c
SunFA_Cedilla	G	セディーユ付き G
SunFA_Cedilla	g	セディーユ付き g
SunFA_Cedilla	K	セディーユ付き K
SunFA_Cedilla	k	セディーユ付き k
SunFA_Cedilla	L	セディーユ付き L
SunFA_Cedilla	l	セディーユ付き l
SunFA_Cedilla	N	セディーユ付き N
SunFA_Cedilla	n	セディーユ付き n
SunFA_Cedilla	R	セディーユ付き R
SunFA_Cedilla	r	セディーユ付き r
SunFA_Cedilla	S	セディーユ付き S
SunFA_Cedilla	s	セディーユ付き s
SunFA_Cedilla	T	セディーユ付き T
SunFA_Cedilla	t	セディーユ付き t

表 5-8 アクセントデッドキーに基づく Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
SunFA_Circum	Space	サーカムフレクトアクセント記号
SunFA_Circum	0	度記号
SunFA_Circum	1	上付きの 1
SunFA_Circum	2	上付きの 2
SunFA_Circum	3	上付きの 3
SunFA_Circum	感嘆符	破断線
SunFA_Circum	マイナス	長音記号
SunFA_Circum	アンダースコア	長音記号
SunFA_Circum	ピリオド	中点
SunFA_Circum	スラッシュ	垂直線
SunFA_Circum	A	サーカムフレクトアクセント付き A
SunFA_Circum	a	サーカムフレクトアクセント付き a
SunFA_Circum	C	サーカムフレクトアクセント付き C
SunFA_Circum	c	サーカムフレクトアクセント付き c
SunFA_Circum	E	サーカムフレクトアクセント付き E
SunFA_Circum	e	サーカムフレクトアクセント付き e
SunFA_Circum	G	サーカムフレクトアクセント付き G
SunFA_Circum	g	サーカムフレクトアクセント付き g
SunFA_Circum	H	サーカムフレクトアクセント付き H
SunFA_Circum	h	サーカムフレクトアクセント付き h
SunFA_Circum	I	サーカムフレクトアクセント付き I

表 5-8 アクセントデッドキーに基づく Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
SunFA_Circum	i	サーカムフレクトアクセント付き i
SunFA_Circum	J	サーカムフレクトアクセント付き J
SunFA_Circum	j	サーカムフレクトアクセント付き j
SunFA_Circum	O	サーカムフレクトアクセント付き O
SunFA_Circum	o	サーカムフレクトアクセント付き o
SunFA_Circum	S	サーカムフレクトアクセント付き S
SunFA_Circum	s	サーカムフレクトアクセント付き s
SunFA_Circum	U	サーカムフレクトアクセント付き U
SunFA_Circum	u	サーカムフレクトアクセント付き u
SunFA_Diaeresis	二重引用符	ウムラウト
SunFA_Diaeresis	Space	ウムラウト
SunFA_Diaeresis	A	ウムラウト付き A
SunFA_Diaeresis	a	ウムラウト付き a
SunFA_Diaeresis	E	ウムラウト付き E
SunFA_Diaeresis	e	ウムラウト付き e
SunFA_Diaeresis	I	ウムラウト付き I
SunFA_Diaeresis	i	ウムラウト付き i
SunFA_Diaeresis	O	ウムラウト付き O
SunFA_Diaeresis	o	ウムラウト付き o
SunFA_Diaeresis	U	ウムラウト付き U
SunFA_Diaeresis	u	ウムラウト付き u
SunFA_Diaeresis	y	ウムラウト付き y
SunFA_Diaeresis	Y	ウムラウト付き Y



次の図にキリル文字 (ロシア語) キーボードの配列を示します。

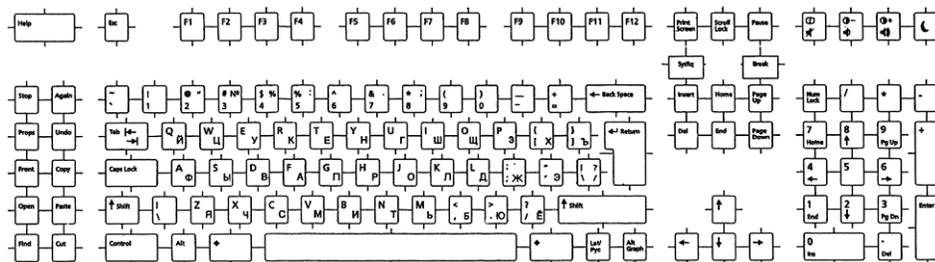


図 5-3 キリル文字 (ロシア語) キーボード

キリル文字入力モードでは、英語やヨーロッパ言語のテキストを入力することはできません。英語/ヨーロッパ言語入力モードに戻るには、Control-Space キーを押すか、入力モード選択ウィンドウのステータス領域をクリックし、「英語/ヨーロッパ言語 (English/European)」入力モードを選択します。詳細については121 ページの「入力方式の選択」を参照してください。

また、対応する入力モード切り替えキーシーケンスを入力することで、他の入力モードに切り替えることもできます。

## ギリシャ語入力モード

ギリシャ語入力モードを選択するには、Compose g g を押すか、入力モード選択ウィンドウで「ギリシャ語 (Greek)」を選択します。入力モード選択ウィンドウの表示方法については、121 ページの「入力方式の選択」を参照してください。

ギリシャ語入力モードでは、英語やヨーロッパ言語のテキストを入力することはできません。英語/ヨーロッパ言語入力モードに戻るには、Control-Space キーを押すか、入力モード選択ウィンドウのステータス領域をクリックし、「英語/ヨーロッパ言語 (English/European)」入力モードを選択します。次の図にギリシャ語キーボード (ヨーロッパキーボード) の配列を示します。

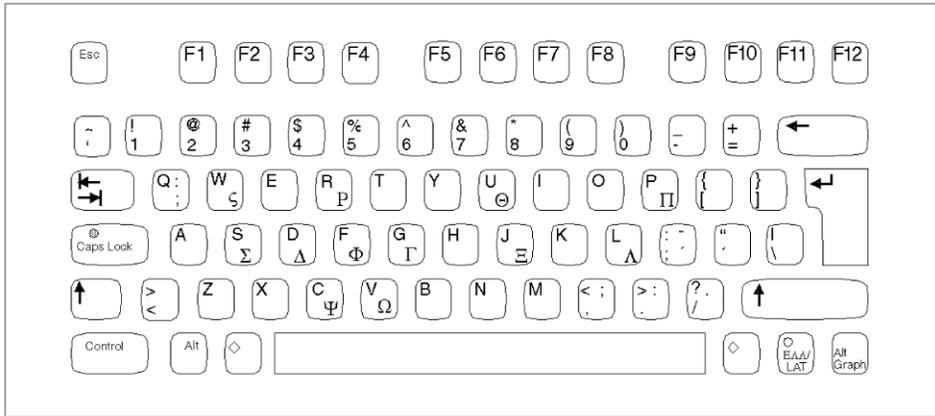


図 5-4 ギリシャ語キーボード (ヨーロッパキーボード)

次の図にギリシャ語キーボード (UNIX キーボード) を示します。

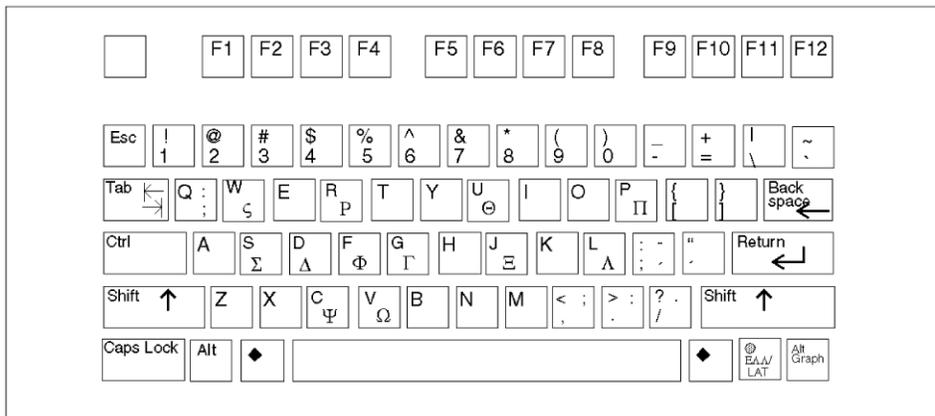


図 5-5 ギリシャ語キーボード (UNIX キーボード)

以下の Compose キーシーケンスがギリシャ語入力モードでサポートされています。Compose キーシーケンスのいくつかは、アクセントデッドキーで始まります。「ordfemine」は女性序数識別 (feminine ordinal indicator) キーの略語です。

表 5-9 ギリシャ語入力モードの Compose キーシーケンス

その次に押すキー	その次に押すキー	結果
セミコロン	a	tonos 付きギリシャ語アルファ (小文字)

表 5-9 ギリシャ語入力モードの Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
セミコロンの	e	tonos 付きギリシャ語エプシロン (小文字)
セミコロンの	h	tonos 付きギリシャ語エータ (小文字)
セミコロンの	i	tonos 付きギリシャ語イオータ (小文字)
セミコロンの	o	tonos 付きギリシャ語オミクロン (小文字)
セミコロンの	y	tonos 付きギリシャ語ユプシロン (小文字)
セミコロンの	v	tonos 付きギリシャ語オメガ (小文字)
セミコロンの	A	tonos 付きギリシャ語アルファ (大文字)
セミコロンの	E	tonos 付きギリシャ語エプシロン (大文字)
セミコロンの	H	tonos 付きギリシャ語エータ (大文字)
セミコロンの	I	tonos 付きギリシャ語イオータ (大文字)
セミコロンの	O	tonos 付きギリシャ語オミクロン (大文字)
セミコロンの	Y	tonos 付きギリシャ語ユプシロン (大文字)
セミコロンの	V	tonos 付きギリシャ語オメガ (大文字)
dead_acute	ギリシャ語アルファ (小文字)	tonos 付きギリシャ語アルファ (小文字)
dead_acute	ギリシャ語エプシロン (小文字)	tonos 付きギリシャ語エプシロン (小文字)
dead_acute	ギリシャ語エータ (小文字)	tonos 付きギリシャ語エータ (小文字)
dead_acute	ギリシャ語イオータ (小文字)	tonos 付きギリシャ語イオータ (小文字)
dead_acute	ギリシャ語オミクロン (小文字)	tonos 付きギリシャ語オミクロン (小文字)

表 5-9 ギリシャ語入力モードの Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
dead_acute	ギリシャ語ユプシロン (小文字)	tonos 付きギリシャ語ユプシロン (小文字)
dead_acute	ギリシャ語オメガ (小文字)	tonos 付きギリシャ語オメガ (小文字)
dead_acute	ギリシャ語アルファ (大文字)	tonos 付きギリシャ語アルファ (大文字)
dead_acute	ギリシャ語エプシロン (大文字)	tonos 付きギリシャ語エプシロン (大文字)
dead_acute	ギリシャ語イータ (大文字)	tonos 付きギリシャ語イータ (大文字)
dead_acute	ギリシャ語イオータ (大文字)	tonos 付きギリシャ語イオータ (大文字)
dead_acute	ギリシャ語オミクロン (大文字)	tonos 付きギリシャ語オミクロン (大文字)
dead_acute	ギリシャ語ユプシロン (大文字)	tonos 付きギリシャ語ユプシロン (大文字)
dead_acute	ギリシャ語オメガ (大文字)	tonos 付きギリシャ語オメガ (大文字)
dead_acute	a	tonos 付きギリシャ語アルファ (小文字)
dead_acute	e	tonos 付きギリシャ語エプシロン (小文字)
dead_acute	h	tonos 付きギリシャ語エータ (小文字)
dead_acute	i	tonos 付きギリシャ語イオータ (小文字)
dead_acute	o	tonos 付きギリシャ語オミクロン (小文字)
dead_acute	y	tonos 付きギリシャ語ユプシロン (小文字)
dead_acute	v	tonos 付きギリシャ語オメガ (小文字)
dead_acute	A	tonos 付きギリシャ語アルファ (大文字)
dead_acute	E	tonos 付きギリシャ語エプシロン (大文字)

表 5-9 ギリシャ語入力モードの Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
dead_acute	H	tonos 付きギリシャ語エータ (大文字)
dead_acute	I	tonos 付きギリシャ語イオータ (大文字)
dead_acute	O	tonos 付きギリシャ語オミクロン (大文字)
dead_acute	Y	tonos 付きギリシャ語ユプシロン (大文字)
dead_acute	V	tonos 付きギリシャ語オメガ (大文字)
コロン	i	dialytika 付きギリシャ語イオータ (小文字)
コロン	y	dialytika 付きギリシャ語ユプシロン (小文字)
コロン	I	dialytika 付きギリシャ語イオータ (大文字)
コロン	Y	dialytika 付きギリシャ語ユプシロン (大文字)
dead_diaeresis	i	dialytika 付きギリシャ語イオータ (小文字)
dead_diaeresis	y	dialytika 付きギリシャ語ユプシロン (小文字)
dead_diaeresis	I	dialytika 付きギリシャ語イオータ (大文字)
dead_diaeresis	Y	dialytika 付きギリシャ語ユプシロン (大文字)
dead_diaeresis	ギリシャ語イオータ (小文字)	dialytika 付きギリシャ語イオータ (小文字)
dead_diaeresis	ギリシャ語ユプシロン (小文字)	dialytika 付きギリシャ語ユプシロン (小文字)
dead_diaeresis	ギリシャ語イオータ (大文字)	dialytika 付きギリシャ語イオータ (大文字)
dead_diaeresis	ギリシャ語ユプシロン (大文字)	dialytika 付きギリシャ語ユプシロン (大文字)
セミコロン	セミコロン	ギリシャ語 tonos
コロン	コロン	ウムラウト/dialytika

表 5-9 ギリシャ語入力モードの Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
ordfeminine	0	プラス
ordfeminine	1	セクション記号
ordfeminine	2	上付きの 2
ordfeminine	3	上付きの 3
ordfeminine	5	破断線
ordfeminine	6	著作権表示記号
ordfeminine	7	否定記号
ordfeminine	8	ソフトハイフン
ordfeminine	9	度記号
ordfeminine	ハイフン	分数 2 分の 1
ordfeminine	バックスラッシュ	ポンド記号
ordfeminine	左中括弧	修飾文字の逆コンマ
ordfeminine	右中括弧	修飾文字のアポストロフィ
ordfeminine	左角括弧	左向き二重引用符
ordfeminine	右角括弧	右向き二重引用符
SunFA_Acute	a	tonos 付きギリシャ語アルファ (小文字)
SunFA_Acute	e	tonos 付きギリシャ語エプシロン (小文字)
SunFA_Acute	h	tonos 付きギリシャ語イータ (小文字)
SunFA_Acute	i	tonos 付きギリシャ語イオータ (小文字)
SunFA_Acute	o	tonos 付きギリシャ語オミクロン (小文字)
SunFA_Acute	y	tonos 付きギリシャ語ユプシロン (小文字)
SunFA_Acute	v	tonos 付きギリシャ語オメガ (小文字)
SunFA_Acute	A	tonos 付きギリシャ語アルファ (大文字)

表 5-9 ギリシャ語入力モードの Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
SunFA_Acute	E	tonos 付きギリシャ語エプシロン (大文字)
SunFA_Acute	H	tonos 付きギリシャ語イータ (大文字)
SunFA_Acute	O	tonos 付きギリシャ語オミクロン (大文字)
SunFA_Acute	I	tonos 付きギリシャ語イオータ (大文字)
SunFA_Acute	Y	tonos 付きギリシャ語ユプシロン (大文字)
SunFA_Acute	V	tonos 付きギリシャ語オメガ (大文字)
SunFA_Acute	ギリシャ語アルファ (小文字)	tonos 付きギリシャ語アルファ (小文字)
SunFA_Acute	ギリシャ語エプシロン (小文字)	tonos 付きギリシャ語エプシロン (小文字)
SunFA_Acute	ギリシャ語イータ (小文字)	tonos 付きギリシャ語イータ (小文字)
SunFA_Acute	ギリシャ語イオータ (小文字)	tonos 付きギリシャ語イオータ (小文字)
SunFA_Acute	ギリシャ語オメガ (小文字)	tonos 付きギリシャ語オメガ (小文字)
SunFA_Acute	ギリシャ語オミクロン (小文字)	tonos 付きギリシャ語オミクロン (小文字)
SunFA_Acute	ギリシャ語ユプシロン (小文字)	tonos 付きギリシャ語ユプシロン (小文字)
SunFA_Acute	ギリシャ語アルファ (大文字)	tonos 付きギリシャ語アルファ (大文字)
SunFA_Acute	ギリシャ語エプシロン (大文字)	tonos 付きギリシャ語エプシロン (大文字)
SunFA_Acute	ギリシャ語イータ (大文字)	tonos 付きギリシャ語イータ (大文字)
SunFA_Acute	ギリシャ語イオータ (大文字)	tonos 付きギリシャ語イオータ (大文字)
SunFA_Acute	ギリシャ語オミクロン (大文字)	tonos 付きギリシャ語オミクロン (大文字)

表 5-9 ギリシャ語入力モードの Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	結果
SunFA_Acute	ギリシャ語ユプシロン (大文字)	tonos 付きギリシャ語ユプシロン (大文字)
SunFA_Acute	ギリシャ語オメガ (大文字)	tonos 付きギリシャ語オメガ (大文字)
SunFA_Diaeresis	i	dialytika 付きギリシャ語イオータ (小文字)
SunFA_Diaeresis	y	dialytika 付きギリシャ語ユプシロン (小文字)
SunFA_Diaeresis	I	dialytika 付きギリシャ語イオータ (大文字)
SunFA_Diaeresis	Y	dialytika 付きギリシャ語ユプシロン (大文字)
SunFA_Diaeresis	ギリシャ語イオータ (小文字)	dialytika 付きギリシャ語イオータ (小文字)
SunFA_Diaeresis	ギリシャ語ユプシロン (小文字)	dialytika 付きギリシャ語ユプシロン (小文字)
SunFA_Diaeresis	ギリシャ語イオータ (大文字)	dialytika 付きギリシャ語イオータ (大文字)
SunFA_Diaeresis	ギリシャ語ユプシロン (大文字)	dialytika 付きギリシャ語ユプシロン (大文字)

表 5-10 3つのキーを使用するギリシャ語入力モードの Compose キーシーケンス

その次に押すキー	その次に押すキー	その次に押すキー	結果
セミコロン	コロン	y	dialytika と tonos 付きギリシャ語ユプシロン (小文字)
コロン	セミコロン	y	dialytika と tonos 付きギリシャ語ユプシロン (小文字)
セミコロン	コロン	i	dialytika と tonos 付きギリシャ語イオータ (小文字)
コロン	セミコロン	i	dialytika と tonos 付きギリシャ語イオータ (小文字)

表 5-10 3つのキーを使用するギリシャ語入力モードの Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	その次に押すキー	結果
dead_acute	dead_diaeresis	y	dialytika と tonos 付きギリシャ語ユプシロン (小文字)
dead_diaeresis	dead_acute	y	dialytika と tonos 付きギリシャ語ユプシロン (小文字)
dead_acute	dead_diaeresis	i	dialytika と tonos 付きギリシャ語イオータ (小文字)
dead_diaeresis	dead_acute	i	dialytika と tonos 付きギリシャ語イオータ (小文字)
dead_acute	dead_diaeresis	ギリシャ語ユプシロン (小文字)	dialytika と tonos 付きギリシャ語ユプシロン (小文字)
dead_diaeresis	dead_acute	ギリシャ語ユプシロン (小文字)	dialytika と tonos 付きギリシャ語ユプシロン (小文字)
dead_acute	dead_diaeresis	ギリシャ語イオータ (小文字)	dialytika と tonos 付きギリシャ語イオータ (小文字)
dead_diaeresis	dead_acute	ギリシャ語イオータ (小文字)	dialytika と tonos 付きギリシャ語イオータ (小文字)
SunFA_Acute	SunFA_Diaeresis	i	dialytika と tonos 付きギリシャ語イオータ (小文字)
SunFA_Diaeresis	SunFA_Acute	i	dialytika と tonos 付きギリシャ語イオータ (小文字)
SunFA_Acute	SunFA_Diaeresis	y	dialytika と tonos 付きギリシャ語ユプシロン (小文字)
SunFA_Diaeresis	SunFA_Acute	y	dialytika と tonos 付きギリシャ語ユプシロン (小文字)
SunFA_Acute	SunFA_Diaeresis	ギリシャ語イオータ (小文字)	dialytika と tonos 付きギリシャ語イオータ (小文字)

表 5-10 3つのキーを使用するギリシャ語入力モードの Compose キーシーケンス (続き)

その次に押すキー	その次に押すキー	その次に押すキー	結果
SunFA_Diaeresis	SunFA_Acute	ギリシャ語イオータ (小文字)	dialytika と tonos 付きギリシャ語イオータ (小文字)
SunFA_Acute	SunFA_Diaeresis	ギリシャ語ユプシロン (小文字)	dialytika と tonos 付きギリシャ語ユプシロン (小文字)
SunFA_Diaeresis	SunFA_Acute	ギリシャ語ユプシロン (小文字)	dialytika と tonos 付きギリシャ語ユプシロン (小文字)

表 5-11 4つのキーを使用するギリシャ語入力モードの Compose キーシーケンス

その次に押すキー	その次に押すキー	その次に押すキー	その次に押すキー	結果
セミコロン	コロン	セミコロン	コロン	ギリシャ語 dialytika、tonos
コロン	セミコロン	コロン	セミコロン	ギリシャ語 dialytika、tonos

## ヘブライ語入力モード

ヘブライ語入力モードを選択するには、Compose h h を押すか、入力モード選択ウィンドウで「ヘブライ語 (Hebrew)」を選択します。入力モード選択ウィンドウの表示方法については、121 ページの「入力方式の選択」を参照してください。

