

Tru64 UNIX

Using International Software

Part Number: AA-RSFEA-TE

September 2002

Product Version: Tru64 UNIX Version 5.1B or higher

This manual describes how to use internationalized software on the Tru64 UNIX operating system. It is a companion to *Writing Software for the International Market*, which describes how to create international software with the tools provided on the Tru64 UNIX operating system.

© 2002 Hewlett-Packard Company

Microsoft®, Windows®, Windows NT®, and MS-DOS® are trademarks of Microsoft Corporation in the U.S. and/or other countries. Motif®, OSF/1®, UNIX®, X/Open®, and The Open Group™ are trademarks of The Open Group in the U.S. and/or other countries. All other product names mentioned herein may be the trademarks of their respective companies.

Confidential computer software. Valid license from Compaq Computer Corporation, a wholly owned subsidiary of Hewlett-Packard Company, required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

None of Compaq, HP, or any of their subsidiaries shall be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for HP or Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Contents

About This Manual

1 Working in a Multilanguage Environment

1.1	Overview of Using Internationalized Software	1-1
1.2	Configuring International Software	1-2
1.3	Setting Locale and Language	1-3
1.4	Selecting Keyboard Type	1-5
1.4.1	Determining Keyboard Layout	1-6
1.4.2	Entering the Euro Currency Symbol	1-6
1.5	Defining the Search Path for Specialized Components	1-11
1.6	Supporting User-Defined Characters	1-11
1.7	Using Printer Interface Features That Support Local Languages	1-12
1.7.1	Generic Internationalized Print Filters	1-12
1.7.1.1	pccof Print Filter	1-12
1.7.1.2	wpsof Print Filter	1-13
1.7.2	Print Filters for Specific Local Language Printers	1-14
1.7.3	Support for Local Language Printers in <i>/etc/printcap</i>	1-15
1.7.4	Enhancements to Printer Configuration Software	1-18
1.7.5	Print Commands and the Printer Daemon	1-21
1.7.6	Choosing PostScript Fonts for Different Locales	1-21
1.8	Using Mail in a Multilanguage Environment	1-26
1.8.1	The <i>sendmail</i> Utility	1-26
1.8.2	The <i>mailx</i> Command and MH Commands	1-26
1.8.3	The <i>comsat</i> Server	1-28
1.9	Displaying Reference Pages in Languages Other Than English	1-28
1.10	Converting Data Files from One Codeset to Another	1-29
1.11	Miscellaneous Base System Commands	1-30

2 Using Asian Input Methods and Terminal Drivers

2.1	Determining Input Method	2-1
2.2	Determining the Keyboard Input Mode Switch State	2-5
2.3	Using Terminal Interface Features for Asian Languages	2-6
2.3.1	Converting Between Application and Terminal Codesets ..	2-9

2.3.2	Command Line Editing That Supports Multibyte Characters	2-9
2.3.3	Kana-Kanji Conversion: Customization of Japanese Input Options	2-11
2.3.4	Thai Terminal Support	2-15

3 Using Asian Language Support Enhancements for Motif Applications

3.1	Tuning the X Server for Ideographic Languages	3-1
3.2	Setting Fonts for Display of Local Languages	3-4
3.2.1	Accessing Local Language Fonts for Remote Displays	3-4
3.3	Customizing a Terminal Emulation Window for Asian Languages	3-5

A Bitmap and Monospaced Fonts for Locales

B Phrase Utility

B.1	Enabling the SIM Service	B-3
B.2	Creating and Maintaining a Phrase Database	B-3
B.3	Using a Phrase Database	B-7
B.3.1	Phrase Input Supported Through the SIM Service	B-7
B.3.2	Phrase Input from the Input Options Application	B-8

Index

Examples

1-1	Setting Up a Local Language Printer with lprsetup	1-18
-----	---	------

Figures

B-1	User Interface Screen of the Phrase Utility	B-4
-----	---	-----

Tables

1-1	Locale and Key Combination Summary	1-8
1-2	Symbols in /etc/printcap File for Local Language Printers	1-15
2-1	The stty Command Options for Controlling Terminal Line Discipline	2-7

2-2	The stty Options to Explicitly Set Application and Terminal Code	2-9
2-3	The stty Options Used with atty to Enable and Disable History Mode	2-10
2-4	Command Line Editing in History Mode	2-11
2-5	The stty Options to Enable and Customize Japanese Input ...	2-12
2-6	The stty Options for Thai Terminals	2-15
3-1	X Server Options for Tuning the Font-Cache Mechanism	3-3
A-1	Bitmap Fonts for Asian Locales	A-1
A-2	Bitmap Fonts for *.ISO8859-1 Locales	A-3
A-3	Bitmap Fonts for *.ISO8859-2 Locales	A-5
A-4	Bitmap Fonts for *.ISO8859-4 Locales	A-8
A-5	Bitmap Fonts for *.ISO8859-5 Locales	A-10
A-6	Bitmap Fonts for *.ISO8859-7 Locales	A-12
A-7	Bitmap Fonts for *.ISO8859-8 Locales	A-14
A-8	Bitmap Fonts for *.ISO8859-9 Locales	A-16
A-9	Bitmap Fonts for *.ISO8859-15 Locales	A-18
A-10	Bitmap Fonts for *.UTF-8 Locales	A-21
B-1	Phrase Input Definitions	B-2
B-2	The stty Options Used for the SIM Service	B-3

About This Manual

The HP Tru64 UNIX operating system provides you with a number of internationalization tools that allow you to use the system productively in a non-English language environment. These tools allow you to perform the following tasks:

- Change the local environment and set the appropriate input method
- Set keyboards, terminal drivers, and printer controls to the appropriate language
- Use mail, view reference pages, and enter base system commands in the language of choice
- Use input methods for Asian language conversion
- Use Motif applications in non-English language environments and set fonts for local languages

This manual describes these tasks. It also contains appendices describing bitmap fonts for different locales and how to set up and use the system Phrase Utility.

Audience

This manual was extracted from the Tru64 UNIX *Writing Software for the International Market* manual. The *Writing Software for the International Market* manual is directed at programmers of internationalized software.

This manual is intended for Tru64 UNIX users and system administrators working in a multilanguage or non-English international environment. Translators who translate the messages displayed by internationalized programs will also find this manual useful. The manual assumes that you are familiar with the operating system in its default English-language environment. Where appropriate, the manual makes note of the requirement for superuser privileges.

Organization

This manual is organized as follows:

<i>Chapter 1</i>	Describes the use of operating system features in a non-English environment. The descriptions include establishing a locale, setting a keyboard map, defining paths, using printer interface features, and using mail and reference pages.
<i>Chapter 2</i>	Discusses setting input methods and keyboard input modes for Asian languages.
<i>Chapter 3</i>	Describes how to use international support features in a CDE Motif environment.
<i>Appendix A</i>	Presents tables describing the characteristics of the bitmap fonts for locales installed with the operating system.
<i>Appendix B</i>	Describes the Phrase Utility and creation of phrase input used with Chinese and Korean input methods.

Related Documentation

The *Writing Software for the International Market* manual is the companion document to this manual. It describes operating system internationalization features from a programmer's perspective.

The Tru64 UNIX *Documentation Overview* manual provides information on all of the documentation provided with the operating system.

The Tru64 UNIX documentation is available on the World Wide Web at the following URL:

<http://www.tru64unix.compaq.com/docs/>

Programming for the World: A Guide to Internationalization (by Sandra Martin O'Donnell and published by Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1994) provides information about cultural and linguistic requirements around the world and the changes needed in computer systems to handle those requirements.

Reader's Comments

HP welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: readers_comment@zk3.dec.com

A Reader's Comment form is located on your system in the following location:

```
/usr/doc/readers_comment.txt
```

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

The following conventions are used in this manual:

<code>%</code>	A percent sign represents the C shell system prompt.
<code>\$</code>	A dollar sign represents the system prompt for the Bourne and Korn shells.
<code>#</code>	A number sign represents the superuser prompt.
<code>% cat</code>	Boldface type in interactive examples indicates typed user input.
<code><i>file</i></code>	Italic (slanted) type indicates variable values, placeholders, and function argument names.
<code>...</code>	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
<code>:</code>	A vertical ellipsis indicates that a portion of an example that would normally be present is not included.

<code>cat(1)</code>	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, <code>cat(1)</code> indicates that you can find information on the <code>cat</code> command in Section 1 of the reference pages.
<code>Ctrl/x</code>	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash, for example, <code>Ctrl/c</code> .
<code>Alt x</code>	Multiple key or mouse button names separated by spaces indicate that you press and release each in sequence, for example, <code>Alt Space</code> .

Working in a Multilanguage Environment

By default, the Tru64 UNIX operating system is installed with support for United States English. However, the system administrator can choose to install one or more Worldwide Language Support (WLS) subsets, which provide the tools and features that allow you to work in languages other than English. Language subsets can also be added after the base operating system installation. WLS installation and the language subsets available for installation are described in the operating system *Installation Guide*.

This chapter explains how to perform the various setup tasks and use the software features of language environments other than English. The chapter assumes that you are familiar with the operating system in its default English-language environment.

1.1 Overview of Using Internationalized Software

To enable input and display in any language other than English, you must always set the locale in which your process runs. Locales for particular languages are installed as WLS subsets, and you can use the `locale -a` command to display the available locales. See Section 1.3 for information on locales and how to set or change locale settings.

If you have superuser privileges, you can use the Configure International Software utility from the SysMan Menu to set the default login locale and perform many other internationalization tasks. See Section 1.2 or the Configure International Software online help for information on using this utility.

Depending on the language, you may need to perform tasks in addition to setting a locale. This chapter describes how to perform the following tasks:

- Select keyboard type (Section 1.4)
- Define search paths for specialized data and executable files that are language specific (Section 1.5)
- Apply printer-control characters, filters, and fonts that are appropriate for local language printers (Section 1.7)
- Mail text in languages other than English (Section 1.8)
- Display reference pages (Section 1.9)

- Convert data files from one codeset to another (Section 1.10)
- Display, edit, and print text in languages other than English (Section 1.11)

This chapter discusses these topics as they apply to particular languages or groups of languages. For complete information about using the internationalization features of applications that run in the Common Desktop Environment (CDE), see Chapter 3 and the Tru64 UNIX *CDE Companion* manual.

1.2 Configuring International Software

This chapter describes how to set locales, keyboard mappings, and other aspects of internationalization support using system utilities and commands. However, if you are a system manager or administrator with superuser privileges, you can use the Configure International Software utility to configure Worldwide Language Support on your system.

The Configure International Software utility is menu-oriented function available from the SysMan Menu under the Software option. As superuser, a system manager or administrator can use Configure International Software to perform the following tasks:

- Configure access to Worldwide Language Support tools and libraries for individual accounts or system-wide.
- Configure Asian terminal driver support and merge that configuration into the system configuration file (`/usr/sys/conf/`). Using this task, the system administrator can do the following:
 - Activate Asian codeset options, including traditional Chinese Big-5, Telecode for Taiwan, simplified Chinese to traditional Chinese conversion, UTF-8, and Thai language support.
 - Add UNIX Terminal Extension (UTX) support options to the Asian terminal driver. The UTX support options include on-demand font loading (ODL), Katakana conversion, and Software Phrase Input Method (SIM).
 - Define the Pseudo Terminal Driver protocol.
 - Establish dynamic or static linking of the Asian terminal driver to the kernel.
 - Specify the number of UTX pseudodevices that will be created on the system.
 - Rebuild the kernel.
- Configure Wnn, the character-cell input method for Japanese.

- Remove installed country support subsets (locales). If you do not have superuser privilege, you can view, but not delete, installed subsets.
- Remove installed fonts. If you do not have superuser privilege, you can view, but not delete, installed fonts.
- Establish a default login language, switch between dense code and Unicode locales, and choose an input method for a locale that supports multiple input methods.
- View installed keyboard map files. You do not have to be superuser to view installed keymaps.

1.3 Setting Locale and Language

Locales are the method whereby the operating system implements localization. A locale establishes information within a computer system that is specific to each supported language, cultural data, and coded character set (codeset) combination. A locale provides information on the following:

- Repertoire of available characters
- Language-specific sorting rules
- Language-specific rules and symbols for monetary and numeric data, date, and time
- Path for translated message files, application resource files, help files, or some combination of these

To view the locales installed on your system, use the command `locale -a` or use the Manage Locales option of the Configure International Software utility.

See `l10n_intro(5)` for information on the languages and codesets that the operating system supports with locales. See `locale(4)` for information on the contents of a locale. This section describes the two types of locales (dense code and Unicode) that Tru64 UNIX offers and how you establish a locale on the operating system.

When Worldwide Language Support is installed on your system, two types of locales are installed for localization support: Unicode locales and dense code locales. Unicode locales conform to Unicode and ISO/IEC 10646 standards and use UTF-32 as the wide-character encoding. Unicode locales whose names end in UTF-8 use file and internal processing code defined in the standards. Other, non-UTF-8 Unicode locales use traditional UNIX and proprietary codesets for the file code and use UTF-32 for internal process code. A subset of these locales have a `@ucs4` modifier; they are provided for backward compatibility and are the same as the locales without `@ucs4`. You cannot select `@ucs4` locales from the CDE login menu; you must specify the locale name in the LANG environment variable.

Dense code locales use dense code for wide-character encoding to minimize table size.

The distinction between dense code and Unicode locales is of interest to programmers and is described in the *Writing Software for the International Market* manual. For users of internationalized software on Tru64 UNIX, dense code locales are functionally equivalent to Unicode locales and a Unicode locale exists for each dense code locale. However, not all Unicode locales have a dense code version.

The Unicode locales are installed in `/usr/i18n/lib/nls/ucslc/`. Dense code locales are installed in `/usr/i18n/lib/nls/loc`. The active default is determined by the symbolic link, `/usr/i18n/lib/nls/dloc`. If you are superuser, you can switch between Unicode and dense code locales by changing the setting of the symbolic link, as described in `l10n_intro(5)` or you can use the Configure International Software utility from the SysMan Menu. See the online help for Configure International Software for more information.

To set a locale for system use, define the `LANG` environment variable as one of the installed locales. For example, under the C shell:

```
% setenv LANG en_US.ISO8859-1
```

This command sets the user environment to the values defined for United States English using the ISO8859-1 codeset. If a locale is not installed, internationalized applications assume the POSIX (C) locale, which supports only English.

See the discussion of internationalization in the *System Administration* manual and in the *Command and Shell User's Guide* for more detailed information on using locales and defining the associated variables for system and user setup. See `i18n_intro(5)` for a discussion of locale variables such as `LANG`.

For graphical applications, you need to select a language to take advantage of the text translations and local language features available with the Common Desktop Environment (CDE) and other Motif applications. For Asian languages, the correct language selection is particularly important because it enables the following features:

- Support for the appropriate input method in these applications
- Entry of file names and other parameters that use ideographic characters
- Cursor positioning on correct character and word boundaries
- Line wrapping at correct word boundaries

See the *CDE Companion* manual for general information about setting language in CDE.

CDE assumes that all applications that run during a session operate in the language that was set at the start of the session. On Tru64 UNIX systems, you can work around this restriction with the following actions:

1. In a `dtterm` window, set the `LANG` or `LC_ALL` environment variable to the locale in which you want to run the new application. For example:

```
% setenv LANG ko_KR.deckorean
```

2. If the setting is for a Japanese, Chinese, or Korean locale, use the system command line to start the appropriate input method server before invoking the application. For example:

```
% /usr/bin/x11/dxhangulim &
```

See Section 2.1 for information about Asian input method servers.

3. In the same window as step 1, use the system command line to invoke the application you want to run in the new locale. For example:

```
% /usr/dt/bin/dtterm &
```

4. If you need to change your keyboard setting to work in the new locale, do so before starting to work in the new application's window. See Section 1.4 for information about setting keyboard type.

1.4 Selecting Keyboard Type

For English language input, a standard keyboard provides a sufficient number of keys (combined with shift states) to enter all uppercase and lowercase letters, numerals, and punctuation marks. For many other languages, the default keyboard does not provide enough keys and shift states to enter all characters.

Terminal users must use a localized keyboard or, if their keyboard includes a Compose key, use Compose-key sequences to enter non-English language characters from single-byte codesets. Some terminals also provide software emulation of a number of keyboard layouts for languages that are based on single-byte codesets. The user manual for each terminal explains how you can use its keyboard to enter non-English characters. Entry of multibyte characters in Asian languages requires special terminal hardware.

If the appropriate support files are installed on the system, workstation users can set the keyboard type to be appropriate for languages for which standard keyboard types exist. You must set the keyboard type for Western and Eastern European languages, Japanese, Thai, and Hebrew. However, you are not required to set the keyboard type for Chinese and Korean languages.

In CDE, use Keyboard Options (one of the desktop applications) to change your keyboard type. See the *CDE Companion* manual for more information about changing keyboard type.

From the system command line, use the `dxkeyboard` command to invoke Keyboard Options to choose a keyboard map and change the keyboard type.

Unlike the language setting, the keyboard setting is a global attribute that applies to all windows. Therefore, if you are working in windows that were created with different language settings, you may need to change the keyboard setting as you move from one window to another.

Keep in mind that no matter what setting you make using CDE applications, that setting does not change the setting that applies when you log on to the system. The keyboard setting when you log on to the system is always the system-default keyboard. See `keyboard(5)` for information about changing the system-default keyboard.

1.4.1 Determining Keyboard Layout

You can use an `xkbprint` command to access a keyboard layout for your current keyboard setting. For example, the following command accesses the layout and creates a PostScript file that you can print:

```
% /usr/bin/X11/xkbprint -label symbols -o mykeyboard.ps :0
```

See `xkbprint(1X)` for more information about the `xkbprint` command.

If you change your keyboard from the one whose characters are printed on the hardware keys, you need to know how characters are mapped to keys and whether any characters must be entered by using a mode-switch key or key sequence. For some languages, such as Czech, up to four different characters can be mapped to the same key. In such cases, you use the key defined as the mode switch to toggle among different sets of characters mapped to the same key.

You can use the `dxkeycaps` command to display and edit keyboard mappings of the keyboard attached to your workstation. The display shows the keyboard, with keycaps drawn according to the current server keymap. Using the mouse, you can bring up a menu of options, including the option to change the key symbol generated by a particular key. See `dxkeycaps(1X)` for more information on command options.

Mode switching is a character entry mechanism that is different from Compose sequences. A particular keyboard setting may support Compose sequences (which require one key to be defined as a multikey), mode switching (which requires at least one key to be defined as a mode-switch key), both, or neither of these input mechanisms.

1.4.2 Entering the Euro Currency Symbol

In 2002, the euro currency became the basic monetary unit in the European countries belonging to the Economic and Monetary Union (EMU).

To enter, display, and print the euro symbol, Worldwide Language Support (WLS) must be installed on your system, and you must perform the following steps:

1. Configure the system with supporting locales, keyboard mappings, and fonts.
2. Use the correct key sequences, codeset converters, and print filters.

This section describes these steps and provides examples of setting locales and selecting keyboard types.

For more information on using the euro currency symbol, see the Tru64 UNIX Best Practices Web page.

Support for the input, display, and printing of any symbol on the operating system requires a codeset that includes that symbol and a font set that can display the symbol. A keyboard mapping that associates keystrokes with the symbol is also useful, although Compose key and cut-and-paste alternatives also exist. The requirements of codeset, font set, and entry method apply whether the symbol is an English language letter, a Chinese language character, or the euro monetary symbol.

The Unicode (UTF-8) and ISO/IEC 8859-15 (Latin-9) codesets include the euro symbol. With WLS installed, the operating system provides these codesets by means of country-specific locales. The operating system also provides the keyboard mappings specific to each country and the Xfont library, which enables display of the euro symbol.

To enter and display the euro currency symbol, perform the following steps:

1. Run under a locale that supports the euro currency symbol. Table 1-1 lists the locales that support the euro symbol. To start one of these locales on your system execute the following steps:
 - a. From the Options Menu of the CDE login screen, choose Language.
 - b. From the Language Options Menu, choose a locale.
2. Choose a keyboard map that is appropriate for the selected euro-enabled locale and for the keyboard type you are using. To choose a keyboard map execute the following steps:
 - a. Enter `/usr/dt/bin/dxkeyboard` at the command line to display the `dxkeyboard` Menu.

Alternatively, you can select Keyboard Options from the Desktop_Apps Menu of the CDE Applications Manager to display the `dxkeyboard` Menu.

- b. From the `dxkeyboard` Menu, choose the keyboard map that matches the locale and keyboard type you are using. The keyboard type can usually be found on the underside of the keyboard. The reference page for the language you are using (for example, `Italian(5)`) describes associated keyboard types and maps for specific locales.
3. Enter the euro currency symbol using the key combination described for your current locale. If your keyboard supports a Compose sequence, press the Compose key followed by C and an equal sign (=) to generate the euro symbol. (The appropriate Compose sequence keys for the euro symbol are described for various locales in `euro(5)`.)

The Configure International Software utility available from the SysMan Menu provides system managers and administrators with an alternative way to manage locales and keymaps on the system. You must be superuser to use Configure International Software.

Table 1–1 is organized by country and lists the locales that support the euro currency symbol and the associated PC-style and VT-style key combination that will generate a euro symbol. The key combinations in this table are supported by `xkb-format` keymaps, which are the default in CDE.

Table 1–1: Locale and Key Combination Summary

Country	Locale	Euro Symbol Input	
		VT-Style Keyboard Combination	PC-Style Keyboard Combination
Catalan (Spain)	<code>ca_ES.UTF-8</code> <code>ca_ES.ISO8859-15</code>	Left Compose/E	Right Alt/E
Chinese – PRC (simplified)	<code>zh_CN.UTF-8</code> (simplified) <code>zh_HK.UTF-8</code> (traditional, Hong Kong) <code>zh_TW.UTF-8</code> (traditional, Taiwan)	There is no Chinese keyboard combination. Use the Qu-Wei Input Method to enter the Unicode value for euro (U+20AC), as described in the Tru64 UNIX <i>Technical Reference for Using Chinese Features</i> online manual.	
Danish (Denmark)	<code>da_DK.UTF-8</code> <code>da_DK.ISO8859-15</code>	Left Compose/E	Right Alt/E
Dutch (Netherlands)	<code>n1_NL.UTF-8</code> <code>n1_NL.ISO8859-15</code>	Left Compose/E	Right Alt/E
Dutch/Flemish (Belgium)	<code>n1_BE.UTF-8</code> <code>n1_BE.ISO8859-15</code>	Left Compose/E	Right Alt/E

Table 1–1: Locale and Key Combination Summary (cont.)

Country	Locale	Euro Symbol Input	
		VT-Style Keyboard Combination	PC-Style Keyboard Combination
English (United Kingdom and Irish Republic)	en_GB.UTF-8	Left	Right Alt/4
	en_GB.ISO8859-15	Compose/4	
English (United States)	en_US.UTF-8	Left	Right Alt/E
	en_US.ISO8859-15	Compose/E	
Finnish (Finland)	fi_FI.UTF-8	Left	Right Alt/E
	fi_FI.ISO8859-15	Compose/E	
French (France)	fr_FR.UTF-8	Left	Right Alt/E
	fr_FR.ISO8859-15	Compose/E	
French (Belgium)	fr_BE.UTF-8	Left	Right Alt/E
	fr_BE.ISO8859-15	Compose/E	
French (Canada)	fr_CA.UTF-8	Left	Right Alt/E
	fr_CA.ISO8859-15	Compose/E	
French (Switzerland)	fr_CH.UTF-8	Left	Right Alt/E
	fr_CH.ISO8859-15	Compose/E	
German (Germany)	de_DE.UTF-8	Left	Right Alt/E
	de_DE.ISO8859-15	Compose/E	
German (Switzerland)	de_CH.UTF-8	Left	Right Alt/E
	de_CH.ISO8859-15	Compose/E	
Icelandic (Iceland)	is_IS.UTF-8	Left	Left
	is_IS.ISO8859-15	Compose/E	Compose/E
Italian (Italy)	it_IT.UTF-8	Left	Right Alt/E
	it_IT.ISO8859-15	Compose/E	
Japanese (Japan)	ja_JP.UTF-8	There is no Japanese Compose table or keymap support for the euro symbol. To enter the euro symbol, use the vi or dtpad editor to cut the symbol from a supporting application and paste it to the target application under the ja_JP.UTF-8 locale.	

Table 1–1: Locale and Key Combination Summary (cont.)

Country	Locale	Euro Symbol Input	
		VT-Style Keyboard Combination	PC-Style Keyboard Combination
Korean (Korea)	ko_KO.UTF-8	There is no Korean Compose table or keymap support for the euro symbol. To enter the euro symbol, use the vi or dtpad editor to cut the symbol from a supporting application and paste it to the target application under the ko_KO.UTF-8 locale.	
Norwegian (Norway)	no_NO.UTF-8 no_NO.ISO8859-15	Left Compose/E	Right Alt/E
Portuguese (Portugal)	pt_PT.UTF-8 pt_PT.ISO8859-15	None	Right Alt/E
Spanish (Spain)	es_ES.UTF-8 es_ES.ISO8859-15	Left Compose/E	Right Alt/E
Swedish (Sweden)	sv_SE.UTF-8 sv_SE.ISO8859-15	Left Compose/E	Right Alt/E

The Alternate function key, described as the Alt key in this table, is also described as the Gr key on some keyboards. (In both cases, the key is on the right side of the keyboard.) For more information on keyboard mappings and keyboards, see `keyboard(5)` for your version of the operating system (<http://www.tru64unix.compaq.com/docs/>).

When you install WLS and languages that support the euro symbol, you receive text and PostScript print filters that are sensitive to system locale settings and that provide fonts containing the euro currency symbol. For example, the generic PostScript print filter (`wpssof`) supports UTF-8 and ISO 8859-15 formats.

No additional user action is required for euro symbol printing support.

The operating system provides two locales (`en_EU.UTF-8@euro` and `en_US.UTF-8@euro`) that specifically assign the euro symbol in the `LC_MONETARY` section of the locale. These locales supplement the UTF-8 and ISO8859-15 locales that define the currency symbol as the euro, as described in `euro(5)`. Because setting `LC_MONETARY` overrides the environment variable `LANG`, you can set `LANG` to a locale that does not support the euro symbol and set `LC_MONETARY` to `en_EU.UTF-8@euro` to obtain euro symbol

support. For more information, see `euro(5)`. Also, see `i18n_intro(5)` for more information on locale-related environment variables.

1.5 Defining the Search Path for Specialized Components

European languages are supported by data and executable files installed at system-default locations. Asian language support for some commands and programming libraries requires files that are subordinate to the `/usr/i18n` directory. These files supplement or replace files in system-default locations.

When you install one or more of the Asian language subsets, the installation procedure makes the following adjustments to variable settings on a system-wide basis:

- `I18NPATH`

The `I18NPATH` variable defines the location of files that provide Asian language support and that are not in system default locations. This variable is set to the following:

```
/usr/i18n
```

Your system administrator can choose to install files for Asian language support at a location different from `/usr/i18n`; however, the `/usr/i18n` directory must contain a link to the other location.

- `PATH`

The `PATH` variable points to the location of commands and is set to the following:

```
$I18NPATH/usr/bin:$PATH
```

The `/etc/i18n_profile` file includes the `PATH` and `I18NPATH` variable assignments on a system-wide basis for Bourne and Korn shell users. For C shell users, the installation process includes the `/etc/i18n_login` file and the `/etc/csh.login` file to correctly set search paths for Hebrew and Asian languages. Unless specifically noted in descriptions of particular commands or utilities, individual users do not need to change process-specific search paths to find localized binaries and utilities.