次の図にヘブライ語キーボードの配列を示します。

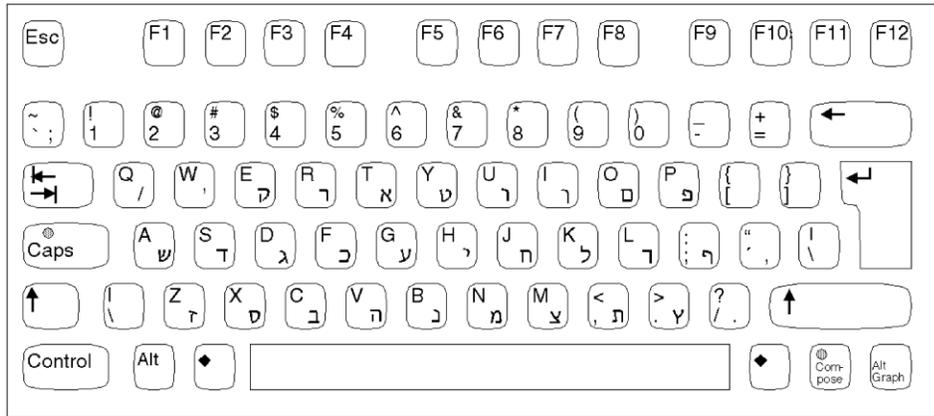


図 5-6 ヘブライ語キーボード

## 日本語入力モード

日本語入力モードを選択するには、Compose ja を押すか、入力モード選択ウィンドウで「日本語 (Japanese)」を選択します。入力モード選択ウィンドウの表示方法については、121 ページの「入力方式の選択」を参照してください。

日本製の日本語入力方式を使用するには、システムに日本語ロケールをインストールし、システムを再起動する必要があります。日本語ロケールをインストールした後は、すべての UTF-8 ロケールで ATOK12 を使用できます。Wnn6 は、ja\_JP.UTF-8 を除き、UTF-8 ロケールでは使用できません。

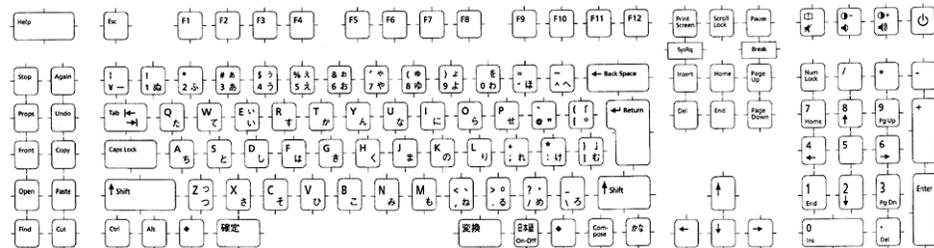


図 5-7 日本語キーボード

---

## 韓国語入力モード

韓国語入力モードを選択するには、Compose k o を押すか、入力モード選択ウィンドウで「韓国語 (Korean)」を選択します。入力モード選択ウィンドウの表示方法については、121 ページの「入力方式の選択」を参照してください。

韓国製の韓国語入力システムを使うには、システムに Ko.UTF-8 ロケールをインストールする必要があります。韓国語入力システムの使い方については、『*Korean Solaris User's Guide*』を参照してください。

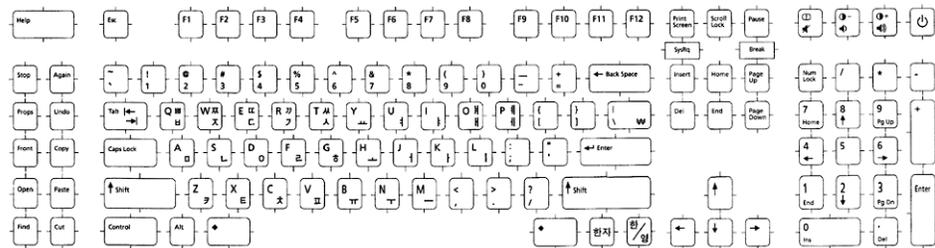


図 5-8 韓国語キーボード

---

## 簡体字中国語入力モード

簡体字中国語入力モードを選択するには、Compose s c を押すか、入力モード選択ウィンドウで「簡体字中国語 (S-Chinese)」を選択します。121 ページの「入力方式の選択」を参照してください。

中国製の簡体字中国語入力システムを使用するには、システムに zh.UTF-8 ロケールをインストールする必要があります。簡体字中国語入力システムの使い方については、『*Simplified Chinese Solaris User's Guide*』を参照してください。

---

## 繁体字中国語入力モード

繁体字中国語入力モードを選択するには、Compose t c を押すか、入力モード選択ウィンドウで「繁体字中国語 (T-Chinese)」を選択します。入力モード選択ウィンドウの表示方法については、121 ページの「入力方式の選択」を参照してください。

台湾製の繁体字中国語入力システムを使うには、システムに zh\_TW.UTF-8 ロケールをインストールする必要があります。繁体字中国語入力システムの使い方の詳細については、『*Traditional Chinese Solaris User's Guide*』を参照してください。

---

## 繁体字中国語 (香港) 入力モード

繁体字中国語 (香港) 入力モードを選択するには、Compose h k を押すか、入力モード選択ウィンドウで「繁体字中国語 (香港) (T-Chinese (Hong Kong))」を選択します。入力モード選択ウィンドウの表示方法については、121 ページの「[入力方式の選択](#)」を参照してください。

香港製の繁体字中国語 (香港) 入力システムを使うには、システムに繁体字中国語 (香港) ロケールをインストールする必要があります。

---

## Unicode 16 進数入力モード

Unicode 16 進数入力モードを選択するには、Compose u h を押すか、入力モード選択ウィンドウで「Unicode 16 進 (Unicode Hex)」を選択します。8 進数入力モードを選択するには、Compose u o を押すか、「Unicode 8 進数 (Unicode Octal)」を選択します。入力モード選択ウィンドウの表示方法については、121 ページの「[入力方式の選択](#)」を参照してください。

この入力モードを使用するには、文字の 16 進数コード値または、8 進数コード値を知っている必要があります。コード値と文字の対応については、『*The Unicode Standard, Version 4.0*』を参照してください。

Unicode 16 進数のコード入力モードで文字を入力する場合は、4 桁の 16 進数を入力します。次はその例です。

- 00A1 (逆感嘆符)
- 03B2 (ギリシャ語ベータの小文字)
- AC00 (韓国語ハングルの音節)
- 30A1 (日本語カタカナの文字「ア」)
- 4E58 (Unified Han 文字)

16 進数に使用する A、B、C、D、E、F は、大文字でも小文字でもかまいません。16 進法の代わりに 8 進法にする場合、0 から 7 までの 8 進法の数字を入力できます。間違えて入力した 16 進数は、Delete キーか Backspace キーで削除できます。

---

## 表検索入力モード

表検索入力モードを選択するには、Compose 1 1 を押すか、入力モード選択ウィンドウで「Unicode 表から選択 (Lookup)」を選択します。入力モード選択ウィンドウの表示方法については、121 ページの「入力方式の選択」を参照してください。

2 つめの検索ウィンドウには、このグループの候補が同時に最大 80 まで表示されます。次の候補群を表示するには Control-n キーを、前の候補語を表示するには Control-p キーをそれぞれ押します。

---

## システム環境

この節では、ロケールの環境変数、TTY 環境の設定、32 ビットおよび 64 ビット STREAMS モジュール、および端末サポートについて説明します。

### ロケールの環境変数

システムに `en_US.UTF-8` がインストールされていることを確認してください。さまざまなカテゴリの現在のロケール設定を調べるには、`locale` ユーティリティを使用します。

```
system% locale
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_ALL=
```

`en_US.UTF-8` ロケールデスクトップ環境を使用するには、まずロケールを選択します。TTY 環境で、まず環境変数 `LANG` を `en_US.UTF-8` に設定してこのロケールを選択します。C シェルの例を以下に示します。

```
system% setenv LANG en_US.UTF-8
```

`LC_ALL`、`LC_COLLATE`、`LC_CTYPE`、`LC_MESSAGES`、`LC_NUMERIC`、`LC_MONETARY`、`LC_TIME` カテゴリには、何も設定されていないか、`en_US.UTF-8` が設定されている必要があります。これらのカテゴリに値が設定されていると、その値が、それより優先度の低い `LANG` 環境変数の値の代わりに使用されます。環境変数の階層については、`setlocale(3C)` のマニュアルページを参照してください。

また、CDE ログイン画面の「オプション」->「言語」メニューで en\_US.UTF-8 を選択することにより、en\_US.UTF-8 環境を起動することもできます。

## TTY 環境の設定

使用している端末や端末エミュレータによっては、コードセット固有の一定の STREAMS モジュールをストリームにプッシュする必要がある場合があります。

STREAMS モジュールの詳細および一般的なストリームについては、『*STREAM Programming Guide*』を参照してください。

次の表に、端末環境で en\_US.UTF-8 ロケールによってサポートされる 64 ビット STREAMS モジュールを示します。詳細は、『*Solaris 64 ビット 開発ガイド*』を参照してください。

表 5-12 en\_US.UTF-8 によりサポートされる STREAMS モジュール

32 ビット STREAMS モジュール	説明
/usr/kernel/strmod/sparcv9/u8lat1	UTF-8 と ISO8859-1 (西ヨーロッパ) のコード変換 STREAMS モジュール
/usr/kernel/strmod/sparcv9/u8lat2	UTF-8 と ISO8859-2 (東ヨーロッパ) のコード変換 STREAMS モジュール
/usr/kernel/strmod/sparcv9/u8koi8	UTF-8 と KOI8-R (キリル文字) のコード変換 STREAMS モジュール

注 - Solaris 10 リリース以降、SPARC sun4u プラットフォーム用の 32 ビットカーネルはサポートされません。表 5-12 は、x86 プラットフォーム用の 32 ビットカーネルにだけ適用されます。詳細は、『*Solaris 10 ご使用にあたって*』を参照してください。

次の表に、en\_US.UTF-8 によってサポートされる 64 ビット STREAMS モジュールを示します。

表 5-13 en\_US.UTF-8 によりサポートされる 64 ビット STREAMS モジュール

64 ビット STREAMS モジュール	説明
/usr/kernel/strmod/sparcv9/u8lat1	UTF-8 と ISO8859-1 (西ヨーロッパ) のコード変換 STREAMS モジュール
/usr/kernel/strmod/sparcv9/u8lat2	UTF-8 と ISO8859-2 (東ヨーロッパ) のコード変換 STREAMS モジュール

表 5-13 en\_US.UTF-8 によりサポートされる 64 ビット STREAMS モジュール (続き)

64 ビット STREAMS モジュール	説明
/usr/kernel/strmod/sparcv9/u8koi8	UTF-8 と KOI8-R (キリル文字) のコード変換 STREAMS モジュール

## ▼ カーネルに STREAMS モジュールをロードする方法

1. スーパーユーザーとして、**Solaris** システムが **64** ビットと **32** ビットのどちらであるかを判断します。

```
system# isainfo -v
```

- 64 ビットの Solaris システムは、次の情報を返します。

```
64-bit sparcv9 applications
32-bit sparc applications
```

- 32 ビットの Solaris システムは、次の情報を返します。

```
32-bit sparc applications
```

- 32 ビットの x86 システムは、次の情報を返します。

```
32-bit i386 applications
```

2. システムにすでに **STREAMS** モジュールがロードされていないかを確認します。

```
system# modinfo | grep modulename
```

u8lat1 などの STREAMS モジュールがすでにインストールされている場合は、出力は次のようになります。

```
system# modinfo | grep u8lat1
```

```
89 ff798000 4b13 18 1 u8lat1 (UTF-8 <--> ISO 8859-1 module)
```

3. モジュールがロードされていない場合は、**modload(1M)** コマンドを使用してモジュールをロードします。

- 32 ビットのシステムの場合、次のように入力します。

```
system# modload /usr/kernel/strmod/u8lat1
```

- 64 ビットのシステムの場合、次のように入力します。

```
system# modload /usr/kernel/strmod/sparcv9/u8lat1
```

適切な u8lat1 STREAMS モジュールがカーネルにロードされ、モジュールをストリームにプッシュできるようになります。

## ▼ カーネルから STREAMS モジュールをアンロードする方法

1. スーパーユーザーとして、カーネルにモジュールがロードされていることを確認します。

u8lat1 がロードされていることを確認するには、次のように入力します。

```
system# modinfo | grep u8lat1
89 ff798000 4b13 18 1 u8lat1 (UTF-8 <--> ISO 8859-1 module)
```

2. **modunload(1M)** コマンドを使用して、カーネルをアンロードします。  
u8lat1 モジュールをアンロードするには、次のように入力します。

```
system# modunload -i 89
```

## ▼ Latin-2 端末および STREAMS モジュールを設定する方法

1. **strchg(1M)** を次のように使用します (2 つめのコマンド行)。

```
system% cat > tmp/mystreams
ttcompat
ldterm
u8lat1
ptem
^D
system% strchg -f /tmp/mystreams
```

strchg(1) を使用する場合、スーパーユーザーかデバイスの所有者としてログインしていることを確認します。

2. **strconf** コマンドを実行して、現在の構成を確認します。

```
system% strconf
ttcompat
ldterm
u8lat1
ptem
pts
system%
```

3. **strchg** コマンドを実行して、元の構成に戻します。

```
system% cat > /tmp/orgstreams
ttcompat
ldterm
ptem
^D
system% strchg -f /tmp/orgstreams
```

## UTF-8 文字の入出力が可能な dtterm、xterm、および端末

Solaris OS の旧リリースとは異なり、UTF-8 コードセットの入出力をサポートする dtterm と xterm 端末エミュレータおよびその他の端末では、ストリームに追加 STREAMS モジュールを置く必要はありません。stty(1) ユーティリティを使用して端末環境を設定した場合は、ldterm モジュールはコードセットに依存せず、Unicode/UTF-8 をサポートします。

Unicode ロケールの端末環境を設定するには、`stty(1)` ユーティリティを使用します。

```
system% /bin/stty defeucw
```

現在の設定を照会する場合は、次のように `stty` に `-a` オプションを指定します。

```
system% /bin/stty -a
```

---

注 - `/usr/ucb/stty` は国際化されていないので、代わりに `/bin/stty` を使ってください。

---

## Latin-1、Latin-2、および KOI8-R の端末サポート

Latin-1 (ISO8859-1)、Latin-2 (ISO8859-2)、または KOI8-R だけをサポートする端末の場合、STREAMS は次のように構成されている必要があります。

```
head <-> ttcompat <-> ldterm <-> u8lat1 <-> TTY
```

これは Latin-1 をサポートする端末用の構成です。Latin-2 端末の場合は、STREAMS モジュール `u8lat1` を `u8lat2` に置き換えます。KOI8-R 端末の場合は、このモジュールを `u8koi8` に置き換えます。

STREAMS モジュールがすでにカーネルにロードされていることを確認してください。

## ~/ .cshrc への設定の保存

必要な STREAMS モジュールがカーネルにロードされたら、以下の行を `.cshrc` ファイル (C シェルの場合) に保存すると便利です。

```
setenv LANG en_US.UTF-8
if ($?USER != 0 && $?prompt != 0) then
    cat >! /tmp/mystreams$$ << _EOF
        ttcompat
        ldterm
        u8lat1
        ptem
    _EOF
    /bin/strchg -f /tmp/mystreams$$
    /bin/rm -f /tmp/mystreams$$
    /bin/stty cs8 -istrip defeucw
endif
```

`.cshrc` ファイルにこれらの行がすでにある場合は、STREAMS モジュールを使用するたびにこれらのコマンドを入力する必要はありません。2 つめの `_EOF` は行の先頭の位置になければなりません。

---

## コード変換

Unicode ロケールがサポートされていることにより、`iconv` および `sdtconvtool` ユーティリティを使って、多くの国の主なコードセット間のさまざまなコード変換を行うことができるようになります。

最新の Solaris 環境では、`geniconvtbl` ユーティリティによって、ユーザー定義のコード変換が行えます。`geniconvtbl` ユーティリティを使って作成したユーザー定義コード変換は、`iconv(1)` と `iconv(3)` の両方で使用できます。このユーティリティの詳細については、`geniconvtbl(1)` と `geniconvtbl(4)` のマニュアルページを参照してください。

`iconv`、`iconv_open`、`sdtconvtool` に適用できる `fromcode` と `toencode` の名前については、[付録 A](#) の表を参照してください。`iconv` コード変換の詳細については、`iconv(1)` と `sdtconvtool(1)` のマニュアルページを参照してください。利用可能なコード変換については、`iconv(5)` のマニュアルページを参照してください。また、[付録 A](#) も参照してください。

---

注 - UCS-2、UCS-4、UTF-16、UTF-32 はすべて、Unicode 4.0 と ISO/IEC 10646-1:2000 規格で定義されているバイト順序マーク (BOM) 文字を認識する Unicode/ISO/IEC 10646 表示書式です。ただし、この文字が文字ストリームの最初の位置になければなりません。UCS-2BE、UCS-4BE、UTF-16BE、UTF-32BE など他の書式は、BOM 文字を認識しない固定幅の Unicode/ISO/IEC 10646 表示書式です。この書式のバイト順序は、ビッグエンディアンであるとみなされます。一方、UCS-2LE、UCS-4LE、UTF-16LE、UTF-32LE などの表示書式のバイト順序はリトルエンディアンであるとみなされます。また、これらの表示書式も、BOM 文字を認識しません。

対応する ISO8859-\* や KOI8-\* のスクリプトや言語については、<http://czyborra.com/charsets/iso8869.html> を参照してください。

---

---

## DtMail サポート

サポートするスクリプトが増えたために、`en_US.UTF-8` ロケールで実行される Solaris DtMail は以下に示す各種の MIME 文字セットをサポートします。

- US-ASCII (7 ビット US ASCII)
- UTF-8 (UCS Transmission Format 8 ビット)
- UTF-7 (UCS Transmission Format 7 ビット)
- ISO-8859-1 (Latin-1)
- ISO-8859-2 (Latin-2)

- ISO-8859-3 (Latin-3)
- ISO-8859-4 (Latin-4)
- ISO-8859-5 (Latin/キリル文字)
- ISO-8859-6 (Latin/アラビア語)
- ISO-8859-7 (Latin/ギリシャ文字)
- ISO-8859-8 (Latin/ヘブライ語)
- ISO-8859-9 (Latin-5)
- ISO-8859-10 (Latin-6)
- ISO-8859-13 (Latin-7/バルト諸言語)
- ISO-8859-14 (Latin-8/ケルト語)
- ISO-8859-15 (Latin-9)
- ISO-8859-16 (Latin-10)
- KOI8-R (キリル文字)
- ISO-2022-JP および EUC-JP (日本語)
- ISO-2022-KR および EUC-KR (韓国語)
- ISO-2022-CN (簡体字中国語)
- ISO-8859-13 (Latin-7/バルト諸言語)
- ISO-8859-14 (Latin-8/ケルト語)
- KOI8-U (キリル文字/ウクライナ語)
- Shift\_JIS (日本語 Shift JIS)
- GB2312 (簡体字中国語 EUC)
- TIS-620 (タイ語)
- UTF-16 (UCS Transmission Format 16 ビット)
- UTF-16BE (UTF-16 Big-Endian)
- UTF-16LE (UTF-16 Little-Endian)
- Windows-1250
- Windows-1251
- Windows-1252
- Windows-1253
- Windows-1254
- Windows-1255
- Windows-1256
- Windows-1257
- Windows-1258
- Big5 (繁体字中国語)
- UTF-32 (UCS Transmission Format 32 ビット)
- UTF-32BE (UTF-32 Big-Endian)
- UTF-32LE (UTF-32 Little-Endian)

このサポートによって、世界中のどの地域のどの文字セットで符号化された電子メールでも、ほとんどの場合は1つのDtMailのインスタンスだけで表示できます。DtMailは、到着した電子メールを、そこに指定されているMIME文字セットとContent-Transfer-Encodingに基づいて復号化します。Windows-125x MIME文字がサポートされています。

送信の場合は、受信者のメールユーザーエージェント(メールクライアント)が理解できるMIME文字セットを指定する必要があります。あるいは、指定せずに、en\_US.UTF-8ロケールによって提供されるデフォルトのMIME文字セットを使用す

することもできます。送信する電子メールの文字セットを切り替えるには、「新規メッセージ (New Message)」ウィンドウで Control-y キーを押すか、「書式 (Format)」メニューボタンをクリックしてから「文字セットを変更 (Change Char Set)」ボタンをクリックします。使用可能な次の文字セットが、「送信 (Send)」ボタンの上の左下隅に表示されます。

電子メールのヘッダーか本体に、その MIME 文字セットで表せない文字が含まれていると、システムはすべての文字を表せる UTF-8 文字セットに自動的に切り替わります。

メッセージに 7 ビット US-ASCII 文字しか使用されていない場合、電子メールのデフォルト MIME 文字セットは US-ASCII です。すべてのメールユーザーエージェントで文字や情報の欠落なしにメッセージを解釈できます。

複数のスクリプトの文字が含まれている電子メールの、デフォルトの MIME 文字セットは UTF-8 です。この場合、UTF-8 の 8 ビット文字はすべて Quoted-Printable エンコーディングによって符号化されます。MIME、登録されている MIME 文字セット、および Quoted-Printable エンコーディングについての詳細は、RFC 2045、2046、2047、2048、2049、2279、2152、2237、1922、1557、1555、1489 を参照してください。