1.6 Supporting User-Defined Characters

The national character sets for Japan, Taiwan, and China do not include some of the characters that can appear in Asian place and personal names. Such characters are defined by users and reside in site-specific databases. These databases are called user-defined character (UDC) databases. When you define ideographic characters, you must also define font glyphs, collating files, and other support files for the characters.

The *Writing Software for the International Market* manual provides details on how you set up and use UDC databases and how to edit `cp_dirs`, the UDC database configuration file.

1.7 Using Printer Interface Features That Support Local Languages

When you install the operating system and include language variant subsets, your printing subsystem is enhanced with the following features:

- Two generic internationalized print filters, `pcfof` and `wpsof`, that work with HP and third-party printers
- A set of print filters that support escape sequences used by local language printers
- Entries in the `/etc/printcap` file to support printer code conversion and on-demand loading of font files
- An `lprsetup` command that lets you add entries for local language printers to the `/etc/printcap` file
- `lp`, `lpr`, `lpc`, `lpq`, `lprm`, and `lpstat` commands that support additional options for printing and printer control
- PostScript outline or TrueType fonts that can be used by the `wpsof` filter and other software

The following sections discuss these features.

1.7.1 Generic Internationalized Print Filters

The `pcfof` and `wpsof` print filters enable use of HP printers, particularly those for which no other printer-specific solution is described in this chapter. You also need to use these filters if your printer is from another vendor. Both of these filters rely on a printer customization file (`.pcf` file) to supply certain device-specific information. Operating system software includes a basic set of `.pcf` files. System administrators can add more `.pcf` files to describe the capabilities of additional printers used at your site.

1.7.1.1 `pcfof` Print Filter

The `pcfof` filter handles both PostScript printers and text printers, such as the HP PCL printer. For PostScript files, the filter requires that the appropriate local language PostScript fonts be available on the printer. This restriction limits the filter's usefulness on many printers, particularly for printing PostScript files that require Japanese fonts. This filter can be set up to do codeset conversion when the printer locale differs from the one required for a text file print job. The filter also has `.pcf` files that are

appropriate to use for a number of third-party text printers. See `pcfof(8)` and the *System Administration* manual for details on using this print filter.

1.7.1.2 `wpsof` Print Filter

The `wpsof` filter is used only with PostScript printers. The filter converts the single-byte and multibyte characters used in an international environment to printable PostScript output. Thus, print jobs that include local language characters can be printed on printers where local language fonts are not resident. To use this filter, the printer must support PostScript Level 2 (or higher) or PostScript Level 1 with the composite font extension.

The PostScript fonts can be outline fonts installed on the system, TrueType fonts, or low-resolution bitmap fonts. TrueType fonts and low-resolution bitmap fonts are made available to the filter through an X font server, which requires that the X font server be running. In searching for fonts, the filter first attempts to use PostScript outline fonts. If those are not available, the filter uses high-resolution, rasterized, TrueType fonts. If those are not available, the filter uses low-resolution bitmap fonts.

The `wpsof` filter is sensitive to locale setting. When processing a character, the filter determines if the character is printable in the current locale and uses the codeset part of the locale definition to find an appropriate font (outline, TrueType, or low-resolution). Except for file formats that include Byte Order Mark (that is UTF-16 or UTF-32 format), you must set the locale appropriately before printing files that contain characters in languages other than English.

For example, you can set up a printer configuration file for use with the `wpsof` print filter to convert bitmap fonts for other locales to PostScript when printing files that use UTF-8 encoding. Unicode includes characters for almost all languages, and any given font is limited to a small subset of supported characters. Therefore, you can customize a `unicode conversion preference` entry in the printer configuration file to specify a codeset look-up order favoring the fonts that are available for the language of the text most frequently printed.

The `wpsof` filter prints multilanguage text files by first converting each character in the text file to a matching character in a UNIX codeset for which fonts are available and then converting the file to PostScript. The filter can also print PostScript files that have been generated by some CDE applications.

See `wpsof(8)` and the *System Administration* manual for details on using this print filter.

1.7.2 Print Filters for Specific Local Language Printers

A print filter processes text data for a particular model of printer. The filter handles the device dependencies of the printer and performs device accounting functions. When each print job is complete, the print filter writes an accounting record to the file specified by the `af` field of the printer's entry in the `/etc/printcap` file.

The print filters for local language text printers can handle text files that contain ASCII and local language characters, or output files created by the `nroff` command. When processing `nroff` output, the filter removes multibyte characters that extend beyond the page boundary and translates `nroff` control sequences for underlining, superscripting, and subscripting to control sequences appropriate for the printer. However, the filter does not support multiple `nroff` control sequences on the same character.

The PostScript print filters can print PostScript files in addition to text and `nroff` output files.

A local language print filter can be the specified filter in both the `of` and `if` fields in the `/etc/printcap` file. For general information on `/etc/printcap` entries, see the *System Administration* manual and `printcap(4)`. Supplementary information is provided in `il8n_printing(5)`. A reference page for a specific language (for example, `Japanese(5)`) lists the names of print filters that support printing characters in that language.

The following print filters process text data for Asian languages:

Language	Filter	Printer
Japanese	1a84of	LA84-J
Japanese	1a86of	LA86-J
Japanese	1a90of	LA90-J
Japanese	1a280of	LA280-J
Japanese	1a380of	LA380-J
Japanese	1n03jaof	LN03-J
Japanese	1n05jaof	LN05-J
Japanese	1n82rof processes both PostScript and text data	LN82R
Simplified Chinese	1a88cof	LA88-C
Simplified Chinese	1a380cbof	LA380-CB
Korean	1a380kof	LA380-K
Korean	d1510kaof	DL510-KA

Language	Filter	Printer
Traditional Chinese	cp382dof	CP382-D
Thai	thailpof	EP1050+

1.7.3 Support for Local Language Printers in `/etc/printcap`

The `/etc/printcap` file describes characteristics of each printer on the system. Printer characteristics are specified by symbol/value pairs, where each symbol is a 2-character mnemonic. Each time you submit a print job, the `lpd` printer daemon and printer spooling system uses information in the `/etc/printcap` file to determine how that job is handled.

Table 1–2 describes `/etc/printcap` symbols that are specific to local language printer support. See `printcap(4)` for descriptions of other symbols used in the `/etc/printcap` file. See Section 1.7.4 for an example of using the `lprsetup` command to add several of these options to the `/etc/printcap` file for a local language printer.

Table 1–2: Symbols in `/etc/printcap` File for Local Language Printers

Symbol	Type	Default	Description
ya	str	None	Double-quoted list of keyword value assignments This assignment list specifies most of the printer options related to country-specific support. The option keywords, which are explained following this table, include <code>flocale</code> , <code>font</code> , <code>line</code> , <code>odldb</code> , <code>odlstyle</code> , <code>onehalf</code> , <code>plocale</code> , <code>spcom</code> , <code>tacdata</code> , and <code>tm</code> .
yp	str	NULL	Printer ID that conforms to the WoToTo Standard (for Thai printers)

Table 1–2: Symbols in /etc/printcap File for Local Language Printers (cont.)

Symbol	Type	Default	Description
<code>ys</code>	num	NULL	<p>Size of the SoftODL character cache</p> <p>The <code>ys</code> entry is applied to text print filters. It must be present and its value must be greater than zero to enable on-demand loading of font files. These font files are the ODL support files created by the <code>cgen</code> utility for user-defined characters. The location of the SoftODL support files is identified by the path for system-wide ODL files in the database location configuration file <code>/usr/var/i18n/conf/cp_dirs</code>. ODL files for private UDC databases are not downloaded to printers.</p> <p>For optimal performance, the cache value specified for the <code>ys</code> field should match the printer cache size. To find out the cache size for a particular printer, see the printer's manual.</p>
<code>yt</code>	str	<code>fifo</code>	<p>The SoftODL character replacement method</p> <p>The <code>yt</code> entry applies to text print filters. The value for this entry can be either <code>fifo</code> (first-in/first-out) or <code>lru</code> (least recently used). You can type either uppercase or lowercase letters for these values. To find out which value is appropriate for a particular printer, see the printer's manual.</p>

The `ya` symbol is defined for printing languages whose characters are not included in the Latin-1 character set. The value assigned to the `ya` symbol is a quoted string that can include one or more of the following keywords:

- `flocale=locale_name`

Specifies the locale for interpretation of file text. The print filter uses this locale to validate characters in the text. For an Asian language that is supported by more than one codeset, a difference between the `flocale` and `plocale` values determines whether codeset conversion is done before the file is printed. If `flocale` is not specified, the filter interprets the file in the current locale.
- `font=font_name`

Specifies the name of the outline font for printing PostScript files. This font must be appropriate for the specified `plocale` value.
- `line=number_of_lines`

Specifies the number of lines for each page. When used in combination with the `-w` flag of the `lpr` command, the line number can control the font size and orientation of printed output.
- `odldb=odl_database_path`

Specifies the pathname of the SoftODL database. By default, the printer uses the system-wide database as specified in the `cp_dirs` file.

- `odlstyle=style-NxN`

Specifies the SoftODL font style and size to use, for example, `normal-24x24`. If `odlstyle` is not specified, the default style and size set for the system-wide database is used.

- `onehalf`

For the Thai language, specifies that characters be printed on one-and-one-half lines, rather than three lines, to produce more compressed and natural-looking output. The `onehalf` keyword is valid only for the `thailpof` print filter.

- `plocale=locale_name`

Specifies the printer locale. Print filters for some printers, such as the LA380–CB printer, are country specific because the printer has built-in fonts that are encoded in a particular codeset. For these printers, you must specify `plocale`, and the `codeset` part of `locale_name` must match the codeset of the built-in fonts. Other printers are generic and suitable for printing files in a variety of languages. For these printers, you can use a generic print filter (such as `wwpsof`) and do not need to specify a `plocale` value unless you want to use a font that is not the default for the language being printed.

- `spcom`

Enables space-compensation mode for languages, such as Thai, that contain nonspacing characters. These characters can combine with other characters for display and therefore do not occupy space. Many of the existing tools that align text do not handle nonspacing characters correctly. If you want to print the Thai output that these tools generate, you should specify the `spcom` keyword to ensure proper text alignment in the printed file. This keyword is valid only when used with a Thai print filter or with the `th_TH.TACTIS plocale` value.

- `tacdata=tac_data_path`

Specifies the location of the character code tables used with the `thailpof` print filter. By default, `tac_data_path` is `/usr/lbin/tac_data`.

- `tm`

Enables text morphing for printing Thai characters. Text morphing replaces some characters with others to produce better printed output. See `Thai(5)` for information on text morphing.

1.7.4 Enhancements to Printer Configuration Software

The Printing selection on the SysMan Menu is the desktop application that helps you add, delete, or change the characteristics of the printers on your system. The `lprsetup` utility is an alternative way to do these operations if your system is not running CDE. In both cases, the software performs necessary tasks, such as creating the printer spooling directory, linking the appropriate filter to the printer, and writing the entry for the printer in the `/etc/printcap` file. You must be superuser to run `lprsetup`. See `lprsetup.dat(4)` for information about mapping the product names of supported printers to their system identifiers. See the *System Administration* manual for detailed information and examples for printer setup.

Example 1–1 demonstrates how you use the `lprsetup` command to set up a local language printer, in this case `ln03s-ja`.

Example 1–1: Setting Up a Local Language Printer with `lprsetup`

```
# /usr/sbin/lprsetup 1
Tru64 UNIX Printer Setup Program

Command < add modify delete exit view quit help >: add

Adding printer entry, type '?' for help.

Enter printer name to add [lp11] : 2

Printer Types:

 1. Compaq Advanced Server ClientPS
 2. Compaq Advanced Server ClientText
 3. Compaq LN16
 4. Compaq LN32
 5. Digital Colormate PS
 6. Digital DEClaser 1100
 7. Digital DEClaser 1150
 8. Digital DEClaser 2100
 9. Digital DEClaser 2150
10. Digital DEClaser 2200
11. Digital DEClaser 2250
12. Digital DEClaser 2300
13. Digital DEClaser 2400
14. Digital DEClaser 3200
15. Digital DEClaser 3250
16. Digital DEClaser 3500
17. Digital DEClaser 5100
18. Digital LA100
19. Digital LA120
20. Digital LA210
21. Digital LA280
22. Digital LA30N
23. Digital LA30N A4
24. Digital LA30W
25. Digital LA30W A4
26. Digital LA324
27. Digital LA380
28. Digital LA380CB
```

Example 1-1: Setting Up a Local Language Printer with lprsetup (cont.)

```
29. Digital LA380K
30. Digital LA400
31. Digital LA424
32. Digital LA50
33. Digital LA600
34. Digital LA70
35. Digital LA75
36. Digital LA84
37. Digital LA86
38. Digital LA88
39. Digital LA88C
40. Digital LA90
41. Digital LG02
42. Digital LG04 Plus
43. Digital LG05 Plus

Press 'ENTER' to continue scrolling, type '(q)uit' to end scrolling: q

Help Types:

    ?          - General help
    printer?   - Specific printer type information

Enter index number, help type, '(q)uit', or 'ENTER' [Generic Unknown type]:
ln03ja 3
:
:

Enter printer synonym: draft 4

Enter printer synonym:

Set device pathname 'lp' [] ? /foo

Do you want to capture print job accounting data ([y|n])? n

Set spooler directory 'sd' [/var/spool/printers/lpd11] ?

Set printer error log file 'lf' [/var/adm/printers/lp11.lperr] ?

Enter the name of the printcap symbol you wish to modify. Other
valid entries are:
    'q'      to quit (no more changes)
    'p'      to print the symbols you have specified so far
    'l'      to list all of the possible symbols and defaults
The names of the printcap symbols are:

af br cf ct df dn du fc ff fo fs gf if lf lo lp
mc mj mx nc nf of on pl pp ps pw px py rf rm rp
rs rw sb sc sd sf sh st tf tr vf xc xf xn xs ya
yd yj yp ys yt

Enter symbol name: yt 5

Enter a new value for symbol 'yt'? [none] Return

Enter symbol name: ?

Enter the name of the printcap symbol you wish to modify. Other
valid entries are:
    'q'      to quit (no more changes)
```