## X アプリケーションで使用されるフォントセット

X アプリケーションで使用されるフォントセットについては、118 ページの「Unicode ロケール: en\_US.UTF-8 サポート」を参照してください。

Solaris デスクトップ環境では、各文字セットにフォントのセットが割り当てられています。

以下のリストは、最新の Solaris 環境でサポートされている Latin-1 のフォントです。

```
-dt-interface system-medium-r-normal-xxs sans utf-10-100-72-72-p-59-iso8859-1
-dt-interface system-medium-r-normal-xs sans utf-12-120-72-72-p-71-iso8859-1
-dt-interface system-medium-r-normal-s sans utf-14-140-72-72-p-82-iso8859-1
-dt-interface system-medium-r-normal-m sans utf-17-170-72-72-p-97-iso8859-1
-dt-interface system-medium-r-normal-l sans utf-18-180-72-72-p-106-iso8859-1
-dt-interface system-medium-r-normal-xl sans utf-20-200-72-72-p-114-iso8859-1
-dt-interface system-medium-r-normal-xxl sans utf-24-240-72-72-p-137-iso8859-1
```

-dt-interface user-\* や -dt-application-\* 別名 (エイリアス) など、CDE 共通のフォント別名については、『*Common Desktop Environment: Internationalization Programmer's Guide*』を参照してください。

en\_US.UTF-8 ロケールでは、utf もこのロケールの共通フォント別名としてサポートされています。これは、X 論理フォント記述名のスタイルフィールドの追加属性として使用されています。したがって、適切なフォントセットを用意するためには、次の例のように、フォントセットの作成で追加のスタイルを指定する必要があります。

```
fs = XCreateFontSet(display,
"-dt-interface system-medium-r-normal-s*utf*",
&missing_ptr, &missing_count, &def_string);
```

## CDE/Motif アプリケーションとしての FontList 定義

フォントセット定義の場合と同じように、アプリケーションの XmFontList リソース定義にも、このロケールでサポートされる追加のスタイル属性を指定する必要があります。

```
*fontList:\
-dt-interface system-medium-r-normal-s*utf*:
```

## 第 6 章

---

# CTL: Complex Text Layout

---

CTL (Complex Text Layout) 機能により、論理テキスト表現と物理テキスト表現の間の複雑な変換が必要な書き込み方式を Motif API がサポートできます。アラビア語、ヘブライ語、およびタイ語はこのような変換を必要とします。CTL Motif では文字の整形 (合字、区別的発音符、セグメントの順序付けなど) を行うことができます。また、静的および動的なテキストウィジェットの変換、双方向テキスト機能、および動的テキストウィジェットでのタブの使用がサポートされています。テキストのレンダリングはレンディション層を通じて処理されるので、その他のウィジェットライブラリの CTL サポートへの拡張は容易です。

この章の内容は次のとおりです。

- 163 ページの「CTL の概要」
- 164 ページの「CTL アーキテクチャの概要」
- 164 ページの「X ベースのアプリケーションに対する CTL サポート」
- 165 ページの「XOC リソース」
- 165 ページの「CTL 技術をサポートするための Motif の変更点」
- 176 ページの「CTL アプリケーションの開発」

---

## CTL の概要

CTL 機能を利用するには、PLS (Portable Layout Services) ライブラリおよび適切な言語エンジンを用意する必要があります。CTL は言語エンジンとのインタフェースとして PLS を使い、レンダリングを実行する前に言語エンジンを使ってテキストを変換します。CTL をサポートするアプリケーションは、CTL のマニュアルに指定されているリソースをあらかじめインクルードする必要があります。

XomCTL でサポートされる、言語の複雑な整形および再配置機能について、以下に具体的に示します。これらの機能は、ロケールごとに異なる PLS モジュールの行う変換を通じて提供されます。

- 配置の変更

- 合字 (複数文字 → 1 文字) および文字の合成 (1 文字 → 複数文字)
- 区別的発音符
- 双方向性
- 対称的スワッピング
- 数値による整形
- 文字列の妥当性

---

## CTL アーキテクチャの概要

図 6-1 に CTL アーキテクチャの構成を示します。図の一番上に位置する Dt アプリケーションは Motif CTL 機能を用いてテキストのレンダリングを行います。これを受けて、Motif はロケールに固有の言語エンジンと PLS 経由でインタフェースし、位置の変更、数値の整形などの操作を行います。

CTL アーキテクチャは、ロケールに固有のエンジンを使用して、新しい言語をサポートします。つまり、Motif や Dt アプリケーションを変更することなくタイ語やベトナム語がサポートされます。

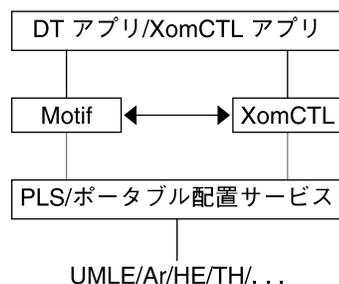


図 6-1 CTL アーキテクチャ

---

## X ベースのアプリケーションに対する CTL サポート

純粋な X Windows アプリケーション (X ベースの端末エミュレータなど) は、XomCTL (X Library Output Module における Complex Text Layout サポート) を通じて CTL サポートを使用することができます。XomCTL では、X11 のダムフォントサポートを始め、Open Source XI18N 実装のすべての機能を使用できます。

---

## XOC リソース

最新の Solaris 環境では、次の XOC リソースが提供されます。

XNText	CTL 操作を行うためのテキストバッファを設定する
XNTextLayoutNumGlyphs	テキストバッファにテキストのグリフがいくつあるかを知らせる
XNTextLayoutModifier	Motif の XmNLayoutModifier と同じ
XNTextLayoutProperty	PLS Property (input-to-output および output-to-input) と同じ
XNTextLayoutMapInpToOut	PLS Property (input-to-output および output-to-input) と同じ
XNTextLayoutMapOutToInp	PLS Property (input-to-output および output-to-input) と同じ

これらのリソースの説明は、X/Open または PLS の仕様に記載されています。

---

## CTL 技術をサポートするための Motif の変更点

CTL 技術は、Motif の次の変更によってサポートされています。

XmNlayoutDirection	オブジェクトの配置を制御する
XmStringDirection	文字列の文字を表示する方向を指定する
XmRendition	新しい擬似リソースを XmRendition に追加する
XmText および XmTextField	XmRenditionに関連付けられたテキストのレイアウト動作に影響を及ぼす
XmTextFieldGetLayoutModifier	レンディションレイアウトオブジェクトのレイアウト修飾子文字列を返す
XmTextGetLayoutModifier	ウィジェットに関連付けられたレンディションの現在のレイアウトオブジェクト設定の値を返す

<code>XmTextFieldSetLayoutModifier</code>	レンディションに関連付けられたレイアウトオブジェクトのレイアウト識別子値を設定する
<code>XmTextSetLayoutModifier</code>	ウィジェットに関連付けられたレンディションのレイアウトオブジェクト設定を変更する
<code>XmStringDirectionCreate</code>	合成文字列を作成する

## XmNlayoutDirection リソース

`XmNlayoutDirection` リソースはオブジェクトレイアウトを制御します。このリソースは、`LayoutObject` の方向の値を次のように使用します。

`XmNlayoutDirection` の概要 (特に、`XmStringDirection` と `XmNlayoutDirection` の関係) については、『*Motif Programmer's Guide (Release 2.1)*』の 11.3 節を参照してください。

### レイアウト方向の決定

`XmNlayoutDirection` が `XmDEFAULT_DIRECTION` として指定されている場合、ウィジェットのレイアウトの方向は作成時に支配的な擬似 XOC に設定されます。動的テキスト (`XmText` および `XmTextField`) では、ウィジェットに使用されている `XmRendition` に関連付けられた擬似 XOC が支配的擬似 XOC になります。静的テキスト (`XmList`、`XmLabel`、`XmLabelG`) のレイアウト方向は、方向の指定されている最初の合成文字列コンポーネントに基づいて設定されます。次の 2 通りのいずれかで指定します。

- コンポーネントの種類が `XmSTRING_COMPONENT_LAYOUT_PUSH` または `XmSTRING_COMPONENT_DIRECTION` の場合
- コンポーネントの種類が `XmSTRING_COMPONENT_LOCALE_TEXT`、`XmSTRING_COMPONENT_WIDECHEAR_TEXT`、または `XmSTRING_COMPONENT_TEXT` の場合。関連付けられた `XmRendition` および `LayoutObject` から設定

`XmNlayoutDirection` が `XmDEFAULT_DIRECTION` として設定されていない場合、`XmNlayoutModifier @ls orientation` の値がレイアウト修飾子文字列に明示的に指定されていない場合は、`XmNlayoutDirection` の値は XOC およびその `LayoutObject` に受け渡されます。

`XmNlayoutDirection` と `XmNlayoutModifier @ls orientation` の両方の値が明示的に指定されている場合、上記の動作の中間になります。ウィジェットオブジェクトのレイアウトは `XmNlayoutDirection` により制御され、レイアウトの変換は `XmNlayoutModifier @ls orientation` の値により制御されます。

詳細については、『CAE Specification: Portable Layout Services: Context-dependent and Directional Text』(The Open Group: Feb 1997; ISBN 1-85912-142-X; document number C616) を参照してください。このマニュアルでは、コンテキストに依存する双方向のテキスト変換を、既存の POSIX ロケールモデルに対する論理拡張として処理するとき使用する、移植可能な機能について説明しています。複合テキスト言語 (Complex-text languages) をサポートしたいシステムプログラマまたはアプリケーションプログラマを対象としています。

## XmStringDirection リソース

XmStringDirection は、文字列に含まれる各文字の表示方向を指定するデータ形式です。

XmNlayoutDirection リソースは、文字列の方向を指定するコンポーネントを持たないすべての複合文字列 (XmString) に対してデフォルトのレンダリング方向を設定します。したがって、XmNlayoutDirection リソースに対して適切な値を設定するだけで、レイアウト方向を設定できます。複合文字列の各コンポーネントの方向を指定する必要はありません。アプリケーションにおいて XmString がレンダリングされる場合、その文字列が方向 (XmStringDirection) を明示して作成されたかどうかを確認されます。文字列が方向コンポーネントを提供しない場合、アプリケーションは XmNlayoutDirection リソースの値から現在のウィジェットを確かめ、その値を XmString のデフォルトのレンダリング方向として使用します。

## XmRendition リソース

CTL により、以下に示す新しい擬似リソースが XmRendition に追加されました。擬似リソースの説明は、表の後にあります。

表 6-1 XmRendition の新しいリソース

名前	クラス/タイプ	アクセス	デフォルト値
XmNfontType	XmCFontType/XmFontType	CSG	XmAS_IS
XmNlayoutAttrObject	XmClayoutAttrObject/String	CG	NULL
XmNlayoutModifier	XmClayoutModifier/String	CSG	NULL

### XmNfontType

レンディションフォントオブジェクトのタイプを指定します。CTL では、このリソースの値が XmFONT\_IS\_XOC の値になっている必要があります。それ以外の値が設定されていると、XmNlayoutAttrObject と XmNlayoutModifier リソースは無視されます。

このリソースの値が `XmFont_IS_XOC` で、`XmNfont` リソースが設定されていない場合、`XmNfontName` リソースの値は、その作成時点で、`XmNlayoutAttrObject` リソースにより指定されたロケール、または現在のロケールの `XOC` オブジェクトに変換されます。さらに、`XmNlayoutModifier` リソースの値は `XOC` に関連付けられた任意のレイアウトオブジェクトに受け渡されます。

#### `XmNlayoutAttrObject`

`AttrObject` 引数を指定します。このリソースは、この `XmRendition` に関連付けられた `XOC` に関連付けられた `Layout Object` の作成に使用されます。この文字列の構文およびセマンティクスについては、レイアウトサービス `m_create_layout()` の仕様を参照してください。 `Layout Modifier Orientation` の出力値と `XmNlayoutDirection` ウィジェットリソースの相互作用については、上述の `XmNfontType` の説明を参照してください。

#### `XmNlayoutModifier`

この `XmRendition` の `XOC` とともに使用されるレイアウトオブジェクトに受け渡されるレイアウト値を指定します。この文字列の構文およびセマンティクスについては、『*CAE Specification*』を参照してください。

`XmRendition{Retrieve,Update}` を用いてこのリソースを設定すると、文字列はこのレンディションに関連付けられた `XOC` に関連付けられたレイアウトオブジェクトに受け渡されます。レイアウトサービスは以上の手続きにより動的に構成されます。 `Orientation`、`Context`、`TypeOfText`、`TextShaping`、または `ShapeCharset` を変更すると予測不能な動作が生じる可能性があります。

## 追加のレイアウト動作

`XmNlayoutModifier` は、`XmRendition` に関連付けられたテキストのレイアウト動作に影響します。たとえば、数値のレイアウトデフォルト値が `NUMERALS_NOMINAL` ならば、`XmNlayoutModifier` に `@ls numerals=nominal:national` または `@ls numerals=:national` を設定することにより、`NUMERALS_NATIONAL` に変更することができます。

レイアウト値は以下の各グループに分類できます。

- エンコーディング記述 – `TypeOfText`、`TextShaping`、`ShapeCharset` (およびロケールのコードセット)  
`TypeOfText` はセグメント順であり、不透明なブロックとして表すことができます。通常、レンディションオブジェクトを通じてこれらの値を動的に変更することは無意味です。ほとんどの場合、予測不能な動作を生じる結果となります。
- レイアウト動作 – `Orientation`、`Context`、`ImplicitAlg`、`Swapping`、および `Numerals`。 `Orientation` と `Context` は、動的に変更すべきではありません。 `ImplicitAlg`、`Swapping`、`Numerals` を変更するようにしてください。
- 編集動作 – `CheckMode`

## XmText と XmTextField リソース

Xm CTL では、XmText と XmTextField が拡張され、ビジュアル的な移動機能と削除機能が追加されました。これは、Motif 2.0 CStext ウィジェット以降の機能です。Motif 2.1 の標準の Text および TextField では、論理的順序と物理的順序は区別されません。「次 (*next*)」および「前方 (*forward*)」は、「右へ」を意味し、「前 (*previous*)」および「後方 (*backward*)」は「左へ」を意味します。一方、CStext ではこれらは区別されており、物理名だけを用いる一連の動作が新たに定義されています。これらには、`left-character()`、`delete-right-word()` などがあります。これらの処理ルーチンはすべてウィジェットの `XmNlayoutDirection` に依存するように定義されていて、適切な「次の - (*next -*)」または「前の - (*previous -*)」の動作が呼び出されます。

Xm CTL の拡張は CStext の場合よりも多少複雑です。Xm CTL の拡張は、ウィジェットのグローバルな方向だけでなく、擬似 XOC (中立的安定を含む) により決定されるカーソルの周囲の物理文字の個々の方向性にも依存します。

新しいリソース名により、選択基準を決定し、レンディションのタグを提供し、そのえ方を決定することができます。

新しい Xm CTL の動作は、`{Move, Delete, Kill}` と `{Left, Right}` と `{Character, Word}` を一つに合成したような動作になっています。次の表に、動作のセットを示します。

表 6-2 Xm CTL の新しいリソース

名前	クラス/タイプ	アクセス	デフォルト値
<code>XmNrenditionTag</code>	<code>XmCRenditionTag/XmRString</code>	CSG	<code>XmFONTLIST_DEFAULT_TAG</code>
<code>XmNalignment</code>	<code>XmCAlignment/XmRAlignment</code>	CSG	<code>XmALIGNMENT_BEGINNING</code>
<code>XmNeditPolicy</code>	<code>XmCEditPolicy/XmREditPolicy</code>	CSG	<code>XmEDIT_LOGICAL</code>

### `XmNrenditionTag`

このウィジェットに対して使用される `XmRendition` のレンディションタグを指定します。`XmRendition` は `XmNrenderTable` リソースに含まれます。

### `XmNalignment`

ウィジェット内で使用されるテキストのそろえ方を指定します。`XmALIGNMENT_END` と `XmALIGNMENT_CENTER` だけがサポートされています。

### `XmNeditPolicy`

ウィジェットに対して使用される編集の基準を `XmEDIT_LOGICAL` または `XmEDIT_VISUAL` として指定します。`XmEDIT_VISUAL` を指定すると、選択、カーソル移動、および削除は視覚的に行われます。このリソースを設定すると、同時に標準のキーボード操作の変換規則および削除イベントも、新しい「視覚的」な動作リストまたは既存の論理動作に変更されます。

## 文字の向き決定動作ルーチン

`forward-cell()` と `backward-cell()` は、指定された方向にある文字の向きの問い合わせを行います。たとえば、方向が左から右ならば、これらの動作は対応する形式として *next-/forward-* または *previous-/ backward-* を呼び出します。

## 文字の向き決定の追加動作

Layout Services 変換の `OutToInp` および Property バッファ(入れ子のレベル用) を使用して、文字の向きを決定します。したがって、ウィジェットの動作は各ロケール固有の変換に依存します。`OutToInp` や Property バッファの情報が正しくないと、ウィジェットが予想外の動作を行うことがあります。特に、Property バッファの情報は大きく影響します。ロケール固有のモジュールがこの仕様の適用範囲から外れている場合、同じテキスト、アプリケーション、リソース値、および `LayoutObject` 構成に対して行われる双方向の編集動作が、プラットフォームによって異なる場合があります。

ビジュアルモードの動作はセルベースの動作を表示します。論理モードの動作は論理的な文字ベースの動作を生じます。たとえば、`delete-right-character()` 操作では、ディスプレイセルに対応した入力バッファの文字が削除されます。つまり、`LayoutObject` 変換の「プロパティ」バイトの「新しいセルのインジケータ」が 1 に設定されている入力バッファの文字 1 個、および「新しいセルのインジケータ」が 0 である後続のすべての文字が消去されます。

Property バッファの詳細については、『*CAE Specification*』の `m_transform_layout()` の仕様を参照してください。

同様に、`backward-character()` では入力バッファ内で挿入点が 1 つ前に移動します。これに伴い、カーソルは出力バッファ内で関連付けられた文字に対応する表示上の位置に再描画されます。したがって、複合表示セル上を移動するにはキーストロークを繰り返す必要があります。区別的発音符または合字など、「新しいセルのインジケータ」が 0 の入力バッファ文字を横切って挿入点が移動するため、カーソルの表示位置は実際には変化しません。

この動作は、削除動作が論理/入力バッファ側から実行されるか、物理的/出力側の表示セルレベルから実行されることを意味します。入力バッファと出力バッファの間には 1 対 1 の対応関係が存在しないため、綿密な意味での 1 文字単位の削除モードは存在しません。たとえば、1 つの物理文字が論理文字の一部分だけを表しているような場合が考えられます。

## XmText 動作ルーチン

XmText の動作ルーチンを以下に示します。

`left-character(extend)`

XmNeditPolicy が XmEDIT\_LOGICAL で、引数なしで呼び出された場合、挿入カーソルは 1 論理文字分だけ前に移動します。挿入カーソルが行の先頭にある場

合、1つ前の行があればその行の最後の論理文字に移動します。1つ前の行が存在しない場合には、カーソル位置は変化しません。

XmNeditPolicy が XmEDIT\_VISUAL ならば、カーソルは1つ左に移動します。挿入カーソルが行の先頭にある場合、1つ前の行があればその行の最後の論理文字に移動します。

extend 引数を指定して left-character() が呼び出されると、現在の選択範囲を延長させながら、引数なしの場合と同じように挿入カーソルを移動します。

left-character() 動作は、reason の値として XmCR\_MOVING\_INSERT\_CURSOR を用いて XmNmotionVerifyCallback プロシージャを呼び出します。また、extend 引数を指定して呼び出された場合、この動作により XmNgainPrimaryCallback プロシージャへの呼び出しが発生する場合があります。詳細については、『*Motif Programmer's Reference*』のコールバックに関する説明を参照してください。

#### right-character(extend)

XmNeditPolicy が XmEDIT\_LOGICAL で、引数なしで呼び出された場合、挿入カーソルは1論理文字分だけ次に移動します。挿入カーソルが行の論理的な最後尾にある場合、次の行があればその行の論理的な先頭に挿入カーソルを移動します。

XmNeditPolicy が XmEDIT\_VISUAL ならば、カーソルは1つ右に移動します。挿入カーソルが行末にある場合、次の行があればその行の先頭に移動します。

extend 引数を指定して呼び出された場合、XmNeditPolicy は、現在の選択範囲を延長させながら、引数なしの場合と同じように挿入カーソルを移動します。

right-character() 動作は、reason の値として

XmCR\_MOVING\_INSERT\_CURSOR を用いて XmNmotionVerifyCallback プロシージャを呼び出します。また、extend 引数を指定して呼び出された場合、この動作により XmNgainPrimaryCallback プロシージャへの呼び出しが発生する場合があります。詳細については、『*Motif Programmer's Reference*』のコールバックに関する説明を参照してください。

#### right-word(extend)

XmNeditPolicy が XmEDIT\_LOGICAL で、引数なしで呼び出された場合、挿入カーソルは次の論理ワードの開始位置に移動します。次の論理ワードが存在しない場合には、現在のワードの論理的な最後尾に挿入カーソルを移動します。挿入カーソルが行の論理的な最後尾にあるか、行の論理的な最後のワード内にある場合、挿入カーソルは次の行の論理的な先頭ワードに移動します。次の行が存在しない場合には、現在のワードの論理的な最後尾に挿入カーソルを移動します。