Example 1–1: Setting Up a Local Language Printer with lprsetup (cont.)

```
'p'      to print the symbols you have specified so far
'l'      to list all of the possible symbols and defaults
The names of the printcap symbols are:

af br cf ct df dn du fc ff fo fs gf if lf lo lp
mc mj mx nc nf of on pl pp ps pw px py rf rm rp
rs rw sb sc sd sf sh st tf tr vf xc xf xn xs ya
yd yj yp ys yt

Enter symbol name: q [6]
Printer #11
-----
Symbol  type  value
-----
if     STR    /usr/lbin/ppdof +OPageSize=Letter +Ctektronix740.rpd
lf     STR    /var/adm/printers/lp11.lperr
lp     STR    /foo
mx     INT    0
of     STR    /usr/lbin/ppdof +OPageSize=Letter +Ctektronix740.rpd
pl     INT    66
pw     INT    0
rw     BOOL   on
sd     STR    /var/spool/printers/lpd11
xf     STR    /usr/lbin/xf

Are these the final values for printer 11 ? [y] [Return]

Adding comments to printcap file for new printer, type '?' for help.
Do you want to add comments to the printcap file [n] ? : [Return]

Setup activity is complete for this printer.
Verify that the printer works properly by using
the lpr(1) command to send files to the printer.

Command < add modify delete exit view quit help >: e[7]
```

- 1 Invokes the `lprsetup` program.
- 2 Displays the available printer types (see the reference pages for specific languages for information on the local language printers supported by the `lprsetup` command).
- 3 Enters the printer type. To obtain a description of individual printers, enter `printer?`.
- 4 The utility displays a series of prompts that allow you to specify a synonym for the printer name, device path, whether accounting data will be maintained, and selection of a spooler directory and error log. Enter a question mark to obtain help on any of the prompts.
- 5 Prompts you to enter a printcap symbol. See Table 1–2 for the symbols and parameters of importance to internationalized systems.

For example, `ys` sets the cache size that the SoftODL service uses. By default, this value is the appropriate cache size for the printer and is stored as the value of the `ys` symbol in the `/etc/printcap` file.

- 6] Quits the `lprsetup` dialogue. The utility displays the values assigned and prompts for verification. Once verified, you are prompted to add comments to the `/etc/printcap` file.
- 7] Quits the program to indicate no more changes are needed to the `/etc/printcap` file.

1.7.5 Print Commands and the Printer Daemon

The `lp`, `lpc`, `lpd`, `lpq`, `lpr`, `lprm`, and `lpstat` commands handle the features added to the print subsystem for Asian and other languages not in the Latin-1 group. For example, the `lpr` command includes the `-A` option and additional values for the `-O` option to give you access to such features. See `lpr(1)` for details about local language options and values.

1.7.6 Choosing PostScript Fonts for Different Locales

The fonts for the Chinese and Korean languages do not fit in the memory of most PostScript printers. Fonts for the Thai language and some European languages do fit in memory, but are large enough that they do not fit in printer memory along with fonts for other languages.

For PostScript printers in which language-specific fonts are not printer resident, the `wpsof` print filter (see Section 1.7.1.2) provides a solution. In this case, you specify in a printer's configuration file the names of the fonts you want to use for different languages. The `wpsof` print filter can also create PostScript output from bitmap fonts when PostScript fonts are not available for a particular codeset. See `wpsof(8)` for more information about using this print filter.

The following list associates languages and codesets with the appropriate set of PostScript fonts:

- Western European, Latin-1 (* . ISO8859-1)
PostScript fonts for Latin-1 languages are printer resident; they are not installed from software subsets.
- Hungarian, Czech, Slovak, Slovene (* . ISO8859-2)
`Arial-Bold-ISOLatin2`
`Arial-BoldItalic-ISOLatin2`
`Arial-Italic-ISOLatin2`
`Arial-ISOLatin2`
`ArialNarrow-Bold-ISOLatin2`
`ArialNarrow-BoldItalic-ISOLatin2`
`ArialNarrow-Italic-ISOLatin2`

ArialNarrow-ISOLatin2
 BookAntiqua-Bold-ISOLatin2
 BookAntiqua-BoldItalic-ISOLatin2
 BookAntiqua-Italic-ISOLatin2
 BookAntiqua-ISOLatin2
 BookmanOldStyle-Bold-ISOLatin2
 BookmanOldStyle-BoldItalic-ISOLatin2
 BookmanOldStyle-Italic-ISOLatin2
 BookmanOldStyle-ISOLatin2
 CenturyGothic-Bold-ISOLatin2
 CenturyGothic-BoldItalic-ISOLatin2
 CenturyGothic-Italic-ISOLatin2
 CenturyGothic-ISOLatin2
 CenturySchoolbook-Bold-ISOLatin2
 CenturySchoolbook-BoldItalic-ISOLatin2
 CenturySchoolbook-Italic-ISOLatin2
 CenturySchoolbook-ISOLatin2
 Courier-Bold-ISOLatin2
 Courier-BoldItalic-ISOLatin2
 Courier-Italic-ISOLatin2
 Courier-ISOLatin2
 MonotypeCorsiva-ISOLatin2
 TimesNewRoman-Bold-ISOLatin2
 TimesNewRoman-BoldItalic-ISOLatin2
 TimesNewRoman-Italic-ISOLatin2
 TimesNewRoman-ISOLatin2

- **Russian (*.ISO8859-5)**

Arial-Bold-ISOLatinCyrillic
 Arial-BoldInclined-ISOLatinCyrillic
 Arial-Inclined-ISOLatinCyrillic
 Arial-ISOLatinCyrillic
 Courier-Bold-ISOLatinCyrillic
 Courier-BoldInclined-ISOLatinCyrillic
 Courier-Inclined-ISOLatinCyrillic
 Courier-ISOLatinCyrillic
 Nimrod-Bold-ISOLatinCyrillic
 Nimrod-BoldInclined-ISOLatinCyrillic
 Nimrod-Inclined-ISOLatinCyrillic
 Nimrod-ISOLatinCyrillic
 Plantin-Bold-ISOLatinCyrillic
 Plantin-BoldInclined-ISOLatinCyrillic
 Plantin-Inclined-ISOLatinCyrillic
 Plantin-ISOLatinCyrillic
 TimesNewRoman-Bold-ISOLatinCyrillic
 TimesNewRoman-BoldInclined-ISOLatinCyrillic
 TimesNewRoman-Inclined-ISOLatinCyrillic
 TimesNewRoman-ISOLatinCyrillic

- **Greek (*.ISO8859-7)**

Arial-Bold-ISOLatinGreek
Arial-BoldInclined-ISOLatinGreek
Arial-Inclined-ISOLatinGreek
Arial-ISOLatinGreek
Courier-Bold-ISOLatinGreek
Courier-BoldInclined-ISOLatinGreek
Courier-Inclined-ISOLatinGreek
Courier-ISOLatinGreek
TimesNewRoman-Bold-ISOLatinGreek
TimesNewRoman-BoldInclined-ISOLatinGreek
TimesNewRoman-Inclined-ISOLatinGreek
TimesNewRoman-ISOLatinGreek

- **Hebrew (* .ISO8859-8)**

David-Bold-ISOLatinHebrew
David-BoldOblique-ISOLatinHebrew
David-ISOLatinHebrew
David-Oblique-ISOLatinHebrew
FrankRuhl-Bold-ISOLatinHebrew
FrankRuhl-BoldOblique-ISOLatinHebrew
FrankRuhl-ISOLatinHebrew
FrankRuhl-Oblique-ISOLatinHebrew
Miriam-Bold-ISOLatinHebrew
Miriam-BoldOblique-ISOLatinHebrew
Miriam-ISOLatinHebrew
Miriam-Oblique-ISOLatinHebrew
MiriamFixed-Bold-ISOLatinHebrew
MiriamFixed-BoldOblique-ISOLatinHebrew
MiriamFixed-ISOLatinHebrew
MiriamFixed-Oblique-ISOLatinHebrew
NarkissTam-Bold-ISOLatinHebrew
NarkissTam-BoldOblique-ISOLatinHebrew
NarkissTam-ISOLatinHebrew
NarkissTam-Oblique-ISOLatinHebrew

- **Turkish (* .ISO8859-9)**

Arial-Bold-ISOLatin5
Arial-BoldItalic-ISOLatin5
Arial-Italic-ISOLatin5
Arial-ISOLatin5
ArialNarrow-Bold-ISOLatin5
ArialNarrow-BoldItalic-ISOLatin5
ArialNarrow-Italic-ISOLatin5
ArialNarrow-ISOLatin5
BookAntiqua-Bold-ISOLatin5
BookAntiqua-BoldItalic-ISOLatin5
BookAntiqua-Italic-ISOLatin5
BookAntiqua-ISOLatin5
BookmanOldStyle-Bold-ISOLatin5
BookmanOldStyle-BoldItalic-ISOLatin5

BookmanOldStyle-Italic-ISOLatin5
 BookmanOldStyle-ISOLatin5
 CenturyGothic-Bold-ISOLatin5
 CenturyGothic-BoldItalic-ISOLatin5
 CenturyGothic-Italic-ISOLatin5
 CenturyGothic-ISOLatin5
 CenturySchoolbook-Bold-ISOLatin5
 CenturySchoolbook-BoldItalic-ISOLatin5
 CenturySchoolbook-Italic-ISOLatin5
 CenturySchoolbook-ISOLatin5
 Courier-Bold-ISOLatin5
 Courier-BoldItalic-ISOLatin5
 Courier-Italic-ISOLatin5
 Courier-ISOLatin5
 MonotypeCorsiva-ISOLatin5
 TimesNewRoman-Bold-ISOLatin5
 TimesNewRoman-BoldItalic-ISOLatin5
 TimesNewRoman-Italic-ISOLatin5
 TimesNewRoman-ISOLatin5

- Latin-9 (* .ISO8859-15)

No PostScript fonts are supplied for locales using Latin-9 encoding. However, a default printer configuration file (PCF) is available for Latin-9. When specified for the `wpsof` print filter, this file enables automatic conversion of Latin-9 bitmap fonts to PostScript. See `wpsof(8)` for more information.

- Unicode (* .UTF-8)

The X locale database file used by applications running in the `universal.UTF-8`, `en_US.UTF-8`, or Asian locales (Chinese, Japanese, Korean) contains font definitions that include all the fonts used with the operating system. This enables applications under `en_US.UTF-8` to display all the font characters installed with Worldwide Language Support (WLS). Applications under the Asian locales display all the font characters installed with WLS, except for ISO8859-2, -4, -5, -7, -8, -9, and TACTIS.

See `wpsof(8)` and `Unicode(5)` for more information.

- Traditional Chinese (* .dechanyu).

The operating system provides the following traditional Chinese outline fonts for printing on PostScript printers and for display through Level II Display Postscript extension. For information on use of these fonts with PostScript printers, Display PostScript, or display through a rasterizer, see the *Technical Reference for Using Chinese Features* online manual.

Sung-Light-CNS11643
 Hei-Light-CNS11643

- Simplified Chinese (* .dechanzi).

The operating system provides the following simplified Chinese outline fonts for printing on PostScript printers and for display through Level II Display Postscript extension. For information on use of these fonts with PostScript printers, Display PostScript, or display through a rasterizer, see the *Technical Reference for Using Chinese Features* online manual.

XiSong-GB2312-80
Hei-GB2312-80

- **Korean** (*.deckorean)

Munjo

- **Japanese** (*.deckanji)

PostScript fonts for the Japanese language are normally printer resident; they are not installed from software subsets. However, you can set up a printer configuration file (PCF) with the `wpsof` print filter to convert Japanese bitmap fonts to PostScript for printing files that use Japanese encoding. See `wpsof(8)` for more information.

- **Thai** (*.TACTIS)

AngsanaUPC-Bold
AngsanaUPC-BoldItalic
AngsanaUPC-Italic
AngsanaUPC-Light
CordiaUPC-Bold
CordiaUPC-BoldItalic
CordiaUPC-Italic
CordiaUPC-Light
EucrosiaUPC-Bold
EucrosiaUPC-BoldItalic
EucrosiaUPC-Italic
EucrosiaUPC-Light
FreesiaUPC-Bold
FreesiaUPC-BoldItalic
FreesiaUPC-Italic
FreesiaUPC-Light
IrisUPC-Bold
IrisUPC-BoldItalic
IrisUPC-Italic
IrisUPC-Light
JasmineUPC-Bold
JasmineUPC-BoldItalic
JasmineUPC-Italic
JasmineUPC-Light
KodchiangUPC-Bold
KodchiangUPC-BoldItalic
KodchiangUPC-Italic
KodchiangUPC-Light
LilyUPC-Bold

LilyUPC-BoldItalic
LilyUPC-Italic
LilyUPC-Light
WaterlilyUPC-Bold
WaterlilyUPC-BoldItalic
WaterlilyUPC-Italic
WaterlilyUPC-Light
YuccaUPC-Bold
YuccaUPC-BoldItalic
YuccaUPC-Italic
YuccaUPC-Light

1.8 Using Mail in a Multilanguage Environment

The operating system provides enhanced versions of the following commands and utilities to handle languages based on multibyte-character codesets:

- `sendmail`
- `mailx`
- MH (mail handler)

The following sections discuss enhancements to these components and codeset conversion done by the `comsat` server. See `sendmail(8)`, `mailx(1)`, `mh(1)`, `comsat(8)` for more complete software descriptions.

1.8.1 The `sendmail` Utility

The `sendmail` utility, which is a back end to several user commands, is configured by default to support 8-bit data. The configuration that supports 8-bit data is required for multibyte character support. See `sendmail(8)` for restrictions that apply to the 8-bit configuration.

1.8.2 The `mailx` Command and MH Commands

The `mailx` command and all applicable commands in the MH system support the conversion of mail messages between the mail interchange codeset (used to transfer messages to some hosts) and a user's application codeset. For example, if the mail interchange codeset is `ISO-2022-JP` and the application codeset is `euCJP`, the `mailx` or MH command converts incoming messages to the Japanese EUC codeset before displaying them.

To prevent data loss, when incoming messages are stored in mail folders, the messages are encoded in the codeset in which they are received. Codeset conversion takes place when you extract or display the messages.

To communicate mail interchange code information to other systems, outgoing messages include two additional header lines like the following:

Mime-Version: 1.0

Content-Type: TEXT/PLAIN; charset=ISO-2022-JP

The `charset` field in the preceding example specifies the mail interchange codeset, in this case, `ISO-2022-JP`. This codeset is an ISO 7-bit state-dependent codeset for Japanese characters. Codesets other than those that are part of the ISO standard are identified by the prefix `X-` in the codeset name. For example, when DEC Hanyu is the codeset used for mail interchange, the following header lines are included in outgoing mail messages:

Mime-Version: 1.0

Content-Type: TEXT/PLAIN; charset=X-dechanyu

The `mailx` command and `MH` commands determine the application codeset and set the mail interchange codeset for incoming and outgoing messages based on certain values. The following lists describe, in priority order of highest to lowest, the values these commands use.

- The application codeset is determined from one of the following:
 1. The setting of the `LANG` environment variable
 2. The value of the `lang` component in the `$HOME/.mailrc` file (for the `mailx` command) or the `$HOME/.mh_profile` file (for `MH` commands)
- The mail interchange codeset applied to incoming messages is determined from one of the following:
 1. The `charset` field in the mail header, if additional header lines are present in the message
 2. The codeset specified as the system-wide mail interchange default in the `/usr/lib/mail-codesets` file
If you create this file, make sure it contains the name of a locale as the only entry.

If neither of the preceding values is available, codeset conversion does not occur.

- The mail interchange codeset applied to outgoing messages is determined from one of the following:
 1. The setting of the `EXCODE` environment variable
 2. The setting of the `excode` component as defined in the `$HOME/.mailrc` file (for `mailx` users) or the `$HOME/.mh_profile` file (for users of `MH` commands)

3. The content of the `/usr/lib/mail-codesets` file

If a codeset is not determined for outgoing mail interchange, the mail is sent with no codeset identifier.

1.8.3 The `comsat` Server

The `comsat` server, which notifies you of incoming mail messages, always attempts to convert incoming mail messages from the mail interchange codeset to your application codeset. The following lists describe, in priority order of highest to lowest, the values that the `comsat` server uses to determine the codesets of the mail interchange and your application.

- The mail interchange codeset value is determined from one of the following:
 1. The `charset` field, if included in the mail message header
 2. The codeset specified as the system-wide mail interchange default in the `/usr/lib/mail-codesets` fileIf neither of the preceding values is available, codeset conversion does not occur.
- The application codeset value is determined from one of the following:
 1. The application codeset defined for the `atty` driver of the user's system
 2. The codeset name in the `$(HOME)/.codeset_device_name` file, where `device_name` is the name of the terminal device for the current session

1.9 Displaying Reference Pages in Languages Other Than English

As with the operating system, internationalized applications frequently supply online reference pages (manpages) to document the application and its components. The operating system includes enhanced versions of the `nroff`, `tbl`, and `man` commands to support this requirement.

The `nroff` and `tbl` commands are tools used primarily by programmers to create reference pages. These commands are described in the *Writing Software for the International Market* manual.

The `man` command formats and displays the reference page and can handle multibyte characters in reference page files. By default, the `man` command automatically searches for reference pages in the `/usr/share/locale_name/man` directory before searching the `/usr/share/man` and `/usr/local/man` directories. Therefore, if the `LANG`

environment variable is set to an installed locale and if reference page translations are available for that locale, the `man` command automatically displays reference pages in the appropriate language.

In addition, the `man` command automatically applies codeset conversion (assuming the availability of appropriate converters) when reference page translations for a particular language are encoded in a codeset that does not match the codeset of the user's locale. See `man(1)` for information about redefining the `man` command search path and for more details about codeset conversion.

1.10 Converting Data Files from One Codeset to Another

Each locale is based on a specific codeset. Therefore, when an application uses a file whose data is coded in one codeset and runs in a locale based on another codeset, character interpretation may be meaningless. You may need to set the process environment to a particular locale and use a data file created with a codeset different from the one on which the locale is based. The data file in question might be appropriate for a given language and in a codeset different from your locale for one of the following reasons:

- The data file might have been created on another vendor's system by using a locale based on a vendor-specific codeset. For example, the integration of PCs into the enterprise computing environment increases the likelihood that UNIX users need to process files for which the data encoding is in MS-DOS code page format.
- The locale could be one of several UNIX locales that support the same Asian language, such as Japanese. Asian languages are typically supported by a variety of locales, each based on a different codeset.
- The data file could be in Unicode transformation format (UTF-8, UTF-16, or UTF-32). If characters in this file are to be printed or displayed on the screen, they might need to be converted to encodings for which fonts are available.

You can convert a data file from one codeset to another by using the `iconv` command. Consider the following example:

```
% iconv -f SJIS -t eucJP accounts_local \  
>> accounts_central
```

This `iconv` command performs the following tasks:

- Reads data in the `accounts_local` file, which is encoded in the SJIS codeset
- Converts the data to the `eucJP` codeset
- Appends the results to the `accounts_central` file

An application programmer, on the other hand, might use the `iconv_open()`, `iconv()`, and `iconv_close()` functions for the same purpose. Many commands and utilities, such as the `man` command and internationalized print filters, use the `iconv()` functions and associated converters to perform codeset conversion on the user's behalf.

See the *Writing Software for the International Market* manual and `iconv(3)` for a full description of how to use `iconv` and algorithmic and table converters in internationalized programs.

1.11 Miscellaneous Base System Commands

The following list includes information about features and restrictions that apply when using traditional UNIX commands in local language environments:

- `file` and `jfile` alias

The `file` command recognizes files encoded in Unicode or ISO 10646 formats (16-bit UCS-2 or 32-bit UTF-32). For other kinds of text files, the command recognizes when the character encoding is valid for the codeset of the current locale. The `file` command also has a `jfile` alias. When you use this alias, the command recognizes the most commonly used encodings for Japanese (DEC Kanji, Japanese EUC, Shift JIS, and 7-bit JIS) regardless of the current locale setting. For more information, see `file(1)`.

- `rlogin`

When you use the `rlogin` command to log on to a Tru64 UNIX system from an ULTRIX system, be sure to specify the `-8` flag to pass 8-bit data without stripping. Otherwise, you will have problems entering non-ASCII characters from your terminal.

If you view a large data file while logged on to the remote system, use a pager command, such as `pg`, and not the Hold Screen key. The `-8` option sets the terminal mode of the original host to RAW, disabling flow control. So, if data is sent to the terminal at a rate faster than the terminal can handle it, some data is lost when you use the Hold Screen key.

This `rlogin` restriction applies when logging in from an ULTRIX system or when logging in from any UNIX system whose software does not fully support 8-bit data format.

- Emacs editor

The operating system includes the multilingual Emacs software from the Free Software Foundation. Before using this editor, you must add the `/usr/i18n/mule/bin` directory to your process-specific search path. You can then invoke this editor by using the `mule` command.

- `vi` and `more`

The `vi` and `more` commands discard text that follows an invalid multibyte character. If you encounter this problem, it is likely that your locale setting is not correct for the text being viewed or edited. In this case, reset your locale to one that matches the text and invoke the command again.

When used with Thai characters, the `vi` editor may wrap lines before the right boundary of the screen. This happens because Thai text includes nonspacing characters, which contribute to the character count but not to the display width. The editor wraps lines based on character count. For example, `vi` may wrap a line after entry of 80 characters, even though these characters do not occupy 80 columns on the screen.

- Using local language user names and file names

It is a limitation of UNIX file systems that you cannot use a multibyte character whose second or subsequent byte is an ASCII slash (/) in names of files, users, or other objects. This limitation means that some user-defined characters in the DEC Hanzi and DEC Kanji codesets and certain characters (CNS Plane 2 characters) in the DEC Hanyu codeset cannot be used in these names.

2

Using Asian Input Methods and Terminal Drivers

This chapter describes how to use the operating system in Asian environments and includes information on how to perform the following tasks:

- Establish the appropriate input method for Japanese, Chinese, or Korean characters and phrases (Section 2.1)
- Enable keyboard layouts for Asian languages (Section 2.2)
- Define settings for terminal code, application code, and other characteristics of the terminal driver that are appropriate for Asian languages (Section 2.3)

This chapter describes the operating system commands and utilities you can use to configure the system for an Asian environment. If you are a system manager or administrator with superuser privileges, you can also use the menu-oriented Configure International Software utility found on the SysMan Menu under Software. See Section 1.2 for information on the configuration options available to you with this utility.

The operating system documentation set includes four manuals that offer technical reference and user information for the operating system in an Asian environment. The manuals have the following titles:

- *Technical Reference for Using Chinese Features*
- *Technical Reference for Using Japanese Features*
- *Technical Reference for Using Korean Features*
- *Technical Reference for Using Thai Features*

The Tru64 UNIX documentation is available on the World Wide Web at the following URL:

<http://www.tru64unix.compaq.com/docs/>

2.1 Determining Input Method

In a Motif environment, the operating system provides Asian language input methods that allow you to enter characters or phrases from your alphabetic keyboard, convert that keyboard entry into the appropriate Asian

characters, and transmit those characters to an X client application. The *Writing Software for the International Market* manual describes how to write internationalized applications that take advantage of Asian input methods. Because use of a particular input method is language dependent, that information is covered in the technical references listed in the previous section and in the online help associated with each input method.

This section describes the input methods available on your system and how to invoke them.

Before you can use an Asian input method, you must establish an appropriate locale; set a preediting, or interactive, style; and start the input method server.

See Section 1.3 for information on establishing a locale.

A preediting, or interactive, style determines how you will interact with the input method to enter characters from the keyboard. That is, an Asian input method server often requires you to type multiple keystrokes to compose an Asian character. This process is called preediting. The X Input Method specification, which internationalized applications follow, defines the following preediting styles:

- **On-the-Spot**
The characters being composed are displayed directly in the application window. Application data is moved to allow the preediting string to display at the point of character insertion.
- **Over-the-Spot**
The characters being composed are displayed in a window that is positioned over the point of insertion.
- **Off-the-Spot**
The characters being composed are displayed in a window that is within the application window but not over the point of insertion. Often, the window for the preediting string appears at the bottom of the application window. In this case, the preediting window may block the last line of text from view in the application window. You can resize the application window to make this last line visible.
- **Root Window**
The characters being composed are displayed in a child window of the application Root Window.

To set the preediting style, you can use `dtterm` with an `-xrm` option. You can also use the Session Manager Options Menu. However, the simplest method is to use the `dtimsstart` command as follows:

1. From the command line, enter `/usr/dt/bin/dtimsstart`.

2. Click on the appropriate radio button to choose a preediting style.
3. Click on OK.

For some of the input styles selected in an application, the preediting and status windows are not redrawn correctly if the application window is hidden by other windows. To correct this problem, click on the application window to bring it to the foreground (in focus).

The operating system provides input servers for traditional and simplified Chinese (Hanyu and Hanzi respectively), Japanese, and Korean. Each input server provides you with one or more input methods appropriate for variations within each language. For example, the simplified Chinese Input Server provides input methods for these input styles: 5-Stroke, 5-Shape, Pin-Yin, Symbol, Telex code, Qu-Wei, and Intelligent ABC.

An input server runs as an independent process and communicates with an application to handle input operations. You can run the input server remotely or locally. After you start the input server and activate an input method, any internationalized Motif application that supports the selected input method can communicate with the input server.

Ordinarily, you must start the application after you start the input server. However, if the application contains an `XmText` or `XmTextField` widget with the `Reconnectable` resource set to `True`, the application can connect with the input server regardless of which one starts first.

You can start an individual input server from the command line or, for non-Japanese Asian languages, you can use the `dxim` Input Server. The `dxim` Input Server is a menu-oriented server that supports Korean, traditional Chinese, simplified Chinese, and Phrase input methods under a single protocol and allows you to switch from one input method engine to another from the same or different locales.

When a client application connects to `dxim` and you use the keyboard to enter characters, the `dxim` Input Server converts the character data and sends it to the application. The conversion is English to an Asian language variant using traditional Chinese, simplified Chinese, or Korean Input Method server engines (`dxhanyuim`, `dxhanziim`, and `dxhangulim`, respectively). `dxim` selects an input method based on the client's locale and the input method server engines (called classes) and methods that you have activated.

The `dxim` Input Server supports only Off-the-Spot and Root Window preediting styles. Also, the `dxim` Phrase input is different from the operating system Phrase Utility. While the `dxim` Phrase Input Server method allows you to store characters and phrases in a manner similar to the operating system Phrase Utility, `dxim` does not support the Phrase Utility phrase database.

The `dxim` Input Server provides the following functions:

- Activate or deactivate one or more input servers (simplified Chinese, traditional Chinese, Korean, and Phrase) and establish defaults for the system or for individual users.
- Activate or deactivate one or more input server methods (for example, Tsang-Chi input method under the traditional Chinese Input Server) and establish defaults for the system or for individual users.
- Customize simplified Chinese Input Server methods.
- Customize the keys used to invoke input servers and methods, the appearance of input server windows, and the volume of error warnings.
- Customize the appearance of input server windows (color, font size, and line spacing).