XmNeditPolicy が XmEDIT\_VISUAL で、right-word() が引数なしで呼び出された場合、右側または行末以降の最初の白い空白文字の後の、白以外の最初の空白文字に挿入カーソルを移動します。

extend 引数を指定して呼び出された場合、現在の選択範囲を延長させながら、引数なしの場合と同じように挿入カーソルを移動します。

left-character() 動作は、reason の値として XmCR\_MOVING\_INSERT\_CURSOR を用いて XmNmotionVerifyCallback プロシージャを呼び出します。また、extend 引数を指定して呼び出された場合、この動作により XmNgainPrimaryCallback プロシージャへの呼び出しが発生する場合があります。詳細については、『*Motif Programmer's Reference*』のコールバックに関する説明を参照してください。

delete-left-character()  
XmNeditPolicy が XmEDIT\_LOGICAL ならば、delete-previous-char() と同じです。XmNeditPolicy が XmEDIT\_VISUAL ならば、次のようになります。すなわち、通常モードで範囲が選択されている場合は、選択された部分が削除されます。範囲が選択されていない場合は、この動作により、挿入カーソルの左側の 1 文字が削除されます。追加モードで、選択されている範囲がある場合は、カーソルは選択範囲から切り離されず、XmNpendingDelete が True に設定され、選択された部分が削除されます。範囲が選択されていない場合は、挿入カーソルの左側の 1 文字が削除されます。結果が選択内容に影響を与える場合があります。

delete-left-character() 動作は、reason の値として XmCR\_MODIFYING\_TEXT\_VALUE を用いて XmNmodifyVerifyCallback プロシージャを呼び出します。また、reason の値として XmCR\_VALUE\_CHANGED を用いて XmNvalueChangedCallback プロシージャを呼び出します。

delete-right-character()  
XmNeditPolicy が XmEDIT\_VISUAL ならば、delete-next-character() と同じです。XmNeditPolicy が XmEDIT\_VISUAL ならば、次のようになります。すなわち、通常モードで範囲が選択されている場合は、選択された部分が削除されます。範囲が選択されていない場合は、挿入カーソルの右側の 1 文字が削除されます。追加モードで、選択されている範囲がある場合は、カーソルは選択範囲から切り離されず、XmNpendingDelete が True に設定され、選択された部分が削除されます。範囲が選択されていない場合は、挿入カーソルの右側の 1 文字が削除されます。この動作が選択内容に影響を与える場合があります。

delete-right-character() 動作は、reason の値として XmCR\_MODIFYING\_TEXT\_VALUE を用いて XmNmodifyVerify-Callback プロシージャを呼び出します。また、reason の値として XmCR\_VALUE\_CHANGED を用いて XmNvalue-ChangedCallback プロシージャを呼び出します。

文字合成、合字、区別的発音符などをサポートするため、セルベースのルーチンが提供されています。つまり、1 個のプレゼンテーションセルを占有する 1 個のグリフを用いて、複数の文字を表すことができます。

XmText セルの動作ルーチンを以下に示します。

backward-cell(extend)  
1 セル分、挿入カーソルを前に移動します。XmNeditPolicy が XmEDIT\_LOGICAL ならば、現在のセルよりも論理的に 1 つ前のセルの先頭に挿入カーソルが移動します。1 つ前のセルが存在しない場合には、挿入カーソルは現在のセルの先頭に移動します。

XmNeditPolicy が XmEDIT\_VISUAL ならば、カーソルはその左側のセルの先頭に移動します。prev-cell() 動作は、reason の値として XmCR\_MOVING\_INSERT\_CURSOR を用いて XmNmotionVerifyCallback プロシージャを呼び出します。また、extend 引数を指定して呼び出された場合、この動作により XmNgainPrimaryCallback プロシージャへの呼び出しが発生する場合があります。詳細については、『*Motif Programmer's Reference*』のコールバックに関する説明を参照してください。

forward-cell(extend)

論理的な次のセルの始点に挿入カーソルを移動します。次のセルが存在しない場合には、現在のセルの最後尾にカーソルを移動します。XmNeditPolicy が XmEDIT\_LOGICAL ならば、次のセルにカーソルを移動します。

XmNeditPolicy が XmEDIT\_VISUAL ならば、カーソルはその右側のセルの先頭に移動します。右側にセルが存在しない場合、カーソルは現在のセルの最後尾に移動します。forward-cell() 動作は、reason の値として XmCR\_MOVING\_INSERT\_CURSOR を用いて XmNmotionVerifyCallback プロシージャを呼び出します。また、extend 引数を指定して呼び出された場合、この動作により XmNgainPrimaryCallback プロシージャへの呼び出しが発生する場合があります。詳細については、『*Motif Programmer's Reference*』のコールバックに関する説明を参照してください。

## XmTextFieldGetLayoutModifier リソース

XmTextFieldGetLayoutModifier() は、レイアウト修飾子の文字列を返します。返された文字列はレンディションに用いられるレイアウトオブジェクトの状態を示します。

XmTextFieldGetLayoutModifier() の構文は次のとおりです。

```
#include <Xm/TextF.h>
string XmTextFieldGetLayoutModifier (Widget widget)
```

XmTextFieldGetLayoutModifier() ウィジェットに関連付けられたレンディションに関連して、現在のレイアウトオブジェクトに設定されている値を読み込みます。レイアウトオブジェクト修飾子の値が簡易関数を用いて変更された場合、XmTextFieldGetLayoutModifier 関数は、変更された値だけでなく、レイアウトオブジェクトに関するすべての状態を返します。

XmTextFieldGetLayoutModifier() はレイアウトオブジェクト修飾子の値を文字列として返します。

## XmTextGetLayoutModifier リソース

XmTextGetLayoutModifier() は、レイアウト修飾子の文字列を返します。返された文字列はレンディションに用いられるレイアウトオブジェクトの状態を示します。

XmTextGetLayoutModifier() の構文は次のとおりです。

```
#include <Xm/Text.h>
String XmTextGetLayoutModifier( Widget widget )
```

`XmTextGetLayoutModifier` ウィジェットに関連付けられたレンディションに関連して、現在のレイアウトオブジェクトに設定されている値を読み込みます。レイアウトオブジェクト修飾子の値が簡易関数を用いて変更された場合、`XmTextGetLayoutModifier` 関数は、変更された値だけでなく、レイアウトオブジェクトに関するすべての状態を返します。

`XmTextGetLayoutModifier` はレイアウトオブジェクト修飾子の値を文字列として返します。

## XmTextFieldSetLayoutModifier リソース

`XmTextFieldSetLayoutModifier()` 関数を設定することにより、レイアウト修飾子の値を設定して、レンディションに用いられるレイアウトオブジェクトの状態を変更することができます。

`XmTextFieldSetLayoutModifier()` の構文は次のとおりです。

```
#include <Xm/TextF.h> \
void XmTextFieldSetLayoutModifier(Widget \
widgetstring layout_modifier)
```

`XmTextFieldSetLayoutModifier` は、ウィジェットに関連付けされたレンディションに関連して、レイアウトオブジェクトの設定値を変更します。レイアウトオブジェクトの修飾子の値が簡易関数を用いて変更された場合、入力パラメータに指定された属性だけが変更されます。その他の属性は影響を受けません。

## XmTextSetLayoutModifier リソース

`XmTextSetLayoutModifier()` 関数を設定することにより、レイアウト修飾子の値を設定して、レンディションに用いられるレイアウトオブジェクトの状態を変更することができます。

`XmTextSetLayoutModifier()` の構文は次のとおりです。

```
#include <Xm/Text.h>
void XmTextSetLayoutModifier( Widget widget ,string layout_modifier)
```

`XmTextSetLayoutModifier` は、ウィジェットに関連付けされたレンディションに関連して、レイアウトオブジェクトの設定値を変更します。レイアウトオブジェクトの修飾子の値が簡易関数を用いて変更された場合、入力パラメータに指定された属性だけが変更されます。その他の属性は影響を受けません。

## XmStringDirectionCreate リソース

XmStringDirectionCreate は、複合文字列を作成します。

XmTextSetLayoutModifier() の構文は次のとおりです。

```
#include <Xm/Xm.h>
XmString XmStringDirectionCreate( direction)
XmStringDirection direction
```

XmStringDirectionCreate は 1 個のコンポーネントを持つ複合文字列を、方向を指定して作成します。一方、XmNlayoutDirection リソースは、文字列の方向を指定するコンポーネントを持たないすべての複合文字列 (XmString) に対してデフォルトのレンダリング方向を設定します。したがって、XmNlayoutDirection リソースに対して適切な値を設定するだけで、レイアウト方向を設定できます。特定の方向コンポーネントを持つ複合文字列を作成する必要はありません。

アプリケーションにおいて XmString がレンダリングされる場合、その文字列が方向 (XmStringDirection) を明示して作成されたかどうかを確認されます。アプリケーションに方向コンポーネントが提供されない場合、アプリケーションは XmNlayoutDirection リソースの値から現在のウィジェットを確かめ、その値を XmString のデフォルトのレンダリング方向として使用します。

---

## UIL 引数

次の表に、UIL 引数の名前と型を示します。

表 6-3 UIL

UIL 引数の名前	引数の型
XmNlayoutAttrObject	文字列
XmNlayoutModifier	文字列
XmNrenditionTag	文字列
XmNalignment	整数
XmNeditPolicy	整数

---

# CTL アプリケーションの開発

この節では、CTL アプリケーションの開発方法について説明します。

## レイアウト方向の制御

複合文字列の方向は、左から右に記述される言語 (英語、スペイン語、フランス語、ドイツ語など) の場合と同様に、右から左に記述される言語 (ヘブライ語、アラビア語など) の場合にも利用可能なデータ構造で保管されます。Motif アプリケーションでは、VendorShell または MenuShell から XmNlayoutDirection リソースを使用してレイアウト方向を設定できます。、マネージャ、プリミティブウィジェット、およびガジェットも XmNlayoutDirection リソースを持ちます。デフォルト値は、同じリソースを持つもっとも近い先祖から継承されます。

XmText ウィジェットに関しては、水平方向と垂直方向も指定する必要があります。layoutDirection に XmRIGHT\_TO\_LEFT を設定すると、文字列の方向は右から左に設定されますが、カーソルは真下に移動します。垂直方向が重要で、上から下を希望する場合には、XmRIGHT\_TO\_LEFT\_TOP\_TO\_BOTTOM を指定します。これにより、コンポーネントは希望どおりまず右から左に、続いて上から下へレイアウトされます。

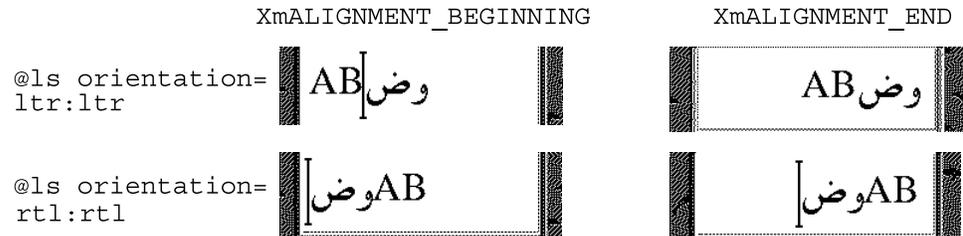
XmText および TextField ウィジェットの動作は、XmRendition の XmNalignment および XmNlayoutModifier リソースにも依存します。これらのリソースおよび XmNlayoutDirection は、Text ウィジェットのレイアウト動作を制御します。このことを図 6-2 で説明します。

図で使用する入力文字列は次のとおりです。

A B و ض

この図の左下のカラムに、XmNlayoutModifier の文字列 @ls orientation= に設定される値を示します。

レイアウト方向: XmLEFT\_TO\_RIGHT



レイアウト方向: XmRIGHT\_TO\_LEFT

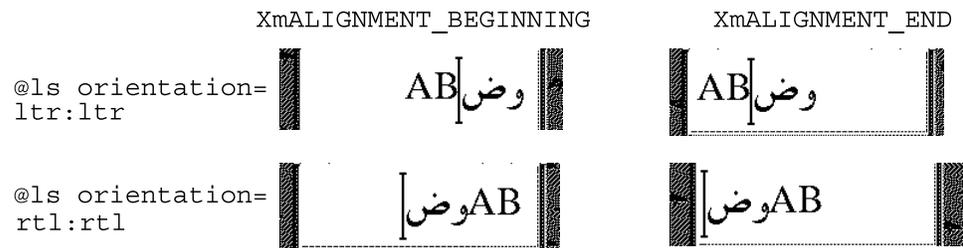


図 6-2 レイアウト方向

図に示すとおり、XmNalignment はレイアウト方向に関連してテキストを右フラッシュするか左フラッシュするかを指定します。XmNlayoutModifier はテキストをセグメントに分割して、向きの値に従って左から右、または右から左に並べ直します。たとえば、XmNlayoutDirection が XmRIGHT\_TO\_LEFT で、XmNalignment の値が XmALIGNMENT\_BEGINNING ならば、文字列は右フラッシュされます。

#### 例 6-1 レンディションの作成

以下に示すプログラム例では、「ArabicShaped」というタグを持つレンディションを使用して、XmNlabelString のタイプが XmCHARSET\_TEXT である XmLabel が作成されます。レンディションの作成には、「ar」（アラビア語ロケールのロケール名に対応）の XmNlayoutAttrObject が使用されます。また、その出力バッファには Numerals の値として NUMERALS\_CONTEXTUAL を、ShapeCharset の値として「iso8859-6」をそれぞれ指定するレイアウト修飾子文字列が使用されています。

ロケール固有のレイアウトモジュールにより、物理文字の出力バッファ内の入力テキストは 16 ビット Unicode コードセットを使用して変換されます。レイアウトロケールが明示的に指定されているため、実行時のロケール設定とは関係なく、このテキストは適切にレンダリングされます。下の例では入力テキストは ISO8859-6 で符号化されています。

```
int n;  
Arg args[10];
```

### 例 6-1 レンディションの作成 (続き)

```
Widget w;
XmString labelString;
XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
    /* alef lam baa noon taa - iso8859-6 */
labelString = XmStringGenerate("\307\344\310\346\312", NULL,
                               XmCHARSET_TEXT, "ArabicShaped");
w = XtVaCreateManagedWidget("a label", xmLabelWidgetClass, parent,
                             XmNlabelString, labelString,
                             XmNlabelType, XmSTRING,
                             NULL);

n = 0;
XtSetArg(args[n], XmNfontName, "-*-medium-r-normal-*-24-*-*-*-*");
    n++;
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
XtSetArg(args[n], XmNlayoutAttrObject, "ar"); n++;
XtSetArg(args[n], XmNlayoutModifier,
          "@ls numerals=:contextual, shapecharset=iso8859-6"); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args,
                              s, n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE_MERGE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
```

### 例 6-2 レンディションの編集

以下に示すプログラム例では、TextField ウィジェットとレンダーテーブルが1個のレンディションにより作成されます。擬似リソース XmNlayoutAttrObject および XmNlayoutModifier は両方とも未指定の状態になっているため、デフォルト値 NULL が使用されます。このことはレンディションに関連付けされたレイアウトオブジェクトが存在する場合、それがデフォルトロケールのものであることを示しています。

サンプルプログラムが正しく機能するためには、使用するロケールのコードセットが ISO 8859-6 で、ロケール固有のレイアウトモードが IMPLICIT\_BASIC アルゴリズムをサポートしている必要があります。コード例では、レンディションの LayoutObject の ImplicitAlg 値がレンディションの XmNlayoutModifier 擬似リソース経由で変更されています。

```
int n;
Arg args[10];
Widget w;
    XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
w = XmCreateTextField(parent, "text field", args, 0);
n = 0;
XtSetArg(args[n], XmNfontName, "-*-medium-r-normal-*-24-*-*-*-*");
    n++;
```

## 例 6-2 レンディションの編集 (続き)

```
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args, n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE_MERGE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
    . . .
n = 0;
XtSetArg(args[n], XmNlayoutModifier, "@ls implicitalg=basic");
    n++;
XmRenditionUpdate(rendition, args, n);
```

## リソースファイルのレンダーテーブルの作成

適切に国際化されたアプリケーションではレンディションおよびレンダーテーブルを、リソースファイルに指定すべきです。レンダーテーブルがファイルとして指定される場合、プログラムの実行ファイルはロケールごとの特定の要求に依存しなくなるため、ロケールの要求に応じて簡単にカスタマイズを行うことができます。

レンダーテーブルをリソースファイルとして指定する場合、次の構文が用いられます。*resource\_spec*: [*tag*[, *tag*]\*]

ここで *tag* には、レンディションの *XmNtag* リソースに適した文字列を指定します。

この行により、指定に従って1つまたは複数のレンディションを含む最初のレンダーテーブルが作成されます。レンディションは指定されたタグに添付されます。

*resource\_spec* [\*|. ] *rendition* [\*|. ] *resource\_name*: *value*

以下の例では、*XmRendition* に関連付けられた CTL リソースのうち、リソースファイルを使用して設定できるものを紹介します。レイアウトオブジェクトを有効にするには、*fontType* が *FONT\_IS\_XOC* に設定されている必要があります。*@ls* を使って指定された *layoutModifier* はレンディションオブジェクトによりレイアウトオブジェクトに受け渡されます。

*layoutModifier* を使ってレイアウトオブジェクト上に設定可能なリソースの完全な一覧については、『*CAE Specification: Portable Layout Services: Context-dependent and Directional Text*』(The Open Group: Feb 1997; ISBN 1-85912-142-X; document number C616) を参照してください。

### 例 6-3 アプリケーションにおけるレンダーテーブルの作成

レンダーテーブルを作成するには、テーブルを構成するレンディションがアプリケーションプログラムにより少なくとも1個は作成されている必要があります。*XmRenderTableAddRenditions()* 関数は新しいレンディションをレンダーテーブルに追加します。新しいレンダーテーブルを作成するには、既存のレンダーテーブル名の代わりに *NULL* を引数に指定して *XmRenderTableAddRenditions()* 関数を使用します。

### 例 6-3 アプリケーションにおけるレンダーテーブルの作成 (続き)

以下に示すプログラム例では、`XmNfontType` を `XmFONT_IS_XOC` に設定して作成されたレンディションを使用してレンダーテーブルが作成されます。

```
int n;
Arg args[10];
Widget w;
XmString labelString;
XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
/* alef lam baa noon taa - iso8859-6 */
labelString = XmStringGenerate("\307\344\310\346\312\312", NULL,
                               XmCHARSET_TEXT, "ArabicShaped");
w = XtVaCreateManagedWidget("a label", xmLabelWidgetClass, parent,
                             XmNlabelString, labelString,
                             XmNlabelType, XmSTRING,
                             NULL);
n = 0;
XtSetArg(args[n], XmNfontName, "-*-medium-r-normal-*-24-*-*-*-*");
n++;
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
XtSetArg(args[n], XmNlayoutAttrObject, "ar"); n++;
XtSetArg(args[n], XmNlayoutModifier,
         "@ls numerals=nominal:contextual, shapecharset=iso8859-6"); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args, n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
```

---

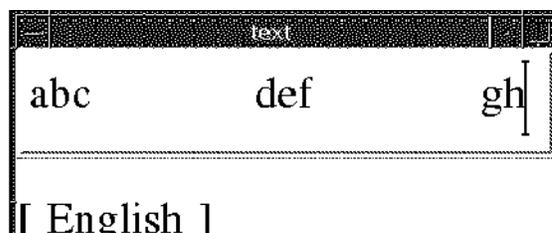
## 水平タブ

復号文字列内にタブ文字を用いてテキストの配置を制御できます。画面上でこれらの文字を解釈するために、ウィジェットはその複合文字列に対する有効なレンディションを参照して、タブストップのリストを検索します。ただし、動的なウィジェット(`TextField` および `XmText`) では、レンディションのタブリソースは使用されません。これらのウィジェットでは、式  $8 * (\text{文字 } 0 \text{ の幅})$  によりタブの幅が計算されます。

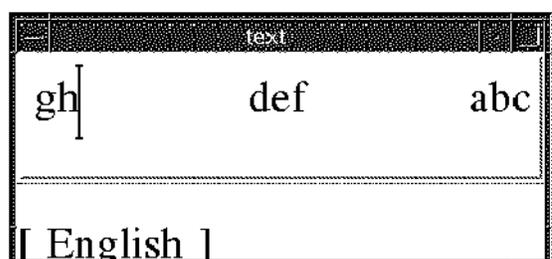
タブは、表示される複合文字列の左マージンからの距離として測定されます。この距離は、レイアウト方向が右から左の場合は、右マージンから測定されます。タブにより挿入される空白文字は、レイアウト方向の指定 (`XmNlayoutDirection`) に従ってテキストの右側または左側に挿入されます。どちら側に挿入されるかはテキストの方向 (たとえばアラビア語の場合は右から左、英語の場合は左から右) には依存しません。

タブに続くテキストの位置は常にタブストップの位置にそろえられます。タブストップはウィジェットの始点から計算されます。この始点の位置は `XmNlayoutDirection` に依存します。タブの動作、およびテキストの方向性や `XmNlayoutDirection` との関連性について、次の図に示します。

この図で用いた入力は `abc\tdef\tgh` です。



レイアウト方向: `XmLEFT_TO_RIGHT`



レイアウト方向: `XmRIGHT_TO_LEFT`

図 6-3 タブ動作

---

## マウスによる選択

一次 (primary) セレクションにはマウスの左ボタンを使用します。このボタンを押すと現在選択されている内容が選択解除され、ボタンが押された時点におけるカーソル位置に挿入カーソルとアンカーが移動します。左ボタンを押しながらマウスをドラッグすると、アンカーの位置からポインタの位置までの間のテキストがすべて選択され、範囲外にあるすべてのテキストが選択解除されます。

選択されたテキストはリソース `XmNeditPolicy` に依存します。このリソースは `XmEDIT_LOGICAL` または `XmEDIT_VISUAL` に設定できます。`XmNeditPolicy` が `XmEDIT_LOGICAL` に設定され、選択されたテキストが双方向ならば、選択されたテキストは個別のセグメントの集合体として表示されます。したがって、視覚的にはなめらかな表示にはなりません。論理バッファ内のテキストは表示内容と 1 対 1 で対応していません。

その結果、双方向テキストの論理文字の連続したバッファは、なめらかに連続した文字にレンダリングされません。一方、`XmNeditPolicy` が `XmEDIT_VISUAL` に設定されると、表示上、選択されたテキストはなめらかになりますが、論理バッファ内では個別のセグメントとして保存されている可能性があります。したがって、双方向テキストに対して一連の選択、削除、挿入操作を同じカーソル位置で実行しても、結果が同じ文字列にはなりません。