The `dxim` Input Server can support multiple client applications running under different locales. When a client connects to `dxim`, the Input Server determines the client's locale and which input methods are active and usable by the client. It then uses the appropriate input method to service the clients' needs. Depending on locale, `dxim` does not restrict an application to a single input server or method. For example, an application under the `ko_KR.eucKR` locale is restricted to the Korean Input Server because it recognizes only Korean characters. However, an application running under `zh_TW.dechanyu`, can use simplified or traditional Chinese Input Server methods because simplified and traditional characters are compatible. For more information on `dxim`, see the `dxim` online help or `dxim(1X)`.

In addition to `dxim`, you can also run each input method server individually. The following input method servers are available in the operating system, along with the input styles that each server supports:

- `dxhanguim`, the Korean Input Server, which supports all four preediting styles (Over-the-Spot, Off-the-Spot, Root Window, and On-the-Spot)
- `dxhanyuim`, the traditional Chinese Input Server, which supports the Off-the-Spot and Root Window preediting styles
- `dxhanziim`, the simplified Chinese Input Server, which supports the Off-the-Spot and Root Window preediting styles
- `dxjim`, the Japanese Input Server, which supports the On-the-Spot, Over-the-Spot, and Root Window preediting styles

Each of these servers has a corresponding reference page and each language has a corresponding technical reference.

The applications that you run may support more, fewer, or none of the preediting styles supported by a particular input server. The `preedit` option

None applies when an input server rejects all input styles supported by the application.

In the CDE, the appropriate input server automatically starts when you select the session language. See Section 3.3 for restrictions that may require you to start an input server manually. For `dxim`, keep in mind that you must stop any active input server process before starting `dxim`.

For programming information on how to specify input styles and establish a priority order for them, see the *Writing Software for the International Market* manual.

2.2 Determining the Keyboard Input Mode Switch State

Asian writing systems include thousands of ideographs, but Asian keyboards include keys for only a small number of these characters. For Asian languages, you use an input method (incorporating control-key sequences, keypad-key sequences, or options in a workstation application) to convert one or more characters that you can input directly from the keyboard to other kinds of characters. Section 2.1 and the following reference pages discuss input methods for Asian languages: `dxim(1X)`, `dxhanziim(1X)`, `dxhanyuim(1X)`, `dxhangulim(1X)`, `dxjim(1X)`, and `Thai(5)`.

If your keyboard has a mode-switch LED (light-emitting diode), it is turned on or off, depending on whether you last toggled the special input mode on or off.

If you are using a workstation and your language is set to an Asian language, you can display the mode-switch LED on the screen by invoking the Keyboard Indicator application with the `-map` option, as follows:

```
% /usr/bin/X11/kb_indicator -map &
```

The `-map` option starts a Motif application that emulates a mode-switch LED. The application window contains one button, which is displayed as on or off, corresponding to the input mode state. If input mode switching is supported for your current language setting, you can click on this button to toggle in and out of input mode.

You can have only one Keyboard Indicator application running during your session. To stop the application, press `Ctrl/c` in the window from which you started the application or enter the following `kill` command (where `process_id` is the application's process ID):

```
% kill -INT process_id
```

If the Keyboard Indicator application is stopped by any other means, you must erase the server status for the application so that it can be restarted

cleanly. To erase server status, enter the following command before restarting the application:

```
% /usr/bin/X11/kb_indicator -clear
```

If your language is set to Hebrew, the Keyboard Manager application (`/usr/bin/X11/decwkm`) provides the same function as the Keyboard Indicator application provides for Asian languages.

2.3 Using Terminal Interface Features for Asian Languages

The Asian terminal driver (`atty`) and the Thai terminal driver (`ttty`) support input and output of English and other language characters over asynchronous terminal lines. When one or both of these drivers are installed, you can set terminal line characteristics to be appropriate for the language you are using. The driver's local language capabilities are supported in the following terminal configurations:

- Terminal connected directly to the host machine by means of a serial line
- Terminal connected through LAT to the host system
- Terminal connected through TCP/IP to the host system

See `atty(7)` and `ttty(7)` for more information about these terminal drivers.

The `stty` command can enable support for multibyte codesets and special character manipulation capabilities, such as the following:

- Automatic codeset conversion between terminal and application (Section 2.3.1)
- Line editing of multibyte characters (Section 2.3.2)
- Japanese input method (Kana-Kanji conversion) (Section 2.3.3)
- Thai terminal support (Section 2.3.4)
- Phrase input (Appendix B)

This section provides general information about using the `stty` command to enable features added to the terminal subsystem for Asian languages.

The `stty` utility sets or reports on terminal input and output characteristics of the device that is the utility's standard input. Table 2-1 describes the `stty` options that set line discipline for Asian languages.

Table 2–1: The stty Command Options for Controlling Terminal Line Discipline

stty Option	Description
<code>adec</code>	Sets the terminal line discipline to handle multibyte data and the processing environment appropriate for simplified Chinese (Hanzi), traditional Chinese (Hanyu), and Korean (Hangul) codesets. This option is supported for both the STREAMS and BSD terminal drivers.
<code>jdec</code>	Sets the terminal line discipline to handle multibyte data and the processing environment appropriate for Japanese codesets. This option sets terminal code to <code>dec</code> and application code to <code>eucJP</code> . The <code>jdec</code> option is supported for both the STREAMS and BSD terminal drivers.
<code>tdec</code>	Sets the terminal line discipline to handle Thai characters and the processing environment appropriate for the Thai codeset. This option is supported only for the BSD terminal driver.
<code>dec</code>	Sets the terminal line discipline back to the default, or standard, <code>tty</code> line discipline and clears characteristics that preceding <code>stty</code> commands may have set for application and terminal code. This option is supported for both the STREAMS and BSD terminal drivers.

Note

Do not set the terminal line discipline to `jdec` or `adec` from a console set up for kernel debugging (running the `KDEBUG` driver). Doing so may cause the console to hang.

The `stty` command requires an appropriate locale setting to be in effect before you can change the terminal line discipline to support that locale. For example, to set your terminal line discipline to handle Korean, enter the following:

```
% setenv LANG ko_KR.deckorean
% stty adec
```

To set your terminal line discipline back to the `tty` default, enter the following:

```
% stty dec
```

Note

When your terminal line discipline is not set to the `tty` default and you want to switch to another nondefault option (to switch from `jdec` to `adec`, for example), first enter the `stty dec` command to clear any application or terminal characteristics

that may not be appropriate for the new setting. The following example switches a terminal line discipline from its current setting of `adec` to `jdec`:

```
% stty dec
% stty jdec
```

The `stty` command entered with the `-a` option or `-all` argument displays the following settings for the current terminal line discipline:

```
% stty adec
% stty all
atty disc:speed 9600 baud; 24 rows; 80 columns
erase = ^?; werase = ^W; kill = ^U; intr = ^C; quit = ^\; susp = ^Z
dsusp = ^Y; eof = ^D; eol <undef>; eol2 <undef>; stop = ^S; start = ^Q
lnext = ^V; discard = ^O; reprint = ^R; status <undef>; time = 0
min = 1
-parenb -parodd cs8 -cstopb hupcl cread -clocal
-ignbrk brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl -iuclc
ixon -ixany -ixoff imaxbel
isig icanon -xcase echo echoe echok -echonl -noflsh -mdmbuf -nohang
-tostop echoctl -echoprt echoke -altwerase iexten -nokerninfo
opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel tabs -onoet
-odl lru size=256
-sim key= class=
tcode=dec acode=deckanji
```

If you are superuser, you can use the `wwconfig` command or the Configure International Software utility from the SysMan Menu to perform the following tasks:

- Define Asian `tty` features (including Thai) and merge those features into the system configuration file. The features you can define include the following:
 - The terminal driver protocol (BSD or Streams).
 - One or more UTX options (Kana-Kanji, On Demand Font Loading, and Software Phrase Input Method).
 - Codeset support (BIG-5, Mitac Telex, simplified/traditional Chinese mapping, or Unicode).
 - The number of UTX pseudodevices.
- List the currently installed or currently active settings for Asian `tty` support.
- Delete one or more Asian `tty` features from the system configuration file.

For more information, see `wwconfig(8)` or the Configure International Software online help.

2.3.1 Converting Between Application and Terminal Codesets

Many terminals support only one codeset, which is a problem when you work on one terminal and need to run applications in locales (particularly Asian locales) that are based on a variety of codesets. Therefore, the `atty` driver provides a mechanism for converting between the codeset that an application uses and the codeset that a terminal supports. You control codeset conversion by using options on the `stty` command line.

The `adec`, `jdec`, and `dec` options of the `stty` command set terminal code and application code appropriately for HP terminals and workstations. You need to explicitly use the `tcode` option, for example, if you are logging in from a Japanese terminal that does not support the same codeset as HP terminals and workstations.

Table 2–2 specifies `stty` options that explicitly set terminal and application code.

Table 2–2: The `stty` Options to Explicitly Set Application and Terminal Code

stty Option	Description
<code>acode codeset</code>	Sets application code to <code>codeset</code> .
<code>tcode codeset</code>	Sets terminal code to <code>codeset</code> .
<code>code codeset</code>	Sets both terminal code and application code to <code>codeset</code> .
<code>conv (-conv)</code>	Enables or disables codeset conversion between internal code and the terminal or application code. Codeset conversion must be enabled for Asian terminal features to work.

The following command lets you run an application that uses DEC Kanji on a terminal that supports only Shifted JIS (a codeset prevalent in the Japanese personal computer market):

```
% stty acode deckanji tcode SJIS
```

The technical reference manuals for the Asian language features, available from the Programming Bookshelf of the operating system documentation Web site, provide additional information about supported application codesets and terminal codesets (<http://www.tru64unix.compaq.com/docs/>).

2.3.2 Command Line Editing That Supports Multibyte Characters

This section discusses how you enable and use command line editing when Asian language support is installed on your system.

When the terminal line discipline and terminal codeset characteristics are set appropriately for multibyte codesets, the `atty` driver handles command line editing appropriately for languages supported by those codesets. For example, when you enter the control sequence to delete a character (assuming you have defined the control sequence), the entire character is deleted, regardless of how many bytes it occupies. The character being erased can be either a single-byte character or a multibyte Asian character when both occur on the same command line.

Word deletion is also supported, even when words combine single-byte and multibyte characters. The `atty` driver accepts single-byte space characters, 2-byte space characters (if applicable to the terminal code setting), or tab characters as word delimiters.

The `erase` and `werase` options of the `stty` command line let you define the control sequence for character and word deletion. For example:

```
% stty erase Ctrl/h
% stty werase Ctrl/j
```

This example specifies that `Ctrl/h` deletes the character preceding the cursor and `Ctrl/j` deletes the word preceding the cursor.

History mode is a mode of command line editing that allows you to recall and optionally modify a command entered previously. The history mode implementation discussed here is one that is customized for Japanese, Chinese, and Korean input and supported only for the BSD terminal driver. Table 2-3 specifies the `stty` options that enable or disable history mode editing for these languages. See Table 2-6 for `stty` options that control history mode for the Thai language.

Table 2-3: The `stty` Options Used with `atty` to Enable and Disable History Mode

stty Option	Description
<code>history key</code>	Sets the toggle key for the history mechanism and enables it.
<code>-history</code>	Disables the history mechanism.

The `atty` driver can maintain a history of up to 32 commands, each with a maximum length of 127 characters. Table 2-4 describes the commands you can use to edit command lines after entering the `history key`.

Table 2–4: Command Line Editing in History Mode

Command/Key	Description
Ctrl/a	Move to the beginning of the line.
Ctrl/d	Delete the character under the cursor.
Ctrl/e	Move to the end of the line.
Up-arrow	Recall the previous command line in the history list.
Down-arrow	Recall the next command in the history list.
Left-arrow	Move the cursor left by one character.
Right-arrow	Move the cursor right by one character.
<i>erase_sequence</i>	Delete the character preceding the cursor.
<i>werase_sequence</i>	Delete the word preceding the cursor.

In Table 2–4, the *erase_sequence* and *werase_sequence* command keys indicate the control sequences defined by the `stty` options `-erase` and `-werase`, respectively.

When editing a command line in history mode, you insert characters as follows:

1. Press the arrow keys to move the cursor to the position immediately to the right of the point where you want to insert characters.
2. Enter the characters you want to insert.

If you enter the control characters that represent “kill,” “interrupt,” or “suspend,” the `tty` driver breaks out of history mode and cancels the command line being edited.

2.3.3 Kana-Kanji Conversion: Customization of Japanese Input Options

In the Japanese language, a particular language element, such as a vowel, can be represented by more than one character. These characters can have both phonetic and ideographic variants; furthermore, the phonetic character variants can print in either 2-column or single-column width. The different classes of characters, listed in the following table, require different input schemes:

Character Class	Description
Kanji	Ideographic

Character Class	Description
Hiragana	Phonetic
Katakana	Phonetic Katakana characters exist in full width (2-column) and half width (single-column) formats. The single-column format of Katakana is referred to as Hankaku.

During a single session, a Japanese user can work with Kanji, Hiragana, and Katakana characters in various combinations. The user therefore must be able to customize terminal input mode to suit the character being entered. When the input device is a JIS terminal rather than a workstation, the user must adjust line discipline and terminal code settings in the software to match hardware capabilities (for example, whether the terminal uses 7-bit or 8-bit encoding).

The `tty` driver supports a mechanism known as Kana-Kanji conversion. This term refers to the conversion between phonetic and ideographic characters and the support for keyboard entry sequences that make Japanese character selection more efficient for the user. You use the `stty` command to enable or disable the Kana-Kanji conversion method and other aspects of Japanese input support. The `stty` options that support Japanese input are described in Table 2–5 and, unless noted otherwise, are used in conjunction with the `jdec` option. For example, the following command sets the terminal line discipline to support Japanese character encoding and also enables Kana-Kanji conversion:

```
% stty jdec ikk
```

Table 2–5: The `stty` Options to Enable and Customize Japanese Input

stty Option	Description
<code>clause mode</code>	Sets the character attribute for marking a clause that results from Kana-Kanji conversion. The <code>mode</code> argument can be bold, underline, reverse, or none.
<code>esc.alw</code>	Changes the terminal state to “shift out” whenever a newline character is output. This option applies only when the <code>tcode</code> (terminal code) <code>stty</code> option is set to <code>jis7</code> or <code>jis8</code> .
<code>-esc.alw</code>	Does not change the current terminal state when a newline character is output. This option applies only when the <code>tcode</code> (terminal code) option is set to <code>jis7</code> or <code>jis8</code> .

Table 2–5: The stty Options to Enable and Customize Japanese Input (cont.)

stty Option	Description
<code>henkan mode</code>	<p>Sets the character attribute for marking a Henkan, or conversion, region that results from Kana-Kanji conversion. The <i>mode</i> argument can be <code>bold</code>, <code>underline</code>, <code>reverse</code>, or <code>none</code>.</p>
<code>ikk</code>	<p>Enables the Japanese input method and spawns the Kana-Kanji conversion daemon, <code>kkcd</code>, if it does not already exist. With the BSD terminal driver in <code>cbreak</code> mode, you must use the <code>jx</code> option before using the <code>ikk</code> option to enable the input method. With the STREAMS terminal driver, you must use the <code>jinkey</code> option before using the <code>ikk</code> option.</p> <p>By default, keymap information is taken from (in highest to lowest priority order):</p> <ol style="list-style-type: none"> 1. The file specified for the <code>kkseq</code> option of the <code>stty</code> command 2. The file defined for the <code>JSYKSEQ</code> environment variable 3. The <code>\$HOME/.jsykkseq</code> file <p>System default keymap files for the Japanese input method reside in the <code>/usr/i18n/skel/ja_JP</code> directory. Dictionaries used with the Japanese input method are taken from the following list (in highest to lowest priority order):</p> <ol style="list-style-type: none"> 1. The files defined for the <code>JSYTANGO</code>, <code>JSYKOJIN</code>, and <code>JSYLEARN</code> environment variables 2. The <code>/usr/i18n/jsy/jsy-tango.dic</code>, <code>\$HOME/jsykojin.dic</code>, and <code>\$HOME/jsylearn.dic</code> dictionary files
<code>-ikk</code>	<p>Disables the Japanese input method and kills the <code>kkcd</code> daemon.</p>

Table 2–5: The stty Options to Enable and Customize Japanese Input (cont.)

stty Option	Description
<i>jinkey sequence</i>	Defines the escape sequence to activate the extended Japanese input method used with the STREAMS terminal driver. The parameter for this option can be more than one character.
<i>imode mode</i>	Sets the mode for handling 8-bit code or Hankaku (single-column) Kana code when the terminal line discipline is set to <i>dec</i> . The <i>mode</i> argument can be one of the following keywords: <ul style="list-style-type: none">• <i>kanji</i>, where the 8-bit code is treated as encoding for Kanji• <i>hiragana</i>, where the 8-bit code is converted to 2-column Hiragana format• <i>katakana</i>, where the 8-bit code is converted to 2-column Katakana format• <i>hankaku</i>, where the 8-bit code is handled in Hankaku (1-column) Katakana format
<i>jx character</i>	Sets the toggle character for entering the extended, or <i>cbreak</i> , Kana-Kanji conversion mode used with the BSD terminal driver. Enter <i>cbreak</i> mode when working in utilities, such as <i>dbx</i> , that do not support the full range of Japanese input options.
<i>-jx</i>	Undefines the toggle character for entering the extended Kana-Kanji conversion mode.
<i>kin esc_sequence</i>	Sets the JIS Kanji “shift in” escape sequence for the JIS terminal.
<i>kkmap</i>	Displays the current keymap for Kana-Kanji conversion. The display is a traversal tree with a maximum of 15 characters for each key sequence.
<i>kkseq file</i>	Sets the Kana-Kanji conversion keymap file for the terminal (see also the table entry for the <i>ikk</i> option).

Table 2–5: The stty Options to Enable and Customize Japanese Input (cont.)

stty Option	Description
<code>knj.bsl</code>	Uses only one backspace to erase one Kanji character.
<code>-knj.bsl</code>	Uses two backspaces to erase one Kanji character.
<code>knj.sp</code>	Uses one 2-byte (zenkaku) space to blank out one Kanji character.
<code>-knj.sp</code>	Uses two ASCII spaces to blank out one Kanji character.
<code>kout <i>esc_sequence</i></code>	Sets the JIS Kanji “shift out” escape sequence for the JIS terminal.

All ASCII characters are represented by single-byte, 7-bit values in DEC Kanji. That is, the most significant bit (MSB) is always set to Off in the byte that represents an ASCII character. For Unicode conversion on Tru64 UNIX, the Japanese_UCS converters (`deckanji`, `sdeckanji`, `SJIS`, and `eucJP`) consider the lower 7-bit values to be ASCII.

2.3.4 Thai Terminal Support

Thai terminals are supported by the `ttty` terminal driver, which uses the BSD protocol. When using a Thai terminal, you must set the terminal line discipline to `tdec`, for example:

```
% stty tdec
```

Table 2–6 describes additional settings that the `stty` command supports for Thai terminals.

Table 2–6: The stty Options for Thai Terminals

stty Option	Description
<code>isc <i>mode</i></code>	Sets the mode for the Wototo Input Sequence Check (ISC). Valid mode values are 0 (pass through), 1 (basic), and 3 (strict). This option is not used in Thai history mode.

Table 2–6: The stty Options for Thai Terminals (cont.)

stty Option	Description
<code>reorder (-reorder)</code>	Enables or disables input reordering. This option is not used in Thai history mode.
<code>thistory key</code> <code>(-thistory)</code>	Enables or disables the Thai history mechanism. The specified key is the one used to toggle in and out of Thai history mode. Lines fewer than three characters in length are not recorded in the history list. The command line editing options you can use when Thai history mode is enabled are the same as those described in Table 2–4.

See `ttty(7)` for more information about these options.

Using Asian Language Support Enhancements for Motif Applications

In Motif environments, such as CDE, you use versions of fonts, codesets, servers, and applications that support features discussed in earlier chapters. This chapter provides more detail on using features that help support Asian languages. Topics include the following:

- Tuning the cache and unit size of the X display server for languages with ideographic characters (Section 3.1)
- Setting fonts for local language display (Section 3.2)
- Customizing an application window for local languages (Section 3.3)

For information on font rendering and on how to use font renderers for multibyte PostScript fonts, UDC fonts, and TrueType fonts, see the *Writing Software for the International Market* manual.

3.1 Tuning the X Server for Ideographic Languages

Asian languages have large ideographic character sets, so all characters needed for display are not loaded into memory at the same time. Instead, only as many characters as will fit in the memory cache are simultaneously loaded. When characters needed for display are not currently cached in memory, the least recently used font glyphs are removed from the cache to make room. The font-cache mechanism allows you to display ideographic text in multiple typefaces, font sizes, and font styles without increasing the amount of memory that systems must have to support ideographic languages.

The X server font-cache mechanism allows you to change the number of cache units and the size of these units to best accommodate the character sets used in displays. You will probably need to change the default values set for cache parameters to achieve the best performance from your system if it will display Asian language text. Consider the following criteria when deciding on the optimal values for font caching:

- The number of ideographic languages that you want to display
If you intend to work with several ideographic languages during the same CDE session, you need larger values for acceptable performance.

- The number of fonts that will be used simultaneously

Variation in font number and size is affected by the type of application you run. A desktop publishing application typically requires more fonts than other types of applications. A software development tool typically requires fewer fonts than other types of applications.

- The number of frequently used characters in the languages you want to display

In Asian languages, only a subset of characters are used frequently. The size of this subset varies from one language to another. For example, approximately 20,000 standard characters are supported for Taiwan but only 5,000 of those characters are used frequently. Estimates for the number of frequently used characters for other Asian countries are: People's Republic of China (3000), Korea (2000), and Japan (2000). Font-cache parameters are tuned to accommodate the subset of frequently used characters.

To change the cache size (which is the number of cache units) and the size of each cache unit, you must modify the X server configuration file `/usr/lib/X11/xdm/Xservers`. This file contains a line, similar to the following one, that starts the X server:

```
:0 local /usr/bin/X11/X
```

You can modify this line to add definitions for cache size and unit size. For example:

```
:0 local /usr/bin/X11/X -cs cache_size -cu unit_size
```

Table 3-1 describes the options that tune the font-cache mechanism.

Table 3–1: X Server Options for Tuning the Font-Cache Mechanism

stty Option	Description
<code>-cs cache_size</code>	<p>Defines the number of cache units.</p> <p>The minimum (and also default) value for this parameter is 1024. If you specify a cache size smaller than 1024, font caching is disabled. For one ideographic language, the recommended value is the lowest multiple of 1024 that accommodates the number of frequently used characters in that language.</p> <p>If a workstation displays multiple ideographic languages simultaneously, you must add together the values required for each language to get the minimum cache size. Specify an even larger value if you intend to run applications, such as desktop publishing software, that require multiple font styles and sizes for each ideographic character.</p>
<code>-cu unit_size</code>	<p>Defines the size of each cache unit.</p> <p>The minimum value for unit size is 31 bytes and the default value is 128 bytes. If you specify a value smaller than 31 bytes, the value has no effect. If a particular font requires more memory space than 128 bytes, the font-cache mechanism automatically allocates one or more additional units to store its glyphs.</p>

Note

Font caching applies only to uncompressed fonts in `pcf` format. Font caching is not applied to any compressed fonts or to fonts in `bdf` format. Because font caching cannot be used with compressed fonts, the 2-byte fonts for Asian languages are not installed in compressed format.

You can calculate cache unit size with the following formula:

```
unit_size = ((floor(ceil((double)WIDTH / 8.0) / 4.0)) + 1.0) * 4.0 * (double)HEIGHT
```

Consider the following calculation for a typical font size of 24x24:

```
unit_size in bytes =  
((floor(ceil((double) 24 / 8.0 / 4.0)) + 1.0) * 4.0 * (double) 24 = 96
```

For 34x34 fonts, the unit size calculation would yield 272 bytes.

Given that 96 bytes are needed to cache a 24x24 font glyph and 272 bytes are needed to cache a 34x34 font glyph, the default unit size of 128 has the following implications:

- For 24x24 fonts, each character needs one cache unit. If cache size is set to 4096, the cache can accommodate 4096 characters.
- For 34x34 fonts, each character needs three cache units. If cache size is set to 4096, the cache can accommodate 1365 characters.

Small fonts (those whose characters require a single 128-byte unit) are used to display ideographic characters. Therefore, you usually have to change only the cache size to achieve acceptable performance in text displays of languages with ideographic characters.

3.2 Setting Fonts for Display of Local Languages

The system on which you install language variant subsets is automatically updated with the fonts required to display text in the supported languages.

In CDE, applications access local language fonts through a font alias mechanism. The `/usr/dt/config/xfonts/i18n/{75,100}dpi/fonts.alias` files rather than files installed in the `/usr/dt/config/xfonts/locale-name/` areas are most critical for resolution of which fonts an application uses. This arrangement supports both a consistent session language and the ability to run an individual application in a language different from the session language.

X or Motif applications require non-ASCII fonts to display Chinese (and other, non-English) characters. Therefore, you must set the font path appropriately before starting an application that displays Chinese characters. An application can find Chinese fonts for codesets other than GBK or GB18030 in either of the following directories:

- `/usr/i18n/lib/X11/fonts/decwin/75dpi`
- `/usr/i18n/lib/X11/fonts/decwin/100dpi`

Chinese fonts for GBK and GB18030 are installed in `/usr/i18n/lib/X11/fonts/SChineseTT` (for TrueType font).

For CDE applications, the appropriate fonts are installed on the system or available through a font server. Thus, you do not need to enter manual commands to ensure that the appropriate fonts are found. You can use the `xset` command to check the font path or to add an appropriate font to the path as described in `xset(1X)`.

3.2.1 Accessing Local Language Fonts for Remote Displays

The system where local language subsets are installed may function as a client in a client/server display environment. In this case, the local language

fonts must also be available to the window managers for all the server systems where native language text is displayed.

To make local language fonts available for other server systems, you must do one of the following:

- Install fonts for other locales on the individual systems used for remote login to the system where language variant subsets are installed.
- Make the fonts known to the other systems through a font server.

You can use the `/usr/bin/X11/xlsfonts` command to determine which fonts are currently installed on your system.

See Appendix A for tables on fonts that describe, for each font set, the language, typeface code, style, sizes, and support for 75 or 100 dots-per-inch resolution.

3.3 Customizing a Terminal Emulation Window for Asian Languages

The following features and restrictions apply to terminal windows that you create when an Asian language is specified for the language setting:

- The language setting determines the additional menu items, push buttons, toggle switches, and text entry fields that may be available to you for customizing terminal window features.
- Terminal emulation always follows the selected language for your session if the terminal is invoked from the CDE Personal Applications Menu. If a terminal window is invoked from another terminal window where the `LANG` or `LC_ALL` variable has been set to the locale for another language, then the language of the new window changes. Setting locale in the parent window does not change the language of the parent window, only of child windows invoked from the parent window.
- For a language supported by an input method server, you must be sure the input server is connected to the terminal window where you input characters in that language. Otherwise, you cannot use the input method for character entry. The connection between a terminal window and an input server does not exist if one of the following occurs:
 - The terminal window was started before the input server started

At the start of a CDE session, the input server starts automatically when the session language is selected. For example, if Chinese is your session language, the input server for Chinese is automatically attached to terminal windows by default. However, if Chinese is your session language and you want to create a window to work in Korean,

you must manually start the input server for Korean (in addition to setting a Korean locale) before invoking the new terminal window.