---

## キーボードによる選択

マウスで実行可能な選択に関連した操作は、キーボードショートカットを使用して実行することもできます。テキストを選択するには `Shift` と矢印キーを組み合わせます。

選択されたテキストはリソース `XmNeditPolicy` に依存します。このリソースは `XmEDIT_LOGICAL` または `XmEDIT_VISUAL` に設定できます。`XmNeditPolicy` が `XmEDIT_LOGICAL` に設定され、選択されたテキストが双方向ならば、選択されたテキストは視覚的にはなめらかな表示にはなりません。これは、論理バッファ内のテキストが表示内容と 1 対 1 で対応していないためです。その結果、双方向テキストの論理文字の連続したバッファは、なめらかに連続した文字にレンダリングされません。

一方、`XmNeditPolicy` が `XmEDIT_VISUAL` に設定されると、表示上、選択されたテキストはなめらかになりますが、論理バッファ内では個別のセグメントとして保存されている可能性があります。したがって、双方向テキストに対して一連の選択、削除、挿入操作を同じカーソル位置で実行しても、結果が同じ文字列にはなりません。

---

## テキストのリソースおよびジオメトリ

以下のテキストのリソースは、ジオメトリに関連しています。

- レンダーテーブル `XmNrenderTable`。ウィジェットがフォントまたはフォントセット、およびテキストを表示するその他の属性を選択する場合に使用  
`Text` および `Textfield` ウィジェットが使用できるのは、`XmNfontType` などフォント関連のレンディションリソースだけです。これらのウィジェットは `XmNlayoutAttrObject` などのレイアウトオブジェクトの属性も指定できます。これらのウィジェットは、通常ロケール識別子を含みます。また、`XmRendition`

に関連付けられた XOC に関連付けられた Layout Object に受け渡されるレイアウト値を指定する `XmNlayoutModifier` も含みます。

- リソース `XmNwordWrap`。テキストがウィジェットよりも大きい場合に、ワードの境目で行を分割するかどうかを指定  
ワードの境目で行を分割しても新しい行は挿入されません。アラビア語のような続け字で記述される言語では、ワードの長さがウィジェットよりも長ければ、そのワードは次の行に折り返されます。ただし、次の行の先頭の文字は、論理バッファ内の前の文字とは関係なく整形されます。

---

## 移植に関する注意事項

CTL (Complex Text Layout) 対応の機能が組み込まれた新しい Motif ライブラリは `/usr/dt/lib/libXm.so.4` に保管されています。`libXm.so.3` をアプリケーションのリンク先に指定した場合、CTL は利用できません。アプリケーションのリンク先ライブラリを表示するには、`ldd app_name` と入力します。既存のアプリケーションを移植して CTL に対応させるには、以下の手順に従います。

1. Makefile に `-DSUN_CTL` を追加します。  
このフラグは重要で、CTL をサポートするために必要なデータ構造が含まれています。この値はコンパイルの際に設定してください。
2. 既存のアプリケーションを再コンパイルします。  
CTL をサポートする Motif ライブラリ `libXm.so.4` が自動的にリンクされます。
3. アプリケーションリソースファイルに `XmText.translations` リソースを追加します。ロケールのレイアウトエンジンを起動するにはこれらのリソースが必要です。
4. マニュアルに添付されているサンプルアプリケーションを参照します。

---

注 - `fontName` リソースには、そのロケールで適切かつ利用可能なフォント名を指定します。

---

たとえば、`XmTextField` または `XmText` のウィジェットにセルベースの文字の流れ (タイ語など) を使用する場合は、対応するウィジェットの変換を次のように設定します。

```
XmText.translations: #override \n\  
<Key>osfRight:forward-cell() \n\  
<Key>osfLeft:backward-cell() \n\  
<Key>osfDelete:delete-next-cell() \n\  

```

```
<Key>osfBackSpace:delete-previous-cell() \n\
```

## 第 7 章

---

# mp による印刷フィルタの拡張

---

この章では、mp ユーティリティに対する印刷フィルタの拡張について説明します。この章の内容は次のとおりです。

- 185 ページの「UTF-8 用の印刷」
- 186 ページの「mp 印刷フィルタの拡張概要」
- 188 ページの「mp.conf 構成ファイルを使用した地域対応」
- 195 ページの「ロケールに依存する prolog ファイル」
- 194 ページの「prolog ファイルの追加とカスタマイズ」
- 195 ページの「PostScript ファイルのカスタマイズ」
- 198 ページの「.xpr ファイル」

---

## UTF-8 用の印刷

最新の Solaris 環境で拡張された mp 印刷フィルタは、UTF-8 で作成されたプレーンテキストファイルを含むさまざまな入力ファイル形式を印刷できます。この印刷フィルタは、Solaris システムで利用できる、TrueType と Type1 のスケーラブルフォントと、X11 のビットマップフォントを使用します。さらに、このフィルタは、プリンタ常駐のフォントを使用することによって、X プリンタサーバークライアントとして動作することもできます。

このユーティリティの出力は、標準的な PostScript™ であり、任意の PostScript プリンタへ送ることができます。さらに、X プリンタサーバークライアントとして設定された場合、mp は、任意のページ記述言語を出力することができます。mp は印刷サーバーによってもサポートされます。

このユーティリティを使うには、以下のコマンドを入力します。

```
system% mp filename | lp
```

また、mp は stdin ストリーム (標準入力) を受け付けるので、次のようにフィルタとしても使うことができます。

```
system% cat filename | mp | lp
```

このユーティリティを、ラインプリンタの印刷フィルタとして使用することもできます。たとえば、以下の一連のコマンドは、印刷サービス LP に対して、プリンタ lp1 が mp 形式のファイルのみを受け付けるように指示しています。このコマンドは、プリンタ lp1 をポート /dev/ttya にインストールします。詳細については、lpadmin (1M) のマニュアルページを参照してください。

```
system# lpadmin -p lp1 -v /dev/ttya -I MP
system# accept lp1
system# enable lp1
```

lpfilter(1M) を使用して、次のようにこのユーティリティをフィルタとして追加できます。

```
system# lpfilter -f lp1 -F pathname
```

このコマンドは、*pathname* で指定した名前のフィルタ記述ファイルにより、コンバータ (この場合は、mp) が使用可能であることを LP に伝えます。*pathname* の内容は次のとおりです。

```
Input types: simple
Output types: MP
Command: /usr/bin/mp
```

このフィルタは、/usr/bin/mp を使用して、デフォルトのファイル入力を PostScript 出力に変換します。

UTF-8 テキストファイルを印刷するには、次のコマンドを使用します。

```
system% lp -T MP UTF-8-file
```

詳細については、mp(1) のマニュアルページを参照してください。

---

## mp 印刷フィルタの拡張概要

Solaris の最新リリースでは、mp 印刷フィルタが拡張されています。最新の mp は、内部的に次の 3 つのモードで動作します。そして、特定のロケールに対応する出力ファイルを生成し、国際テキストを印刷します。以下のモードが使用できます。

- ロケール固有のフォント構成ファイル `mp.conf` を使用する
- ロケール固有の PostScript プロローグファイル `prolog.ps` を使用する
- Xprt (X Print Server) クライアントとして動作する

次の各項では、特定の印刷方法をどのようなときに使用し、それぞれの方法でどのような構成ファイルやサポートファイルが mp によって使用されるかについて説明します。

## mp でロケール固有のフォント構成ファイル mp.conf を使用する場合

コマンド行に `-D` オプションも `-P` オプションも指定しないと、この印刷方法がデフォルトとして使用されます。ただし、  
`/usr/openwin/lib/locale/$LANG/print` か  
`/usr/lib/lp/locale/$LANG/mp` ディレクトリに `prolog.ps` ファイルがある場合は除きます。`prolog.ps` ファイルがあると、`mp` はファイルに組み込まれている PostScript フォントを使って印刷します。ロケールに `prolog.ps` がある場合でも、`-M` オプションを指定すれば、`prolog.ps` ファイルは無視され、`mp.conf` ファイルがあれば代わりに使用されます。

この方法では、`/usr/lib/lp/locale/$LANG/mp/mp.conf` フォント構成ファイルが使用されます。代替のフォントを使って印刷する必要がなければ、通常はこのファイルを変更する必要はありません。このファイルでは、TrueType、Type 1、または PCF フォントが使用できます。

## mp でロケール固有の PostScript プロローグファイル を使用する場合

`/usr/lib/lp/locale/C/` ディレクトリに、この印刷モードに共通の `.ps` 印刷ページレイアウトがあります。これらのファイルのカスタマイズ方法については、[194 ページの「prolog ファイルの追加とカスタマイズ」](#) を参照してください。

コマンド行に `-D` オプションも `-P` オプションも指定せず、かつ `/usr/openwin/lib/locale/$LANG/print/prolog.ps` が存在する場合は、出力の先頭に `prolog.ps` ファイルが付加されます。`.ps` プロローグページの印刷スタイルによっては、出力の先頭にレイアウトファイルがさらに付加されることがあります。

この印刷方法では、PostScript フォントファイルだけが使用されます。`prolog.ps` ファイルのカスタマイズについては、[194 ページの「prolog ファイルの追加とカスタマイズ」](#) を参照してください。

## mp が Xprt (X Print Server) クライアントとして動作する場合

`mp` を Xprt クライアントとして使用すると、`mp` で Xprt 印刷サービスがサポートするネットワークに接続する任意のプリンタの出力を印刷できます。Xprt クライアントとして `mp` を使用すると、PostScript や PCL の多くのバージョンがサポートされます。

Xprt クライアントは、次のルールに基づいて Xprt サーバーへの接続を試みます。

- `mp` コマンドに `-D printer_name@machine[:dispNum]` または `-P printer_name@machine[:dispNum]` オプションを使用した場合、`mp` は `printer_name` という `machine[:dispNum]` 上の Xprt 印刷サービスに接続を試みます。

上記の `machine[:dispNum]` への接続が失敗した場合、あるいは `-D` または `-P` に指定した引数が `printer_name` だけの場合、`mp` コマンドは `printer_name` 引数をサポートする Xprt サーバーで `XPSEVERLIST` を確認します。たとえば、次のようにします。

```
system% setenv XPSEVERLIST "machine1[:dispNum1] machine2[:dispNum2] ..."
```

- 上記のルールを使用してサーバーが見つからなかった場合、`mp` は、`machine[:dispNum]` に設定された `XPDISPLAY` 環境変数を確認します。たとえば、次のようにします。

```
system% setenv XPDISPLAY "machine[:dispNum]"
```

- `XPDISPLAY` 変数が設定されていない場合、または変数が無効な場合、`mp` はデフォルト表示の 2100 で接続を試みます。デフォルト表示の値も無効な場合、`mp` はエラーメッセージを表示して終了します。

`/usr/lib/lp/locale/C/mp` ディレクトリには、Xprt クライアント用の `.xpr` 印刷ページのサンプルレイアウトファイルが含まれています。サンプルファイルは 300 dpi プリンタ用です。対象のプリンタの dpi 値が異なる場合は、対象のプリンタの解像度に合わせてサンプルファイルの dpi 値が自動的に変換されます。

## mp.conf 構成ファイルを使用した地域対応

構成ファイルを使用することによって、フォントエントリまたはフォントグループエントリの追加や変更を柔軟に行うことができます。

システムのデフォルト構成ファイルは、

`/usr/lib/lp/locale/$LANG/mp/mp.conf` です。`$LANG` は、印刷が行われるロケールのロケール環境変数です。ユーザーは、`-u config.file path` オプションを指定することによって独自の構成ファイルを指定できます。

互換性のために 1 文字としてエンコードされた合字または変形グリフを表示書式と呼びます。`mp.conf` ファイルは、主に、あるロケールの中間コードポイントを、そのコードポイントの印刷に使用されるフォントのエンコーディングでの表示書式に対応付けるために使用されます。

中間コードポイントは、ワイド文字か、PLS (Portable Layout Services) レイヤの出力です。ただし、CTL (Complex Text Layout) 印刷では、中間コードポイントは PLS の出力でなければなりません。`mp` が生成するデフォルトの中間コードポイントは PLS 出力です。

サポートされているフォント形式は PCF (Portable Compiled Format)、TrueType、Type1 です。システム常駐とプリンタ常駐の両方の Type1 フォントがサポートされません。`mp.conf` 構成ファイルの形式と内容については、次の点に注意してください。

- 各行の先頭は、有効なキーワード (命令) でなければなりません。
- キーワードの引数は、キーワードと同じ行になければなりません。
- # 文字で始まる行は、その終わりまでコメントとみなされます。
- `0x` で始まる数値引数は 16 進数の値とみなされます。

次の項目はそれぞれ `mp.conf` ファイルの別々のセクションに定義されます。

- フォント別名の定義
- フォントグループの定義
- 特定のロケールにおける中間コード範囲とフォントグループの対応付け
- 各フォントと共有オブジェクトの対応付け。共有オブジェクトは、中間コードポイントと、フォントエンコーディングを対応付ける

## フォント別名の定義

mp.conf ファイルのフォント別名定義セクションには、印刷に使用する各フォントの別名を定義します。各行の形式は次のとおりです。

```
FontNameAlias font-alias-name font-type font-path
```

### *font-alias-name*

フォントの別名には、通常、フォントのエンコーディング/スクリプト名と、フォントが Roman、Bold、Italic、BoldItalic (R、B、I、BI) のどれであるかを示す文字を指定します。

たとえば、フォントが iso88591 Roman なら、  
/usr/openwin/lib/X11/fonts/75dpi/courR18.pcf.Z に別名として  
iso88591R を割り当てます。

### *font-type*

.pcf フォントには PCF、Adobe Type1 フォントには Type1、TrueType フォントには TrueType が使用可能です。mp.conf ファイルに設定できるフォントはこの3種類だけです。

### *font-path*

フォントファイルの絶対パス名。Type1 のプリンタ常駐フォントの場合は、Helvetica など、フォント名だけを指定します。

定義例は次のとおりです。

```
FontNameAlias prnHelveticaR Type1 Helvetica
```

## フォントグループの定義

同じタイプのフォントをフォントグループとして定義することができます。フォントグループの形式は次のとおりです。

<i>keyword</i>	FontGroup
<i>fontgroupname</i>	フォント群のグループ名
<i>GroupType</i>	フォントタイプ。同じタイプのフォント (PCF、Type1、TrueType) からなるフォントグループを作成します。
<i>Roman</i>	フォントグループで使用する Roman フォントの名前
<i>Bold</i>	フォントグループで使用する Bold フォントの名前

<i>Italic</i>	フォントグループで使用する Italic フォントの名前
<i>BoldItalic</i>	フォントグループで使用する BoldItalic フォントの名前

グループの作成では Roman フォントのエントリだけが必須で、Bold、Italic、BoldItalic フォントの指定は任意です。メールやニュースでは、ヘッダーラインの表示に異なるタイプのフォントが使用されることがあります。Roman フォントだけを定義している場合、他のフォントの代わりに Roman フォントが使用されます。

## 対応付けセクション

mp.conf ファイルの対応付けセクションには、中間コードの範囲とロケールのフォントグループの対応付けを定義します。行の形式は次のとおりです。

<i>keyword</i>	MapCode2Font
<i>range_start</i>	0x で始まる 4 バイトの 16 進数値。1 つまたは複数のフォントグループに対応付けるコード範囲の始めを表します。
<i>range_end</i>	対応付けるコード範囲の終わりを表します。この値が「-」の場合は、1 つのコードポイント範囲だけがターゲットフォントに対応付けられません。
<i>group</i>	Type1、PCF、または TrueType フォントグループ。印刷書式に使用されます。

## 関連付けセクション

mp.conf ファイルの関係付けセクションでは、各フォントと共有オブジェクトを関係付けます。共有オブジェクトとは、中間コードポイントと、フォントエンコーディングを対応付けるものです。行の形式は次のとおりです。

<i>keyword</i>	CnvCode2Font
<i>font alias name</i>	フォントに対して定義されている別名
<i>mapping function</i>	中間コードポイントを受け取り、フォントエンコーディングでのグリフィンデックスを取得し、そのグリフを描くために使用される
<i>file path having mapping function</i>	対応付けの関数が格納されている .so ファイル。dumpcs 内のユーティリティを使用すれば、EUC ロケールの中間コードセットを知ることができる

---

注 - mp (1) で使用される現在の TrueType エンジンでは、書式 4 と PlatformID 3 の cmap しか処理できません。構成できるのは Microsoft .ttf ファイルだけです。さらに、TrueType フォントエンジンが正しく機能するためには、文字マップエンコーディングが Unicode か Symbol でなければなりません。Solaris 環境ではほとんどの .ttf フォントがこれらの制約に従っていますので、Solaris ソフトウェアでは mp.conf ファイル内のすべての TrueType フォントを対応付けることができます。

---

PCF type1 X Logical Fonts Description (XLFD) に対応するフォントを対応付けるための共有オブジェクトを作成できます。次に、中間コード範囲と、XLFD で指定されたエンコーディングを対応付ける共有オブジェクトを作成できます。たとえば、次のようにします。

```
-monotype-arial-bold-r-normal-bitmap-10-100-75-75-p-54-iso8859-8
```

対応する PCF フォントは次のとおりです。

```
/usr/openwin/lib/locale/iso_8859_8/X11/fonts/75dpi/ariabd10.pcf.Z
```

このフォントは ISO 8859-8 でエンコードされているため、共有オブジェクトでは、中間コードと、対応する ISO 8859-8 コードポイントを対応付ける必要があります。

次の XLFD をもつ TrueType フォントが

```
-monotype-arial-medium-r-normal--0-0-0-0-p-0-iso8859-8
```

次のフォントに対応しているとします。

```
/usr/openwin/lib/locale/iso_8859_8/X11/fonts/TrueType/arial__h.ttf
```

この場合は、前の TrueType フォントの cmap エンコーディングが Unicode であるため、中間コードと Unicode を対応付ける必要があります。この TrueType フォントの例では、en\_US.UTF-8 ロケールの中間コードが 0xe50000e9 であるとしています。このコードはヘブライ語文字 (PLS レイヤによって生成された) に対応しています。このフォントは Unicode でエンコードされているため、0xe50000e9 が渡されたときにその出力が Unicode の表現形式に対応するように、対応する .so モジュール内の機能を設計する必要があります。この例では 0x000005d9 になります。

対応付け機能の機能プロトタイプは次のようになります。

```
unsigned int function(unsigned int inter_code_pt)
```

mp.conf には、オプションで次のキーワード/値ペアを指定できます。

```
PresentationForm      WC/PLSOutput
```

デフォルト値は PLSOutput です。WC を指定すると、中間コードポイントとしてワイド文字が生成されます。CTL 印刷では、このデフォルト値を使用してください。

ロケールが CTL 以外のロケールで、キーワードの値が PLSOutput であると、mp はこの値を無視し、ワイド文字コードを生成します。

ロケールが CTL をサポートしている場合は、次の表に示すキーワード/値ペアをオプションで指定できます。これらの変数には、表の中間の列に示す値をどれでも指定できます。

表 7-1 キーワード/値ペア (オプション)

キーワード (オプション)	値 (オプション)	デフォルト
Orientation	ORIENTATION_LTR/	ORIENTATION_LTR
	ORIENTATION_RTL/	
	ORIENTATION_CONTEXTUAL	
Numerals	NUMERALS_NOMINAL/	NUMERALS_NOMINAL
	NUMERALS_NATIONAL/	
	NUMERALS_CONTEXTUAL	
TextShaping	TEXT_SHAPED/	TEXT_SHAPED
	TEXT_NOMINAL/	
	TEXT_SHFORM1/	
	TEXT_SHFORM2/	
	TEXT_SHFORM3/	
	TEXT_SHFORM4	

## ▼ プリンタ常駐フォントの追加方法

次の手順例は、新しい PCF、TrueType、または Type1 のプリンタ常駐フォントを構成ファイルに追加する方法を示しています。

この手順を実行して、現在設定されているフォントを変更します。最初の 2 つの手順では、0x00000021 から 0x0000007f の範囲にある文字を表示するための PCF フォントが TrueType フォントに変更されます。

1. 新しいフォントを追加する前に、現在設定されているフォントに対応する、構成ファイルのさまざまなコンポーネントを確認します。

```
FontNameAlias iso88591R PCF /usr/openwin/lib/X11/fonts/75dpi/courR18PCF.Z
FontNameAlias iso88591B PCF /usr/openwin/lib/X11/fonts/75dpi/courB18PCF.Z
.
.
.
FontGroup iso88591 PCF iso88591R iso88591B
.
.
```

```

.
MapCode2Font 0x00000020 0x0000007f iso88591
.
.
CnvCode2Font iso88591R _xuiso88591 /usr/lib/lp/locale/$LANG/mp/xuiso88591.so
CnvCode2Font iso88591B _xuiso88591 /usr/lib/lp/locale/$LANG/mp/xuiso88591.so

```

たとえば、

`/usr/openwin/lib/locale/ja/X11/fonts/TT/HG-MinchoL.ttf` フォントを `en_US.UTF-8` ロケールに対応付けるとします。`HG-MinchoL.ttf` は Unicode TrueType フォントファイルであるため、`.so` モジュールの対応付け機能を使用して、受け取った `ucs-2` コードポイントを直接返します。

```

unsigned short _ttfjis0201(unsigned short ucs2) {
    return(ucs2);
}

```

- a. 対応付けを `ttfjis0201.c` ファイルに保存します。
- b. 共有オブジェクトファイルを作成します。

```
cc -G -Kpic -o ttfjis0201.so ttfjis0201.c
```

## 2. PCF ファイル

(`/usr/openwin/lib/locale/ja/X11/fonts/75dpi/gotmrk20.pcf.Z` など) に対応付けるには、

`/usr/openwin/lib/locale/ja/X11/fonts/75dpi/fonts.dir` ファイル内の `XLFD` に対応する、次のエンコーディングを確認します。

```
-sun-gothic-medium-r-normal--22-200-75-75-c-100-jisx0201.1976-0
```

- a. `jisx0201` エンコーディングの場合、`ucs-2` と `jisx0201` を対応付ける共有オブジェクトを用意します。`.so` モジュールを作成するための対応付けテーブルを取得します。Unicode ロケールの場合、[ftp.unicode.org/pub/MAPPINGS/](http://ftp.unicode.org/pub/MAPPINGS/) ディレクトリ内で Unicode への文字セットの対応付けを見つけます。
- b. これらの対応付けを使用して、`xu2jis0201.c` ファイルを作成します。

```

unsigned short _xu2jis0201(unsigned short ucs2) {
    if(ucs2 >= 0x20 && ucs2 <= 0x7d )
        return (ucs2);
    if(ucs2==0x203e)
        return (0x7e);
    if(ucs2 >= 0xff61 && ucs2 <= 0xff9f)
        return (ucs2 - 0xff60 + 0xa0);
    return(0);
}

```

- c. 対応付けファイルを作成するときには、`usc-2` を `jisx0201` に対応付けるすべての組み合わせを組み込みます。

```
cc -G -o xu2jis0201.so xu2jis0201.c
```

## ▼ 共有オブジェクトファイルの作成方法

次の手順例は、共有オブジェクトファイルを作成する方法を示しています。

1. フォントを追加するには、次の例の **mp.conf** ファイルに対応する行を編集します。

この例は、TrueType フォントの追加方法を示しています。 .so パスは、`xu2jis0201.so` ファイルをポイントします。

```
FontNameAlias    jis0201R TrueType /home/fn/HG-Minchol.ttf
FontGroup        jis0201 TrueType jis0201R
MapCode2Font     0x0020      0x007f jis0201
CnvCode2Font     jis0201R      _ttfjis0201 <.so path>
```

---

注 - PCF フォントを追加するには、キーワードを TrueType から PCF に変更します。

---

2. 変更した **mp.conf** ファイルを使って **mp** コマンドを呼び出し、新しいフォントの **0x0020** から **0x007f** の範囲を印刷します。

同じ .so ファイルを使って、他の日本語の範囲を対応づけることができます。たとえば、`0x0000FF61` から `0x0000FF9F` の範囲を対応付けできます。

---

注 - 後方互換性を保つためには、  
`/usr/openwin/lib/locale/$LANG/print/prolog.ps` ファイルを使用して、現在のロケールで出力を作成します。 `prolog.ps` ファイルを使用する場合、構成ファイルは必要ありません。

---

`mp.conf` ファイルのサンプルは、`/usr/lib/lp/locale/en_US.UTF-8/mp` ディレクトリを参照してください。

---

## prolog ファイルの追加とカスタマイズ

`prolog` ファイルは、主に次の2つに分類されます。

- PostScript `prolog` ファイル (.ps)
- X プリンタサーバークライアント `prolog` ファイル (.xpr)

## PostScript ファイルのカスタマイズ

PostScript ファイルには、次の種類があります。

- 共通 prolog ファイル
- 印刷レイアウト prolog ファイル

### ロケールに依存する prolog ファイル

prolog.ps ファイルは、特殊なフォントを設定するために使用されます。アプリケーションでは、あらかじめ定義されたこのような PostScript フォント名を使って印刷を行いません。この prolog ファイルには、少なくとも、Desk Set Calendar マネージャや mp で使用する次のフォント名が定義されていなければなりません。

- LC\_Times-Roman
- LC\_Times-Bold
- LC\_Helvetica
- LC\_Helvetica-Bold
- LC\_Courier
- LC\_Helvetica-BoldOblique
- LC\_Times-Italic

次の例では、これらのフォントを使って、指定された特定の文字セットを印刷します。

```
100 100 moveto
/LC_Times-Roman findfont 24 scale font setfont
(Any text string in your locale) show
```

Solaris 地域対応キットには、日本語環境向けのサンプル prolog.ps ファイルが含まれています。このファイルは、/usr/openwin/lib/locale/ja/print/ ディレクトリにも含まれています。

次の例では、既存の prolog.ps ファイルに対する合成フォントの追加や変更について説明します。

```
%
(Foo-Fine) makecodeset12
(Base-Font) makeEUCfont
%
```

たとえば、LC\_Base-Font という合成フォントを作成するとします。LC\_Base-Font は、ロケール文字セットが含まれる *Foo-Fine* フォントと *Base-Font* の合成です。フォントの追加や変更を行う場合には、PostScript プログラミングの詳細についての知識は必要ありません。

prolog.ps ファイルを作成する際には、サンプルファイルが参考になります。サンプルの prolog.ps ファイルでは、makecodeset12 と makeEUCfont という 2 つのルーチンを作成する必要があります。ルーチン makecodeset12 では、ローカルのフォントエンコーディング情報を設定します。このルーチンはロケールによって異なる

ります。ルーチン `makeEUCfont` では、ベースフォントとロケールフォントを結合して合成フォントを作成します。`makecodeset12` や `makeEUCfont` の作成には、PostScript についてよく理解していることが必要です。

`prolog.ps` ファイルのサポートは、後方互換性を維持するために提供されるものです。ロケールの印刷出力を生成するために、新しい `prolog.ps` ファイルを作成しないでください。その場合には、`mp.conf` を使用してください。

`prolog.ps` ファイルのパスは次のとおりです。

```
/usr/openwin/lib/locale/$LANG/print/prolog.ps
```

## 共通 PostScript prolog ファイル

共通 `prolog` ファイルの名前は `mp.common.ps` です。

他のすべてのページレイアウト `prolog` ファイルにこのファイルが含まれていなければなりません。

`mp.common.ps` ファイルは `/usr/lib/lp/locale/C/mp/` ディレクトリにあります。このファイルには、フォントのエンコーディングを標準のエンコーディングから ISO 8859-1 エンコーディングに変換する PostScript ルーチンが含まれています。この `.reencodeISO` ルーチンは、フォントのエンコーディングを変換するために印刷レイアウト `prolog` ファイルから呼び出されます。通常は、この `prolog` ファイルをカスタマイズする必要はありません。独自の `prolog` ファイルを作成する場合は、環境変数 `MP_PROLOGUE` を使って、変更された `prolog` ファイルが含まれているディレクトリをポイントする必要があります。

## 印刷レイアウト prolog ファイル

印刷レイアウト `prolog` ファイル (`mp.*.ps`) には、印刷のページレイアウトを制御するルーチンが含まれています。これらの `prolog` ファイルでは、印刷ページのヘッダーやフッターに、ユーザー名、印刷日付、ページ番号を印刷するだけでなく、その他の情報を提供することができます。たとえば、`prolog` ファイルには、印刷可能領域や、印刷のランドスケープやポートレートモードを指定できます。

印刷レイアウト `prolog` ファイルには、次のものがあります。

- `mp.pro.ps`
- `mp.pro.alt.ps`
- `mp.pro.fp.ps`
- `mp.pro.ps`
- `mp.pro.ts.ps`
- `mp.pro.altl.ps`
- `mp.pro.ff.ps`
- `mp.pro.l.ps`
- `mp.pro.ll.ps`

## ■ mp.pro.tm.ps

prolog ファイルには、一定の標準関数が定義されていなければなりません。これらの関数は、新しい印刷ページの開始や終了、または新しい列の終了で呼び出されます。これらの関数の実装では、印刷出力の印刷属性が定義されます。

mp バイナリは、実行時に次の PostScript 変数を定義します。user name、subject、print time などの動的情報を印刷するためのこれらの変数は、すべての印刷レイアウトファイルで使用できます。これらの変数からとられた情報は、通常、印刷ページのヘッダーやフッターに表示されます。

*User* mp を実行しているユーザーの名前。この情報はシステム passwd ファイルから取得されます。

*MailFor* 印刷する記事のタイプを保持するための変数。この変数の値には、次のものがあります。

- Listing for – 入力テキストファイルの場合
- Mail for – 入力メールファイルの場合
- Article from – 入力ニュースグループからの記事の場合

*Subject* メールやニュースのヘッダーからとられた件名。-s オプションを使用すれば、通常のテキストファイルの他に、メールファイルやニュースファイルに対しても件名を強制的に指定できます。

*Timenow* ヘッダーやフッターに表示する印刷時刻。この情報は localtime() 関数からとられます。

印刷レイアウト prolog ファイルには、次の関数が実装されています。これらの関数ではサブ関数を使用できます。

*endpage* 使用法: page\_number endpage  
印刷ページの終わりに達したときに呼び出されます。この関数はページのグラフィックコンテキストを復元し、showpage を出します。prolog ファイルによっては、ヘッダーやフッターの情報が、カラムバイカラムではなく、ページバイページモードのときに表示されることがあります。この関数の実装方法によっては、ヘッダーやフッターのグレースケールロゼンジを表示するサブ関数を呼び出すこともできます。

*newpage* 使用法: page\_number newpage  
新しいページの表示時に実行されるルーチンまたはコマンド。これらのルーチンの機能には、ランドスケープ印刷モードの設定や、印刷グラフィックコンテキストの保存、ページ座標の変換などがあります。

*endcol* 使用法: page\_number col\_number endcol

新しい印刷位置への移動など、ヘッダーやフッター情報を表示するときに使用します。

新しい印刷レイアウト prolog ファイルを追加するには、次の変数を prolog ファイルに明示的に定義する必要があります。

*NumCols*        印刷ページのカラム数。デフォルトは 2 です。

*PrintWidth*     印刷領域の幅をインチ単位で指定します。デフォルトは 6 です。

*PrintHeight*    印刷領域の高さをインチ単位で指定します。デフォルトは 9 です。

## .xpr ファイル

これらのファイルは、デフォルトで /usr/lib/lp/locale/C/mp/ に格納されています。 .xpr ファイルは、mp.common.ps ファイルを除き、それぞれの PostScript prolog レイアウトファイルに対応しています。MP\_PROLOGUE 環境変数を定義することによって、代わりにの prolog ディレクトリを定義できます。

これらのファイルでは、キーワード/値ペアが使用されます。# で始まる行はコメントとみなされます。特に断りがない限り、それぞれのトークンはスペースで区切られます。 .xpr ファイルの 3 つの主要なセクションは、次のキーワードペアで囲まれます。

- STARTCOMMON/ENDCOMMON
- STARTPAGE/ENDPAGE
- STARTCOLUMN/ENDCOLUMN
- STARTFORCEDPAGE/ENDFORCEDPAGE
- STARTFORCEDCOLUMN/ENDFORCEDCOLUMN

3 つの領域では、一定のキーワード/値ペアが使用できます。各領域については、以下の節で説明します。

### STARTCOMMON/ENDCOMMON キーワード

STARTCOMMON キーワードと ENDCOMMON キーワードの間にあるすべてのキーワード/値ペアは、印刷ページの一般的なプロパティを定義するためのものです。キーワードの有効な値は、それぞれスラッシュ文字 (/) を使用して区切ります。

ORIENTATION 0/1

0 は印刷をポートレートモードで行うことを、1 は印刷をランドスケープモードで行うことをそれぞれ示します。

PAGELength *unsigned-integer*

論理ページ当たりの行数です。

LINELENGTH *unsigned-integer*

行あたりの文字数 (単一カラム文字) です。

NUMCOLS *unsigned-integer*

物理ページ当たりの論理ページ数です。

HDNGFONTSIZE *unsigned-integer*

ヘディングのフォントポイントサイズ (デシポイント単位) です。

BODYFONTSIZE *unsigned-integer*

本体のフォントポイントサイズ (デシポイント単位) です。

PROLOGDPI *unsigned-integer*

現在の .xpr ファイルが作成されているインチ当たりドット数スケールです。

YTEXTBOUNDARY *unsigned-integer*

この y 座標では、ページまたは論理ページにおけるテキスト印刷の境界 (カラム) を設定します。この境界は、テキスト印刷が正しい領域内で行なわれているかどうかを確認するための追加のチェックとして使用されます。この境界は Complex Text Layout や EUC の印刷に必要です。対応するフォントから得られる文字の高さ情報が正しいとは限らないからです。

STARTTEXT *unsigned-integerunsigned-integer*

物理ページの最初の論理ページで実際のテキスト印刷が行なわれるデシポイント x/y ポイントです。

PAGESTRING 0/1

1 は、ヘディングのページ番号の前にページ文字列を付加する必要があることを示します。

0 は、ページ番号だけを表示することを示します。

EXTRAHDNGFONT *font string 1, font string 2, ... font string n*

フォント文字列は X Logical Font Description です。キーワード EXTRAHDNGFONT と、コンマで区切ったフォント名リストを分離するトークンは引用符文字 " 文字です。スペースやタブではありません。これらのフォントは、ヘディングの印刷時に、組み込まれているフォントよりも優先して使用されます。通常、EXTRABODYFONT は、  
/usr/openwin/server/etc/XpConfig/C/print/models/<model name>/fonts ディレクトリに設定されているプリンタ常駐フォントを割り当てるために使用されます。

fonts.dir ファイルには、プリンタ常駐フォントの XLFD が含まれています。

通常、.xpr ファイルでは、フォントが次の例のように指定されています。

```
"-monotype-Gill Sans-Regular-r-normal- *-%d-*-p-0-iso8859-2"
```

「%d」(存在する場合は、mp によって、.xpr ファイルに指定されているヘディングフォントのポイントサイズで置き換えられます。x 解像度と y 解像度は \* で指定されます。平均幅のフィールドは 0 に設定され、可能な限りスケラブルなフォントを選択することが示されます。特定のフォント名を指定することもできます。

EXTRABODYFONT *font string 1, font string 2, ... font string n*

これらのフォントがページ本体の印刷に使用されることを除けば、EXTRAHDNGFONT と同じです。

XDISPLACEMENT *signed/unsigned int*

ページに適用する x 座標変位を指定します。ページの内容は、この変位分だけ x 方向にシフトされます。この値は +ve でも -ve でもかまいません。

YDISPLACEMENT *signed/unsigned int*

内容が y 方向にシフトされることを除けば、x 変位と同じです。

これらのキーワードは、プリンタのマージン幅が標準的でないためにページの内容をシフトしたいときに有効です。

## STARTPAGE/ENDPAGE キーワード

このセクションのキーワード値ペアは、STARTPAGE と ENDPAGE キーワードで囲まれた部分です。このセクションには、物理ページに適用する描画やヘディングの情報を指定します。物理ページには多数の論理ページが含まれていることがありますが、これらのキーワードで囲まれる描画ルーチンは物理ページに 1 度だけ適用されます。

有効な描画エンティティは、LINE と ARC です。XDrawLine() および XDrawArc() 関数は、これらのキーワードの値に対して実行されます。

このセクション内の寸法は PROLOGDPI 単位でマップされます。傾きは度数で表されます。

LINE x1 y1 x2 y2	線をつなぐ両端を無符号の x/y 座標で定義します。
ARC x y width height angle1 angle2	x と y は共に円弧の基点を表す無符号の整数です。幅と高さはそれぞれ円弧の幅と高さを表す無符号の整数です。
USERSTRINGPOS x y	ヘディングに印刷するユーザー情報の位置を表す無符号の座標です。
TIMESTRINGPOS x y	ヘディングに印刷する印刷時刻の位置を表す無符号の座標です。
PAGESTRINGPOS x y	ページに印刷するページ文字列の位置を表す無符号の座標です。
SUBJECTSTRINGPOS x y	ページに印刷する件名の位置を表す無符号の座標です。

## STARTFORCEDPAGE/ENDFORCEDPAGE セクション

mp に -n オプションを指定すると、STARTPAGE/ENDPAGE セクションに指定したデコレーションはまったく印刷されません。しかし、-n オプションを指定しても、STARTFORCEDPAGE/ENDFORCEDPAGE セクションのデコレーションは印刷されます。

## STARTCOLUMN/ENDCOLUMN セクション

このセクションの各エンティティが NUMCOLS の回数だけ物理ページに適用されることを除けば、すべてのキーワードは 200 ページの「STARTPAGE/ENDPAGE キーワード」と同じです。たとえば、NUMCOLS が 3 なら、物理ページの印刷可能域が 3 つに分割され、線や円弧、エンディングの指定がページごとに 3 回印刷されます。

## STARTFORCEDCOLUMN/ENDFORCEDCOLUMN セクション

mp に -n オプションを指定すると、STARTCOLUMN/ENDCOLUMN セクションに指定したデコレーションはまったく印刷されません。しかし、-n オプションを指定しても、STARTFORCEDCOLUMN/ENDFORCEDCOLUMN セクションのデコレーションは印刷されます。

## 新しい .xpr ファイルの作成

新しい .xpr prolog ファイルを作成する際には、デフォルトと異なる値だけを指定します。

以下の表は、.xpr ファイルの STARTCOMMON/ENDCOMMON セクションでキーワードの値が指定されていない場合に、mp プログラムが使用するデフォルト値の一覧です。

表 7-2 STARTCOMMON/ENDCOMMON キーワード値

キーワード	値
ORIENTATION	0
PAGELLENGTH	60
LINELENGTH	80
YTEXTBOUNDARY	3005
NUMCOLS	01
HDNGFONTSIZE	120
PROLOGDPI	300
STARTTEXT	135 280
PAGESTRING	0

STARTPAGE/ENDPAGE および STARTCOLUMN/ENDCOLUMN で囲まれる他の 2 つのセクションでは、デフォルト値は必要ありません。

デコレーションなしのページを作成する場合は、ページ当たり 4 つの論理ページをポートレート形式で作成します。次のセクションと値を指定します。

```
STARTCOMMON
NUMCOLS 04
LINELENGTH 20
ENDCOMMON
```

デコレーションなしのページを作成する場合は、次の2つのセクションを指定する必要はありません。

```
STARTPAGE/ENDPAGE
STARTCOLUMN/ENDCOLUMN
```

印刷ページにデコレーションを入れない場合は、これらのパラメータを設定する必要はありません。PROLOGDPI キーワードを指定しなければ、すべての解像度は 300 dpi (デフォルト) で表されます。対象のプリンタの解像度がこれと異なる場合は、プリンタの解像度に合わせて .xpr ファイルが自動的に拡大縮小されます。

.xpr ファイルを作成する前に、用紙のサイズを知っている必要があります。解像度が 300 dpi のプリンタの場合、米国で使用される 8.5x11 インチの用紙の合計サイズは 2550X3300 です。ほとんどのプリンタでは、用紙の左上隅に印刷することはできません。代わりに、用紙の周囲にはマージンが割り当てられます。0,0 から印刷しようとしても、ページの左上隅には印刷されません。新しい .xpr ファイルを作成する場合には、この点に注意してください。

## 付録 A

# iconv コード変換

この付録では、最新の Solaris オペレーティングシステムで使用できる Unicode 関連のコード変換モジュールを示します。

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール

変換元コード (シンボル)	変換先コード (シンボル)
646 (ISO 646)	UCS-2
646 (ISO 646)	USC-2BE
646 (ISO 646)	UCS-2LE
646 (ISO 646)	USC-4
646 (ISO 646)	USC-4BE
646 (ISO 646)	USC-4LE
646 (ISO 646)	UTF-8
646 (ISO 646)	UTF-16
646 (ISO 646)	UTF-16BE
646 (ISO 646)	UTF-16LE
646 (ISO 646)	UTF-32
646 (ISO 646)	UTF-32BE
646 (ISO 646)	UTF-32LE
ISO8859-11	UTF-8
8859-1 (ISO8859-1)	UCS-2
8859-1 (ISO8859-1)	UCS-2BE
8859-1 (ISO8859-1)	UCS-2LE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
8859-1 (ISO8859-1)	UCS-4
8859-1 (ISO8859-1)	UCS-4BE
8859-1 (ISO8859-1)	UCS-4LE
8859-1 (ISO8859-1)	UTF-8
8859-1 (ISO8859-1)	UTF-16
8859-1 (ISO8859-1)	UTF-16BE
8859-1 (ISO8859-1)	UTF-16LE
8859-1 (ISO8859-1)	UTF-32
8859-1 (ISO8859-1)	UTF-32BE
8859-1 (ISO8859-1)	UTF-32LE
8859-2 (ISO8859-2)	UCS-2
8859-2 (ISO8859-2)	UCS-2BE
8859-2 (ISO8859-2)	UCS-2LE
8859-2 (ISO8859-2)	UCS-4
8859-2 (ISO8859-2)	UCS-4BE
8859-2 (ISO8859-2)	UCS-4LE
8859-2 (ISO8859-2)	UTF-8
8859-2 (ISO8859-2)	UTF-16
8859-2 (ISO8859-2)	UTF-16BE
8859-2 (ISO8859-2)	UTF-16LE
8859-2 (ISO8859-2)	UTF-32
8859-2 (ISO8859-2)	UTF-32BE
8859-2 (ISO8859-2)	UTF-32LE
8859-3 (ISO8859-3)	UCS-2
8859-3 (ISO8859-3)	UCS-2BE
8859-3 (ISO8859-3)	UCS-2LE
8859-3 (ISO8859-3)	UCS-4
8859-3 (ISO8859-3)	UCS-4BE
8859-3 (ISO8859-3)	UCS-4LE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
8859-3 (ISO8859-3)	UTF-8
8859-3 (ISO8859-3)	UTF-16
8859-3 (ISO8859-3)	UTF-16BE
8859-3 (ISO8859-3)	UTF-16LE
8859-3 (ISO8859-3)	UTF-32
8859-3 (ISO8859-3)	UTF-32BE
8859-3 (ISO8859-3)	UTF-32LE
8859-4 (ISO8859-4)	UCS-2
8859-4 (ISO8859-4)	UCS-2BE
8859-4 (ISO8859-4)	UCS-2LE
8859-4 (ISO8859-4)	UCS-4
8859-4 (ISO8859-4)	UCS-4BE
8859-4 (ISO8859-4)	UCS-4LE
8859-4 (ISO8859-4)	UTF-8
8859-4 (ISO8859-4)	UTF-16
8859-4 (ISO8859-4)	UTF-16BE
8859-4 (ISO8859-4)	UTF-16LE
8859-4 (ISO8859-4)	UTF-32
8859-4 (ISO8859-4)	UTF-32BE
8859-4 (ISO8859-4)	UTF-32LE
8859-5 (ISO8859-5)	UCS-2
8859-5 (ISO8859-5)	UCS-2BE
8859-5 (ISO8859-5)	UCS-2LE
8859-5 (ISO8859-5)	UCS-4
8859-5 (ISO8859-5)	UCS-4BE
8859-5 (ISO8859-5)	UCS-4LE
8859-5 (ISO8859-5)	UTF-8
8859-5 (ISO8859-5)	UTF-16
8859-5 (ISO8859-5)	UTF-16BE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
8859-5 (ISO8859-5)	UTF-16LE
8859-5 (ISO8859-5)	UTF-32
8859-5 (ISO8859-5)	UTF-32BE
8859-5 (ISO8859-5)	UTF-32LE
8859-6 (ISO8859-6)	UCS-2
8859-6 (ISO8859-6)	UCS-2BE
8859-6 (ISO8859-6)	UCS-2LE
8859-6 (ISO8859-6)	UCS-4
8859-6 (ISO8859-6)	UCS-4BE
8859-6 (ISO8859-6)	UCS-4LE
8859-6 (ISO8859-6)	UTF-8
8859-6 (ISO8859-6)	UTF-16
8859-6 (ISO8859-6)	UTF-16BE
8859-6 (ISO8859-6)	UTF-16LE
8859-6 (ISO8859-6)	UTF-32
8859-6 (ISO8859-6)	UTF-32BE
8859-6 (ISO8859-6)	UTF-32LE
8859-7 (ISO8859-7)	UCS-2
8859-7 (ISO8859-7)	UCS-2BE
8859-7 (ISO8859-7)	UCS-2LE
8859-7 (ISO8859-7)	UCS-4
8859-7 (ISO8859-7)	UCS-4BE
8859-7 (ISO8859-7)	UCS-4LE
8859-7 (ISO8859-7)	UTF-8
8859-7 (ISO8859-7)	UTF-16
8859-7 (ISO8859-7)	UTF-16BE
8859-7 (ISO8859-7)	UTF-16LE
8859-7 (ISO8859-7)	UTF-32
8859-7 (ISO8859-7)	UTF-32BE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
8859-7 (ISO8859-7)	UTF-32LE
8859-8 (ISO8859-8)	UCS-2
8859-8 (ISO8859-8)	UCS-2BE
8859-8 (ISO8859-8)	UCS-2LE
8859-8 (ISO8859-8)	UCS-4
8859-8 (ISO8859-8)	UCS-4BE
8859-8 (ISO8859-8)	UCS-4LE
8859-8 (ISO8859-8)	UTF-8
8859-8 (ISO8859-8)	UTF-16
8859-8 (ISO8859-8)	UTF-16BE
8859-8 (ISO8859-8)	UTF-16LE
8859-8 (ISO8859-8)	UTF-32
8859-8 (ISO8859-8)	UTF-32BE
8859-8 (ISO8859-8)	UTF-32LE
8859-9 (ISO8859-9)	UCS-2
8859-9 (ISO8859-9)	UCS-2BE
8859-9 (ISO8859-9)	UCS-2LE
8859-9 (ISO8859-9)	UCS-4
8859-9 (ISO8859-9)	UCS-4BE
8859-9 (ISO8859-9)	UCS-4LE
8859-9 (ISO8859-9)	UTF-8
8859-9 (ISO8859-9)	UTF-16
8859-9 (ISO8859-9)	UTF-16BE
8859-9 (ISO8859-9)	UTF-16LE
8859-9 (ISO8859-9)	UTF-32
8859-9 (ISO8859-9)	UTF-32BE
8859-9 (ISO8859-9)	UTF-32LE
8859-10 (ISO8859-10)	UCS-2
8859-10 (ISO8859-10)	UCS-2BE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
8859-10 (ISO8859-10)	UCS-2LE
8859-10 (ISO8859-10)	UCS-4
8859-10 (ISO8859-10)	UCS-4BE
8859-10 (ISO8859-10)	UCS-4LE
8859-10 (ISO8859-10)	UTF-8
8859-10 (ISO8859-10)	UTF-16
8859-10 (ISO8859-10)	UTF-16BE
8859-10 (ISO8859-10)	UTF-16LE
8859-10 (ISO8859-10)	UTF-32
8859-10 (ISO8859-10)	UTF-32BE
8859-10 (ISO8859-10)	UTF-32LE
8859-13 (ISO8859-13)	UCS-2
8859-13 (ISO8859-13)	UCS-2BE
8859-13 (ISO8859-13)	UCS-2LE
8859-13 (ISO8859-13)	UCS-4
8859-13 (ISO8859-13)	UCS-4BE
8859-13 (ISO8859-13)	UCS-4LE
8859-13 (ISO8859-13)	UTF-8
8859-13 (ISO8859-13)	UTF-16
8859-13 (ISO8859-13)	UTF-16BE
8859-13 (ISO8859-13)	UTF-16LE
8859-13 (ISO8859-13)	UTF-32
8859-13 (ISO8859-13)	UTF-32BE
8859-13 (ISO8859-13)	UTF-32LE
8859-14 (ISO8859-14)	UCS-2
8859-14 (ISO8859-14)	UCS-2BE
8859-14 (ISO8859-14)	UCS-2LE
8859-14 (ISO8859-14)	UCS-4
8859-14 (ISO8859-14)	UCS-4BE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
8859-14 (ISO8859-14)	UCS-4LE
8859-14 (ISO8859-14)	UTF-8
8859-14 (ISO8859-14)	UTF-16
8859-14 (ISO8859-14)	UTF-16BE
8859-14 (ISO8859-14)	UTF-16LE
8859-14 (ISO8859-14)	UTF-32
8859-14 (ISO8859-14)	UTF-32BE
8859-14 (ISO8859-14)	UTF-32LE
8859-15 (ISO8859-15)	UCS-2
8859-15 (ISO8859-15)	UCS-2BE
8859-15 (ISO8859-15)	UCS-2LE
8859-15 (ISO8859-15)	UCS-4
8859-15 (ISO8859-15)	UCS-4BE
8859-15 (ISO8859-15)	UCS-4LE
8859-15 (ISO8859-15)	UTF-8
8859-15 (ISO8859-15)	UTF-16
8859-15 (ISO8859-15)	UTF-16BE
8859-15 (ISO8859-15)	UTF-16LE
8859-15 (ISO8859-15)	UTF-32
8859-15 (ISO8859-15)	UTF-32BE
8859-15 (ISO8859-15)	UTF-32LE
8859-16 (ISO8859-16)	UCS-2
8859-16 (ISO8859-16)	UCS-2BE
8859-16 (ISO8859-16)	UCS-2LE
8859-16 (ISO8859-16)	UCS-4
8859-16 (ISO8859-16)	UCS-4BE
8859-16 (ISO8859-16)	UCS-4LE
8859-16 (ISO8859-16)	UTF-8
8859-16 (ISO8859-16)	UTF-16

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
8859-16 (ISO8859-16)	UTF-16BE
8859-16 (ISO8859-16)	UTF-16LE
8859-16 (ISO8859-16)	UTF-32
8859-16 (ISO8859-16)	UTF-32BE
8859-16 (ISO8859-16)	UTF-32LE
ACE	UTF-8
ACE-ALLOW-UNASSIGNED	UTF-8
eucJP	UTF-8
gb2312	UTF-8
iso2022	UTF-8
ko_KR-cp933	UTF-8
ko_KR-euc	UTF-8
ko_KR-iso2022-7	UTF-8
ko_KR-johap	UTF-8
ko_KR-johap92	UTF-8
zh_TW-euc	UTF-8
zh_TW-cp937	UTF-8
zh_TW-iso2022-7	UTF-8
GBK	UTF-8
FujitsuJEF-ascii-code	UTF-8
FujitsuJEF-ascii-face	UTF-8
FujitsuJEF-kana-code	UTF-8
FujitsuJEF-kana-face	UTF-8
HitachiKEIS83	UTF-8
HitachiKEIS90	UTF-8
ISO-2022-JP	UTF-8
KOI8-R	UCS-2
KOI8-R	UCS-2BE
KOI8-R	UCS-2LE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
KOI8-R	UCS-4
KOI8-R	UCS-4BE
KOI8-R	UCS-4LE
KOI8-R	UTF-8
KOI8-R	UTF-16
KOI8-R	UTF-16BE
KOI8-R	UTF-16LE
KOI8-R	UTF-32
KOI8-R	UTF-32BE
KOI8-R	UTF-32LE
KOI8-U	UCS-2
KOI8-U	UCS-2BE
KOI8-U	UCS-2LE
KOI8-U	UCS-4
KOI8-U	UCS-4BE
KOI8-U	UCS-4LE
KOI8-U	UTF-8
KOI8-U	UTF-16
KOI8-U	UTF-16BE
KOI8-U	UTF-16LE
KOI8-U	UTF-32
KOI8-U	UTF-32BE
KOI8-U	UTF-32LE
NECJIPS	UTF-8
PCK	UTF-8
PTCP154	UCS-2
PTCP154	UCS-2BE
PTCP154	UCS-2LE
PTCP154	UCS-4

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
PTCP154	UCS-4BE
PTCP154	UCS-4LE
PTCP154	UTF-16
PTCP154	UTF-16BE
PTCP154	UTF-16LE
PTCP154	UTF-32
PTCP154	UTF-32BE
PTCP154	UTF-32LE
PTCP154	UTF-8
UCS-2	646 (ISO 646)
UCS-2	8859-1 (ISO8859-1)
UCS-2	8859-2 (ISO8859-2)
UCS-2	8859-3 (ISO8859-3)
UCS-2	8859-4 (ISO8859-4)
UCS-2	8859-5 (ISO8859-5)
UCS-2	8859-6 (ISO8859-6)
UCS-2	8859-7 (ISO8859-7)
UCS-2	8859-8 (ISO8859-8)
UCS-2	8859-9 (ISO8859-9)
UCS-2	8859-10 (ISO8859-10)
UCS-2	8859-13 (ISO8859-13)
UCS-2	8859-14 (ISO8859-14)
UCS-2	8859-15 (ISO8859-15)
UCS-2	8859-16 (ISO8859-16)
UCS-2	KOI8-R
UCS-2	KOI8-U
UCS-2	PTCP154
UCS-2BE	PTCP154
UCS-2LE	PTCP154

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UCS-4	PTCP154
UCS-4BE	PTCP154
UCS-4LE	PTCP154
UTF-16	PTCP154
UTF-16BE	PTCP154
UTF-16LE	PTCP154
UTF-32	PTCP154
UTF-32BE	PTCP154
UTF-32LE	PTCP154
UTF-8	PTCP154
UCS-2	UCS-4
UCS-2	UCS-4BE
UCS-2	UCS-4LE
UCS-2	UTF-7
UCS-2	UTF-8
UCS-2BE	646 (ISO 646)
UCS-2BE	8859-1 (ISO8859-1)
UCS-2BE	8859-2 (ISO8859-2)
UCS-2BE	8859-3 (ISO8859-3)
UCS-2BE	8859-4 (ISO8859-4)
UCS-2BE	8859-5 (ISO8859-5)
UCS-2BE	8859-6 (ISO8859-6)
UCS-2BE	8859-7 (ISO8859-7)
UCS-2BE	8859-8 (ISO8859-8)
UCS-2BE	8859-9 (ISO8859-9)
UCS-2BE	8859-10 (ISO8859-10)
UCS-2BE	8859-13 (ISO8859-13)
UCS-2BE	8859-14 (ISO8859-14)
UCS-2BE	8859-15 (ISO8859-15)

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UCS-2BE	8859-16 (ISO8859-16)
UCS-2BE	KOI8-R
UCS-2BE	KOI8-U
UCS-2BE	UCS-4
UCS-2BE	UCS-4BE
UCS-2BE	UCS-4LE
UCS-2BE	UTF-8
UCS-2LE	646 (ISO 646)
UCS-2LE	8859-1 (ISO8859-1)
UCS-2LE	8859-2 (ISO8859-2)
UCS-2LE	8859-3 (ISO8859-3)
UCS-2LE	8859-4 (ISO8859-4)
UCS-2LE	8859-5 (ISO8859-5)
UCS-2LE	8859-6 (ISO8859-6)
UCS-2LE	8859-7 (ISO8859-7)
UCS-2LE	8859-8 (ISO8859-8)
UCS-2LE	8859-9 (ISO8859-9)
UCS-2LE	8859-10 (ISO8859-10)
UCS-2LE	8859-13 (ISO8859-13)
UCS-2LE	8859-14 (ISO8859-14)
UCS-2LE	8859-15 (ISO8859-15)
UCS-2LE	8859-16 (ISO8859-16)
UCS-2LE	KOI8-R
UCS-2LE	KOI8-U
UCS-2LE	UCS-4
UCS-2LE	UCS-4BE
UCS-2LE	UCS-4LE
UCS-2LE	UTF-8
UCS-2LE	UTF-32

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UCS-2LE	UTF-32BE
UCS-2LE	UTF-32LE
UCS-4	646
UCS-4	8859-1 (ISO8859-1)
UCS-4	8859-2 (ISO8859-2)
UCS-4	8859-3 (ISO8859-3)
UCS-4	8859-4 (ISO8859-4)
UCS-4	8859-5 (ISO8859-5)
UCS-4	8859-6 (ISO8859-6)
UCS-4	8859-7 (ISO8859-7)
UCS-4	8859-8 (SO 8859-8)
UCS-4	8859-9 (ISO8859-9)
UCS-4	8859-10 (ISO8859-10)
UCS-4	8859-13 (ISO8859-13)
UCS-4	8859-14 (ISO8859-14)
UCS-4	8859-15 (ISO8859-15)
UCS-4	8859-16 (ISO8859-16)
UCS-4	KOI8-R
UCS-4	KOI8-U
UCS-4	UCS-2
UCS-4	UCS-2BE
UCS-4	UCS-2LE
UCS-4	UTF-7
UCS-4	UTF-8
UCS-4	UCS-16
UCS-4	UCS-16BE
UCS-4	UCS-16LE
UCS-4	UTF-32
UCS-4	UCS-32BE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UCS-4	UCS-32LE
UCS-4BE	646
UCS-4BE	8859-1 (ISO8859-1)
UCS-4BE	8859-2 (ISO8859-2)
UCS-4BE	8859-3 (ISO8859-3)
UCS-4BE	8859-4 (ISO8859-4)
UCS-4BE	8859-5 (ISO8859-5)
UCS-4BE	8859-6 (ISO8859-6)
UCS-4BE	8859-7 (ISO8859-7)
UCS-4BE	8859-8 (SO 8859-8)
UCS-4BE	8859-9 (ISO8859-9)
UCS-4BE	8859-10 (ISO8859-10)
UCS-4BE	8859-13 (ISO8859-13)
UCS-4BE	8859-14 (ISO8859-14)
UCS-4BE	8859-15 (ISO8859-15)
UCS-4BE	8859-16 (ISO8859-16)
UCS-4BE	KOI8-R
UCS-4BE	KOI8-U
UCS-4BE	UCS-2
UCS-4BE	UCS-2BE
UCS-4BE	UCS-2LE
UCS-4BE	UCS-8
UCS-4BE	UCS-16
UCS-4BE	UCS-16BE
UCS-4BE	UCS-16LE
UCS-4BE	UCS-32
UCS-4BE	UCS-32BE
UCS-4BE	UCS-32LE
UCS-4LE	646 (ISO 646)

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UCS-4LE	8859-1 (ISO8859-1)
UCS-4LE	8859-2 (ISO8859-2)
UCS-4LE	8859-3 (ISO8859-3)
UCS-4LE	8859-4 (ISO8859-4)
UCS-4LE	8859-5 (ISO8859-5)
UCS-4LE	8859-6 (ISO8859-6)
UCS-4LE	8859-7 (ISO8859-7)
UCS-4LE	8859-8 (SO 8859-8)
UCS-4LE	8859-9 (ISO8859-9)
UCS-4LE	8859-10 (ISO8859-10)
UCS-4LE	8859-13 (ISO8859-13)
UCS-4LE	8859-14 (ISO8859-14)
UCS-4LE	8859-15 (ISO8859-15)
UCS-4LE	8859-16 (ISO8859-15)
UCS-4LE	KOI8-R
UCS-4LE	KOI8-U
UCS-4LE	UCS-2
UCS-4LE	UCS-2BE
UCS-4LE	UCS-2LE
UCS-4LE	UTF-16
UCS-4LE	UTF-16BE
UCS-4LE	UTF-16LE
UCS-4LE	UTF-8
UTF-7	UCS-2
UTF-7	UCS-4
UTF-7	UCS-8
UTF-8	646 (ISO 646)
UTF-8	8859-1 (ISO8859-1)
UTF-8	8859-2 (ISO8859-2)

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UTF-8	8859-3 (ISO8859-3)
UTF-8	8859-4 (ISO8859-4)
UTF-8	8859-5 (ISO8859-5)
UTF-8	8859-6 (ISO8859-6)
UTF-8	8859-7 (ISO8859-7)
UTF-8	8859-8 (ISO8859-8)
UTF-8	8859-9 (ISO8859-9)
UTF-8	8859-10 (ISO8859-10)
UTF-8	8859-11 (ISO8859-11)
UTF-8	8859-13 (ISO8859-13)
UTF-8	8859-14 (ISO8859-14)
UTF-8	8859-15 (ISO8859-15)
UTF-8	8859-16 (ISO8859-16)
UTF-8	ACE
UTF-8	ACE-ALLOW-UNASSIGNED
UTF-8	eucJP
UTF-8	gb2312
UTF-8	iso2022
UTF-8	ko_KR-euc
UTF-8	ko_KR-johap
UTF-8	ko_KR-johap92
UTF-8	ko_KR-iso2022-7
UTF-8	zh_TW-euc
UTF-8	zh_TW-iso2022-7
UTF-8	zh_TW-cp937
UTF-8	FujitsuJEF-ascii-code
UTF-8	FujitsuJEF-ascii-face
UTF-8	FujitsuJEF-kana-code
UTF-8	FujitsuJEF-kana-face

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UTF-8	GBK
UTF-8	HitachiKEIS83
UTF-8	HitachiKEIS90
UTF-8	ISO-2022-JP
UTF-8	KOI8-R
UTF-8	KOI8-U
UTF-8	UTF-7
UTF-8	NECJIPS
UTF-8	PCK
UTF-8	UCS-2
UTF-8	UCS-2BE
UTF-8	UCS-2LE
UTF-8	UCS-4
UTF-8	UCS-4BE
UTF-8	UCS-4LE
UTF-8	UTF-7
UTF-8	UTF-8
UTF-8	UTF-16
UTF-8	UTF-16BE
UTF-8	UCS-16LE
UTF-16	646 (ISO 646)
UTF-16	8859-1 (ISO8859-1)
UTF-16	8859-2 (ISO8859-2)
UTF-16	8859-3 (ISO8859-3)
UTF-16	8859-4 (ISO8859-4)
UTF-16	8859-5 (ISO8859-5)
UTF-16	8859-6 (ISO8859-6)
UTF-16	8859-7 (ISO8859-7)
UTF-16	8859-8 (ISO8859-8)

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UTF-16	8859-9 (ISO8859-9)
UTF-16	8859-10 (ISO8859-10)
UTF-16	8859-13 (ISO8859-13)
UTF-16	8859-14 (ISO8859-14)
UTF-16	8859-15 (ISO8859-15)
UTF-16	8859-15 (ISO8859-15)
UTF-16	8859-16 (ISO8859-16)
UTF-16	KOI8-R
UTF-16	KOI8-U
UTF-16	UCS-4
UTF-16	UCS-4BE
UTF-16	UCS-4LE
UTF-16	UTF-8
UTF-16BE	646 (ISO 646)
UTF-16BE	8859-1 (ISO8859-1)
UTF-16BE	8859-2 (ISO8859-2)
UTF-16BE	8859-3 (ISO8859-3)
UTF-16BE	8859-4 (ISO8859-4)
UTF-16BE	8859-5 (ISO8859-5)
UTF-16BE	8859-6 (ISO8859-6)
UTF-16BE	8859-7 (ISO8859-7)
UTF-16BE	8859-8 (ISO8859-8)
UTF-16BE	8859-9 (ISO8859-9)
UTF-16BE	8859-10 (ISO8859-10)
UTF-16BE	8859-13 (ISO8859-13)
UTF-16BE	8859-14 (ISO8859-14)
UTF-16BE	8859-15 (ISO8859-15)
UTF-16BE	8859-16 (ISO8859-16)
UTF-16BE	KOI8-R

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UTF-16BE	KOI8-U
UTF-16BE	UCS-4
UTF-16BE	UCS-4BE
UTF-16BE	UCS-4LE
UTF-16BE	UTF-8
UTF-16LE	646 (ISO 646)
UTF-16LE	8859-1 (ISO8859-1)
UTF-16LE	8859-2 (ISO8859-2)
UTF-16LE	8859-3 (ISO8859-3)
UTF-16LE	8859-4 (ISO8859-4)
UTF-16LE	8859-5 (ISO8859-5)
UTF-16LE	8859-6 (ISO8859-6)
UTF-16LE	8859-7 (ISO8859-7)
UTF-16LE	8859-8 (ISO8859-8)
UTF-16LE	8859-9 (ISO8859-9)
UTF-16LE	8859-10 (ISO8859-10)
UTF-16LE	8859-13 (ISO8859-13)
UTF-16LE	8859-14 (ISO8859-14)
UTF-16LE	8859-15 (ISO8859-15)
UTF-16LE	8859-16 (ISO8859-16)
UTF-16LE	KOI8-R
UTF-16LE	KOI8-U
UTF-16LE	UCS-4
UTF-16LE	UCS-4BE
UTF-16LE	UCS-4LE
UTF-16LE	UTF-8
UTF-32	UTF-8
UTF-32	UCS-2
UTF-32	UCS-2BE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UTF-32	UCS-2LE
UTF-32	UCS-4
UTF-32	UCS-4BE
UTF-32	UCS-4LE
UTF-32	UTF-16
UTF-32	UTF-16LE
UTF-32	UTF-32BE
UTF-32	646 (ISO 646)
UTF-32	ISO8859-1
UTF-32	ISO8859-2
UTF-32	ISO8859-3
UTF-32	ISO8859-4
UTF-32	ISO8859-5
UTF-32	ISO8859-6
UTF-32	ISO8859-7
UTF-32	ISO8859-8
UTF-32	ISO8859-9
UTF-32	ISO8859-10
UTF-32	ISO8859-13
UTF-32	ISO8859-14
UTF-32	ISO8859-15
UTF-32	ISO8859-16
UTF-32	KOI8-R
UTF-32	KOI8-U
UTF-32BE	UTF-8
UTF-32BE	UCS-2
UTF-32BE	UCS-2BE
UTF-32BE	UCS-2LE
UTF-32BE	UCS-4

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UTF-32BE	UCS-4BE
UTF-32BE	UCS-4LE
UTF-32BE	UTF-16
UTF-32BE	UTF-16BE
UTF-32 BE	UTF-16LE
UTF-32BE	646 (ISO 646)
UTF-32BE	ISO8859-1
UTF-32BE	ISO8859-2
UTF-32BE	ISO8859-3
UTF-32BE	ISO8859-4
UTF-32BE	ISO8859-5
UTF-32BE	ISO8859-6
UTF-32BE	ISO8859-7
UTF-32BE	ISO8859-8
UTF-32BE	ISO8859-9
UTF-32BE	ISO8859-10
UTF-32BE	ISO8859-13
UTF-32BE	ISO8859-14
UTF-32BE	ISO8859-15
UTF-32BE	ISO8859-16
UTF-32BE	KOI8-R
UTF-32BE	KOI8-U
UTF-32LE	UTF-8
UTF-32LE	UCS-2
UTF-32LE	UCS-2BE
UTF-32LE	UCS-2LE
UTF-32LE	UCS-4
UTF-32LE	UCS-4BE
UTF-32LE	UCS-4LE

表 A-1 使用できる Unicode 関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UTF32-LE	UTF-16
UTF32-LE	UTF-16BE
UTF-32LE	UTF-16LE
UTF-32LE	646 (ISO 646)
UTF-32LE	ISO8859-1
UTF-32LE	ISO8859-2
UTF-32LE	ISO8859-3
UTF-32LE	ISO8859-4
UTF-32LE	ISO8859-5
UTF-32LE	ISO8859-6
UTF-32LE	ISO8859-7
UTF-32LE	ISO8859-8
UTF-32LE	ISO8859-9
UTF-32LE	ISO8859-10
UTF-32LE	ISO8859-13
UTF-32LE	ISO8859-14
UTF-32LE	ISO8859-15
UTF-32LE	ISO8859-16
UTF-32LE	KOI8-R
UTF-32LE	KOI8-U

注 - UTF-EBCDIC は新しい IBM コードページ名です。最新の Solaris 環境は、UTF-8 と UTF-EBCDIC コードページ間の双方向変換にも対応しています。

最新の Solaris 環境で使用できる Unicode、IBM/Microsoft EBCDIC、PC の iconv コード変換モジュールは、次の表のとおりです。

表 A-2 使用できる Unicode、IBM/Microsoft EBCDIC、PC のコードページ関連 iconv コード変換モジュール

変換元コード (シンボル)	変換先コード (シンボル)
UTF-8	IBM-037
UTF-8	IBM-273
UTF-8	IBM-277
UTF-8	IBM-278
UTF-8	IBM-280
UTF-8	IBM-284
UTF-8	IBM-285
UTF-8	IBM-297
UTF-8	IBM-420
UTF-8	IBM-424
UTF-8	IBM-500
UTF-8	IBM-850
UTF-8	IBM-852
UTF-8	IBM-855
UTF-8	IBM-856
UTF-8	IBM-857
UTF-8	IBM-862
UTF-8	IBM-864
UTF-8	IBM-866
UTF-8	IBM-869
UTF-8	IBM-870
UTF-8	IBM-871
UTF-8	IBM-875
UTF-8	IBM-880
UTF-8	IBM-1025
UTF-8	IBM-1026
UTF-8	IBM-1112
UTF-8	IBM-1122

表 A-2 使用できる Unicode、IBM/Microsoft EBCDIC、PC のコードページ関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UTF-8	IBM-921
UTF-8	IBM-922
UTF-8	IBM-1046
UTF-8	IBM-1140
UTF-8	IBM-1141
UTF-8	IBM-1142
UTF-8	IBM-1143
UTF-8	IBM-1144
UTF-8	IBM-1145
UTF-8	IBM-1146
UTF-8	IBM-1147
UTF-8	IBM-1148
UTF-8	IBM-1149
UTF-8	CP850
UTF-8	CP852
UTF-8	CP855
UTF-8	CP857
UTF-8	CP862
UTF-8	CP864
UTF-8	CP866
UTF-8	CP869
UTF-8	CP874
UTF-8	CP1250
UTF-8	CP1251
UTF-8	CP1252
UTF-8	CP1253
UTF-8	CP1254

表 A-2 使用できる Unicode、IBM/Microsoft EBCDIC、PC のコードページ関連 iconv コード変換モジュール (続き)

変換元コード (シンボル)	変換先コード (シンボル)
UTF-8	CP1255
UTF-8	CP1256
UTF-8	CP1257
UTF-8	CP1258

IBM および Microsoft EBCDIC/PC のコードページから UTF-8 のコード変換に使用できる iconv コード変換モジュールは、次の表のとおりです。

表 A-3 使用できる iconv コード変換モジュール - IBM および Microsoft EBCDIC/PC コードページから UTF-8

UTF-EBCDIC	UTF-8
IBM-037	UTF-8
IBM-273	UTF-8
IBM-277	UTF-8
IBM-278	UTF-8
IBM-280	UTF-8
IBM-284	UTF-8
IBM-285	UTF-8
IBM-297	UTF-8
IBM-420	UTF-8
IBM-424	UTF-8
IBM-500	UTF-8
IBM-850	UTF-8
IBM-852	UTF-8
IBM-855	UTF-8
IBM-856	UTF-8
IBM-857	UTF-8
IBM-862	UTF-8
IBM-864	UTF-8
IBM-866	UTF-8

表 A-3 使用できる iconv コード変換モジュール - IBM および Microsoft EBCDIC/PC コードページから UTF-8 (続き)

UTF-EBCDIC	UTF-8
IBM-869	UTF-8
IBM-870	UTF-8
IBM-871	UTF-8
IBM-875	UTF-8
IBM-880	UTF-8
IBM-921	UTF-8
IBM-922	UTF-8
IBM-1025	UTF-8
IBM-1026	UTF-8
IBM-1046	UTF-8
IBM-1112	UTF-8
IBM-1122	UTF-8
IBM-1140	UTF-8
IBM-1141	UTF-8
IBM-1142	UTF-8
IBM-1143	UTF-8
IBM-1144	UTF-8
IBM-1145	UTF-8
IBM-1146	UTF-8
IBM-1147	UTF-8
IBM-1148	UTF-8
IBM-1149	UTF-8
CP850	UTF-8
CP852	UTF-8
CP855	UTF-8
CP857	UTF-8
CP862	UTF-8
CP864	UTF-8

表 A-3 使用できる iconv コード変換モジュール - IBM および Microsoft EBCDIC/PC コードページから UTF-8 (続き)

UTF-EBCDIC	UTF-8
CP866	UTF-8
CP869	UTF-8
CP874	UTF-8
CP1250	UTF-8
CP1251	UTF-8
CP1252	UTF-8
CP1253	UTF-8
CP1254	UTF-8
CP1255	UTF-8
CP1256	UTF-8
CP1257	UTF-8
CP1258	UTF-8



# 索引

---

## 数字・記号

16ビットUnicodeコードセット, 177

## A

API, 国際化, 47

## B

Bengali キーボード, 88

## C

### CDE

入力方式, 120

ロケールサポート, 118

### Complex Text Layout (CTL), 16, 163-184

Motif, 165-175

Motif ライブラリ, 183-184

XOC リソース, 165

XOM, 164

アーキテクチャ, 164

キーボードによる選択, 182

区別的発音符, 163-184

合字, 163-184

水平タブ, 180-181

テキストの方向, 163-184

テキストのリソース, 182-183

マウスによる選択, 181-182

レイアウトの方向, 176-179

### Complex Text Layout (CTL) (続き)

レンダーテーブルの作成, 179-180

レンディションの作成, 177

レンディションの編集, 178

Compose キー, 38

Compose キーシーケンス

Latin-1, 124

Latin-2, 128

Latin-3, 129

Latin-4, 130

Latin-5, 131

Latin-9, 131

アクセントデッドキー, 132

ギリシャ語, 141

ギリシャ語、3つのキー, 147

ギリシャ語、4つのキー, 149

.cshrc, STREAMS モジュールの設定, 157

ctype マクロ, 46

C ロケール, 28

## D

Devanagari キーボード, 88

DtMail, MIME 文字セット, 158

dtterm, 156

## E

en\_US.UTF-8, サポート, 118

en\_US.UTF-8, フォントセットの定義, 162

## G

genmsg ユーティリティ, 54-55  
GMT との時差, 31  
Gujarati キーボード, 89  
Gurmukhi キーボード, 89

## I

iconv  
コード変換, 57  
日本語文字コード変換, 86  
iconv 変換モジュール  
EBCDIC/PC コードページ, 227-229  
Unicode, 203-224  
ISO Latin-1, 27  
ISO8859, 文字サポート, 119

## K

Kannada キーボード, 90  
Kedmanee キーボード, 114  
拡張, 115

## L

LANG 環境変数, 153  
Latin-1, Compose キーシーケンス, 124  
Latin-2, Compose キーシーケンス, 128  
Latin-3, Compose キーシーケンス, 129  
Latin-4, Compose キーシーケンス, 130  
Latin-5, Compose キーシーケンス, 131  
Latin-9, Compose キーシーケンス, 131  
LC\_ALL, 27  
libc

API, 43-44  
アプリケーションのリンク, 41-58  
コード変換関数, 48  
正規表現, 48  
通貨処理, 50  
日付と時間の処理, 51  
複数バイト処理, 51  
メッセージング関数, 47  
文字照合, 50  
文字の分類および変換, 49  
ロケールの照会, 48

## libc (続き)

ロケールの変更および照会, 48  
ワイド文字および文字処理, 51  
ワイド文字クラス, 48  
ワイド文字入力および出力, 52, 53  
ワイド文字列, 53

## M

Malayalam キーボード, 90  
mbtwoc, 47-54  
Motif  
TextField, 176  
UIL 引数, 175  
XmNalignment, 169, 176, 177  
XmNeditPolicy, 169  
XmNlayoutDirection, 166, 176  
XmNlayoutModifier, 176, 177  
XmNrenditionTag, 169  
XmRendition, 167, 176  
XmStringDirection, 167  
XmStringDirectionCreate, 175  
XmText, 176  
XmTextFieldGetLayoutModifier, 173  
XmTextFieldSetLayoutModifier, 174  
XmTextGetLayoutModifier, 173  
XmTextSetLayoutModifier, 174

## mp

PostScript 変数, 197  
TrueType, 23  
Xprt クライアント, 187  
印刷フィルタ, 185, 186  
mp.conf ファイル, 188

## P

Pattajoti キーボード, 115  
Portable Layout Services (PLS), 188  
POSIX ロケール, 28  
PostScript  
prolog ファイル, 195  
実行時変数, 197  
prolog ファイル, 194

## S

setlocale コマンド, 153  
SPARC キーボード, 73  
strconf コマンド, 156  
STREAMS  
  TTY 環境, 154  
  コード変換, 154  
  モジュールのロード, 155  
stty, ユーティリティ, 156  
Sun Ray, USB Type 6 キーボード, 23

## T

Tamil キーボード, 90  
Teluga キーボード, 91  
TTY 環境, 設定, 154  
TypeOfText, 168

## U

UIL, 175  
Unicode  
  16 進数入力モード, 152  
  概要, 117  
  変換モジュール, 203-224  
UTC, 31  
UTF-8, サポート, 118

## X

X Logical Fonts Description (XLFD), 191  
X Print Xerver (Xprt), mp, 187  
XmText  
  backward-cell(extend), 172  
  delete-left-character(), 172  
  delete-right-character(extend)  
  , 172  
  forward-cell(extend), 173  
  left-character(extend), 170  
  right-character(extend), 171  
  right-word(extend), 171  
XPG4 アプリケーション, 46  
.xpr ファイル, 194  
xterm, 156

## あ

アプリケーション  
  FontSet/XmFontList の定義, 161  
  XPG4, 46  
  システムライブラリとのリンク, 45  
アラビア語  
  入力モード, 139  
  文字サポート, 119  
アラビア語キーボード, 73  
アルファベット, 35

## い

イギリス向けキーボード, 79  
イタリア語キーボード, 75  
印刷フィルタ, 185  
インド語, 入力方式, 87

## え

英語  
  入力モード, 123  
  文字サポート, 119

## お

オランダ (オランダ語) キーボード, 76

## か

カタカナ, 35  
韓国語  
  入力モード, 151  
  ハングル, 36  
  ハンジャ, 36  
  文字サポート, 119  
韓国語キーボード, 76  
漢字, 35, 36  
完全地域対応パッケージ, 日本語, 86  
簡体字中国語  
  入力モード, 151  
  文字セット, 119

## き

### キーボード

- Bengali, 88
- CTL の選択, 182
- Devanagari, 88
- Gujarati, 89
- Gurmukhi, 89
- Kannada, 90
- Kedmanee, 114
- Kedmanee 拡張, 115
- Malayalam, 90
- Pattajoti, 115
- SPARC のキーボード設定の変更, 73
- Tamil, 90
- Teluga, 91
- Type 6, 69
- Type 4、5、5c, 70
- アラビア語, 73, 139
- イギリス向け, 79
- イタリア語, 75
- オランダ (オランダ語), 76
- 韓国語, 76, 151
- ギリシャ語キーボード (UNIX キーボード), 141
- ギリシャ語キーボード (ヨーロッパキーボード), 141
- キリル文字, 73
- キリル文字 (ロシア語), 140
- スイス (ドイツ語), 78
- スイス (フランス語), 78
- スウェーデン語, 77
- スペイン語, 77
- 地域, 68
- チェコ配列への変更, 72
- デンマーク語, 74
- ドイツ語, 75
- トルコ語 F, 79
- トルコ語 Q, 79
- 日本語, 75, 150
- ノルウェー語, 76
- 配列, 73, 88, 114
- 配列の変更, 70
- 繁体字中国語, 78
- フィンランド語, 74
- フランス語, 74
- 米国/UNIX, 80
- 米国向け, 80
- ヘブライ語, 150

### キーボード (続き)

- ベルギー向け, 73
- ポルトガル語, 77
- ギリシャ語
  - 入力モード, 140, 141
  - 文字サポート, 119
- ギリシャ語キーボード (UNIX キーボード), キーボード, 141
- ギリシャ語キーボード (ヨーロッパキーボード), キーボード, 141
- キリル文字, 入力モード, 139
- キリル文字キーボード, 73

## こ

- コードセット, 27
  - 変換, 158
  - 文字サポート, 119
- コードセットの独立性 (CSI)
  - ASCII スラッシュ, 42
  - Extended UNIX Code (EUC), 41-43
  - Java の国際化, 42
  - NULL バイト, 42
  - Shift-JIS コードセット, 41-43
  - コマンド, 42-43
  - 動的にリンクされたアプリケーション, 45
  - ファイルのエンコーディング, 42
  - 複数バイト文字, 44
  - プロセスコードの書式, 44
  - ライブラリ, 43
  - ロケールデータベース, 43-44
- コード変換モジュール, 203-224, 225-227
- 国際化
  - ISO Latin-1, 27
  - 定義, 25-27
- 国際化 API, 47
- 国際化ドメイン名 (IDN), 56
- コマンド, CSI, 42-43

## し

- 時間帯, 31
- 時間の形式, 31
- システムライブラリ, アプリケーションのリンク, 45

## す

スイス(ドイツ語) キーボード, 78  
スイス(フランス語) キーボード, 78  
スウェーデン語キーボード, 77  
スカンジナビア諸国, 文字サポート, 119  
スクリプトの選択, 121  
スペイン語, 文字サポート, 119  
スペイン語キーボード, 77

## せ

静的リンク, 45

## た

タイ語, 36  
キー配列, 24  
入力方式, 114, 116  
文字サポート, 119  
文字シーケンスチェック, 114-116

## 端末

サポート, 157  
設定オプション, 156

## ち

地域対応, 83-116  
構成ファイル, 188  
定義, 25-27  
チェコ語, 文字サポート, 119  
中華人民共和国, 37  
中国語  
bopomofo, 37  
zhuyin, 37  
漢字, 36  
台湾, 37  
中華人民共和国, 36  
入力方式, 25  
ピンイン方式, 37  
香港, 37

## て

デンマーク語キーボード, 74

## と

ドイツ語, 文字サポート, 119  
ドイツ語キーボード, 75  
トルコ語, 文字サポート, 119  
トルコ語 F キーボード, 79  
トルコ語 Q キーボード, 79

## な

夏時間 (DST), 31

## に

西ヨーロッパの言語  
文字サポート, 119  
日本語  
iconv モジュール, 86  
カタカナ, 36  
漢字, 35  
完全地域対応パッケージ, 86  
地域対応, 83-87  
入力方式, 85  
入力モード, 150  
ひらがな, 36  
フォント, 84-85  
文字サポート, 119  
文字セット, 83-84  
ロケール, 83  
日本語キーボード, 75  
入力方式  
ATOK 方式, 85  
インド語, 92  
インド語方式, 87  
タイ語, 116  
日本語, 85  
入力モード  
en\_US.UTF-8 ロケール, 120  
Unicode 16 進数, 152  
アラビア語, 139  
英語, 123  
韓国語, 151  
簡体字中国語, 151  
ギリシャ語, 140, 141  
キリル文字, 139  
日本語, 150  
繁体字中国語, 151

入力モード (続き)  
表検索, 153  
ヘブライ語, 149

の  
ノルウェー語キーボード, 76

は  
バルト諸国, 文字サポート, 119  
ハンガリー語, 文字サポート, 119  
ハングル, 36  
ハンジャ, 36  
繁体字中国語  
入力モード, 151  
文字サポート, 119  
繁体字中国語キーボード, 78

ひ  
日付の形式, 31  
表検索, 入力モード, 153  
ひらがな, 35  
ピンイン, 37  
ヒンディー語  
デーバナーガリー, 37  
文字サポート, 119

ふ  
フィンランド語キーボード, 74  
フォント  
対応付け, 189  
日本語 TrueType, 85  
日本語ビットマップ, 84  
別名の定義, 189  
フォントセットの定義, 162  
複数バイト文字, 44  
フランス語キーボード, 74

へ  
米国/UNIX キーボード, 80  
米国向けキーボード, 80  
ページサイズ, 一般的なサイズ, 38  
ヘブライ語  
イディッシュ語, 37  
入力モード, 149  
文字サポート, 119  
ベルギー向けキーボード, 73  
変換  
Unicode iconv モジュール, 203-224  
ユーザー定義コードセット, 55

ほ  
ポーランド語, 文字サポート, 119  
ポルトガル語キーボード, 77

ま  
マッピング, インド語スクリプトの英語と表音  
が同等, 92  
マップ  
Bengali 文字, 92  
Gujarati 文字, 94  
Gurmukhi 文字, 97  
Hindi 文字, 100  
Kannada 文字, 102  
Malayalam 文字, 105  
Tamil 文字, 108  
Telugu 文字, 111  
マルチバイト, 変換, 47-54

め  
メッセージカタログ, 54-55

も  
文字  
サポート, 119  
整形, 163-184  
複数バイト, 44  
変換, 47-54

文字エンコーディング  
UTF-16, 118  
UTF-32, 118  
UTF-8, 118  
文字コード化, Unicode, 117

## ゆ

ユーティリティ  
genmsg, 55  
genmsg, 54-55  
iconv, 57  
locale, 153  
stty, 156

## ら

ライブラリ, (CSI), 43

## り

リンク, アプリケーション, 45

## れ

レイアウト動作, 168  
連続表音入力方式, インド語スクリプト, 113

## ろ

ロケール  
C, 28  
Compose キー, 67  
POSIX, 28  
Solaris, 83-116  
アジア, 60  
オーストラリア, 62  
カテゴリ, 30  
環境変数, 153  
完全, 29, 60  
キーボードの相違点, 38  
北アフリカ, 64  
北アメリカ, 64

## ロケール (続き)

北ヨーロッパ, 65  
語の区切り, 35  
時間の形式, 31  
数値の形式, 32  
ソートの順序, 35  
中央アメリカ, 62  
中央ヨーロッパ, 62  
中東, 64  
通貨, 26  
通貨の軽視, 33  
定義, 27  
西ヨーロッパ, 66  
日本語, 83  
東ヨーロッパ, 63  
日付の形式, 31  
部分, 29, 59  
文化的慣習, 29  
ページサイズ, 38  
南アメリカ, 65  
南ヨーロッパ, 66  
文字セット, 35  
ロケールの環境変数, 153  
ロシア語, 文字サポート, 119

## わ

ワイド文字, サポート, 47-54