- The input method server was killed for some reason

If the connection between a terminal window and the input method server was broken, you can start the input method server and then create another terminal window where you can use the input method.

A

Bitmap and Monospaced Fonts for Locales

This appendix contains the following tables on fonts that describe, for each font set, the language, typeface code, style, sizes, and support for 75 or 100 dots-per-inch resolution.

- Table A-1, bitmap fonts for Asian locales
- Table A-2, bitmap fonts for ISO8859-1 locales
- Table A-3, bitmap fonts for ISO8859-2 locales
- Table A-4, bitmap fonts for ISO8859-4 locales
- Table A-5, bitmap fonts for ISO8859-5 locales
- Table A-6, bitmap fonts for ISO8859-7 locales
- Table A-7, bitmap fonts for ISO8859-8 locales
- Table A-8, bitmap fonts for ISO8859-9 locales
- Table A-9, bitmap fonts for ISO8859-15 locales
- Table A-10, bitmap fonts for Unicode (UTF-8) locales

Table A-1: Bitmap Fonts for Asian Locales

Language	Typeface (Code)	Style	Sizes	75dpi	100dpi
Japanese	Gothic (ISO Latin-1)	Normal	8, 10, 12, 14, 18, 24	x	x
	Gothic (Kanji)	Normal	8, 10, 12, 14, 18, 24	x	x
	Gothic (Roman Kana)	Normal	8, 10, 12, 14, 18, 24	x	x
	kmenu (ISO Latin-1)	Normal	12	x	x
	kmenu (Roman Kana)	Normal	12	x	x
	Mincho (ISO Latin-1)	Normal	8, 10, 12, 14, 18, 24	x	x
	Mincho (Kanji)	Normal	8, 10, 12, 14, 18, 24	x	x
	Mincho (Roman Kana)	Normal	8, 10, 12, 14, 18, 24	x	x
	Screen (DECsuppl)	Normal	14, 18, 24	x	
	Screen (DECtech)	Normal	14, 18, 24	x	
	Screen (ISO Latin-1)	Normal	14, 18, 24	x	

Table A–1: Bitmap Fonts for Asian Locales (cont.)

Language	Typeface (Code)	Style	Sizes	75dpi	100dpi
	Screen (Kanji00)	Normal	10, 14, 16, 18, 24	x	
	Screen (Kanji11)	Normal	10, 14, 18, 24	x	
	Screen (Roman Kana)	Normal	10, 14, 18, 24	x	
Korean	Gothic (KSC 5601)	Normal	16, 24	x	
	Myungcho (KSC 5601)	Normal	16, 24, 32	x	
	Screen (KSC 5601)	Normal	18, 24	x	
	KS Roman (DEC KS Roman)	Normal	18, 24	x	
Simplified Chinese	FangSongTi (GB 2312)	Normal	24, 34	x	
	HeiTi (GB 2312)	Normal	16, 24, 34	x	
	KaiTi (GB 2312)	Normal	24, 34	x	
	Screen (GB 2312)	Normal	18, 24	x	
	SongTi (GB 2312)	Normal	16, 24, 34	x	
Traditional Chinese	Hei (CNS11643)	Normal	16, 24	x	
	Hei (DTSCS)	Normal	16, 24	x	
	Screen (CNS11643)	Normal	18, 24	x	
	Screen (DTSCS)	Normal	18, 24	x	
	Sung (CNS11643)	Normal	24, 32	x	
	Sung (DTSCS)	Normal	24, 32	x	
Thai	Screen (TIS 620–2533)	Normal	14, 18, 24	x	
Asia (Misc.)	Screen (DEC Ctrl)	Normal	14, 18, 24	x	
	Screen (DRCS)	Normal	18, 24	x	

The WLS installation subset contains standard bitmap screen fonts and scalable fonts. In general, complex Asian characters require fonts greater than size 16 and the bitmap fonts are designed for legibility at higher resolutions. Thus, you should first try the bitmap font for the desired font size before attempting to use the scalable fonts.

Table A–2: Bitmap Fonts for *.ISO8859-1 Locales

Language	Typeface	Style	Sizes	75dpi	100dpi
Western European languages, including Catalan, Dutch, English, Finnish, French, German, Icelandic, Italian, Norwegian, Portuguese, Spanish, and Swedish	Monotype Arial	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
	Monotype Arial Narrow	Bold-Italic	10, 12, 14, 18, 24, 36	x	x
		Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Book Antiqua	Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
		Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Bookman Old Style	Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
Normal		10, 12, 14, 18, 24, 36	x	x	
Italic		10, 12, 14, 18, 24, 36	x	x	

Table A–2: Bitmap Fonts for *.ISO8859-1 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
	Monotype Century Gothic	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Century Schoolbook	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Courier	Normal	8, 10, 12, 14, 18, 24, 36	x	x
		Italic	8, 10, 12, 14, 18, 24, 36	x	x
		Bold	8, 10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24, 36	x	x
	Monotype Corsiva	Normal	10, 12, 14, 18, 24, 36	x	x
	Monotype Times New Roman	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	DEC Terminal	Normal	14, 18	x	x
		Double-Width	14, 18	x	x

Table A–2: Bitmap Fonts for *.ISO8859-1 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Double-Width, Double-Height	28, 36	x	x
		Narrow	14, 18	x	x
		Double-Width, Narrow	14, 18	x	x
		Double-Width, Double-Height, Narrow	28, 36	x	x
		Bold	14, 18	x	x
		Double-Width, Bold	14, 18	x	x
		Double-Width, Double-Height, Bold	28, 36	x	x
		Narrow, Bold	14, 18	x	x
		Double-Width, Narrow, Bold	14, 18	x	x
		Double-Width, Double-Height, Narrow, Bold	28, 36	x	x

Table A–3: Bitmap Fonts for *.ISO8859-2 Locales

Language	Typeface	Style	Sizes	75dpi	100dpi
Czech, Hungarian, Polish, Slovak, Slovene	Monotype Arial	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x

Table A-3: Bitmap Fonts for *.ISO8859-2 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Arial Narrow	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Book Antiqua	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Bookman Old Style	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Century Gothic	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Century Schoolbook	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x

Table A-3: Bitmap Fonts for *.ISO8859-2 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Courier	Normal	8, 10, 12, 14, 18, 24, 36	x	x
		Italic	8, 10, 12, 14, 18, 24, 36	x	x
		Bold	8, 10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24, 36	x	x
	Monotype Corsiva	Normal	10, 12, 14, 18, 24, 36	x	x
	Monotype Times New Roman	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	DEC Terminal	Normal	14, 18	x	x
		Double-Width	14, 18	x	x
		Double-Width, Double-Height	28, 36	x	x
		Narrow	14, 18	x	x
		Double-Width, Narrow	14, 18	x	x
		Double-Width, Double-Height, Narrow	28, 36	x	x
		Bold	14, 18	x	x
		Double-Width, Bold	14, 18	x	x

Table A–3: Bitmap Fonts for *.ISO8859-2 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Double-Width, Double-Height, Bold	28, 36	x	x
		Narrow, Bold	14, 18	x	x
		Double-Width, Narrow, Bold	14, 18	x	x
		Double-Width, Double-Height, Narrow, Bold	28, 36	x	x

Table A–4: Bitmap Fonts for *.ISO8859-4 Locales

Language	Typeface	Style	Sizes	75dpi	100dpi
Lithuanian	Monotype Arial	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Arial Narrow	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Book Antiqua	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x

Table A-4: Bitmap Fonts for *.ISO8859-4 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
	Monotype Bookman Old Style	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Century Gothic	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Century Schoolbook	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Courier	Normal	8, 10, 12, 14, 18, 24, 36	x	x
		Italic	8, 10, 12, 14, 18, 24, 36	x	x
		Bold	8, 10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24, 36	x	x
	Monotype Corsiva	Normal	10, 12, 14, 18, 24, 36	x	x
	Monotype Times New Roman	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x

Table A–4: Bitmap Fonts for *.ISO8859-4 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	DEC Terminal	Normal	14, 18	x	x
		Double-Width	14, 18	x	x
		Double-Width, Double-Height	28, 36	x	x
		Narrow	14, 18	x	x
		Double-Width, Narrow	14, 18	x	x
		Double-Width, Double-Height, Narrow	28, 36	x	x
		Bold	14, 18	x	x
		Double-Width, Bold	14, 18	x	x
		Double-Width, Double-Height, Bold	28, 36	x	x
		Narrow, Bold	14, 18	x	x
		Double-Width, Narrow, Bold	14, 18	x	x
		Double-Width, Double-Height, Narrow, Bold	28, 36	x	x

Table A–5: Bitmap Fonts for *.ISO8859-5 Locales

Language	Typeface	Style	Sizes	75dpi	100dpi
Russian	Monotype Arial	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x

Table A-5: Bitmap Fonts for *.ISO8859-5 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Courier	Normal	8, 10, 12, 14, 18, 24, 36	x	x
		Italic	8, 10, 12, 14, 18, 24, 36	x	x
		Bold	8, 10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24, 36	x	x
	Monotype Nimrod	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Plantin	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Times New Roman	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	DEC Terminal	Normal	14, 18	x	x
		Double-Width	14, 18	x	x

Table A–5: Bitmap Fonts for *.ISO8859-5 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Double-Width, Double-Height	28, 36	x	x
		Narrow	14, 18	x	x
		Double-Width, Narrow	14, 18	x	x
		Double-Width, Double-Height, Narrow	28, 36	x	x
		Bold	14, 18	x	x
		Double-Width, Bold	14, 18	x	x
		Double-Width, Double-Height, Bold	28, 36	x	x
		Narrow, Bold	14, 18	x	x
		Double-Width, Narrow, Bold	14, 18	x	x
		Double-Width, Double-Height, Narrow, Bold	28, 36	x	x

Table A–6: Bitmap Fonts for *.ISO8859-7 Locales

Language	Typeface	Style	Sizes	75dpi	100dpi
Greek	Monotype Arial	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Courier	Normal	8, 10, 12, 14, 18, 24, 36	x	x
		Italic	8, 10, 12, 14, 18, 24, 36	x	x

Table A–6: Bitmap Fonts for *.ISO8859-7 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Bold	8, 10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24, 36	x	x
	Monotype Times New Roman	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	DEC Terminal	Normal	14, 18	x	x
		Double-Width	14, 18	x	x
		Double-Width, Double-Height	28, 36	x	x
		Narrow	14, 18	x	x
		Double-Width, Narrow	14, 18	x	x
		Double-Width, Double-Height, Narrow	28, 36	x	x
		Bold	14, 18	x	x
		Double-Width, Bold	14, 18	x	x
		Double-Width, Double-Height, Bold	28, 36	x	x
		Narrow, Bold	14, 18	x	x
		Double-Width, Narrow, Bold	14, 18	x	x
		Double-Width, Double-Height, Narrow, Bold	28, 36	x	x

Table A–7: Bitmap Fonts for *.ISO8859-8 Locales

Language	Typeface	Style	Sizes	75dpi	100dpi
Hebrew	DEC David	Normal	8, 10, 12, 14, 18, 24	x	x
		Italic	8, 10, 12, 14, 18, 24	x	x
		Bold	8, 10, 12, 14, 18, 24	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x
	DEC Frankruhl	Normal	8, 10, 12, 14, 18, 24	x	x
		Italic	8, 10, 12, 14, 18, 24	x	x
		Bold	8, 10, 12, 14, 18, 24	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x
	DEC Gam	Normal	8, 10, 12, 14, 18, 24	x	x
		Italic	8, 10, 12, 14, 18, 24	x	x
		Bold	8, 10, 12, 14, 18, 24	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x
	DEC menu	Normal	10, 12	x	x
	DEC Miriam	Normal	8, 10, 12, 14, 18, 24	x	x
		Italic	8, 10, 12, 14, 18, 24	x	x
		Bold	8, 10, 12, 14, 18, 24	x	x
Bold-Italic		8, 10, 12, 14, 18, 24	x	x	
DEC Miriam Fixed	Normal	8, 10, 12, 14, 18, 24	x	x	
	Italic	8, 10, 12, 14, 18, 24	x	x	

Table A-7: Bitmap Fonts for *.ISO8859-8 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Bold	8, 10, 12, 14, 18, 24	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x
	DEC Narkiss Tam	Normal	8, 10, 12, 14, 18, 24	x	x
		Italic	8, 10, 12, 14, 18, 24	x	x
		Bold	8, 10, 12, 14, 18, 24	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x
	DEC Terminal	Normal	14, 18	x	x
		Double-Width	14, 18	x	x
		Double-Width, Double-Height	28, 36	x	x
		Narrow	14, 18	x	x
		Double-Width, Narrow	14, 18	x	x
		Double-Width, Double-Height, Narrow	28, 36	x	x
		Bold	14, 18	x	x
		Double-Width, Bold	14, 18	x	x
		Double-Width, Double-Height, Bold	28, 36	x	x
		Narrow, Bold	14, 18	x	x
		Double-Width, Narrow, Bold	14, 18	x	x
		Double-Width, Double-Height, Narrow, Bold	28, 36	x	x

Table A–8: Bitmap Fonts for *.ISO8859-9 Locales

Language	Typeface	Style	Sizes	75dpi	100dpi
Turkish	Monotype Arial	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Arial Narrow	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Book Antiqua	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
Monotype Bookman Old Style	Normal	10, 12, 14, 18, 24, 36	x	x	
	Italic	10, 12, 14, 18, 24, 36	x	x	
	Bold	10, 12, 14, 18, 24, 36	x	x	
	Bold-Italic	10, 12, 14, 18, 24, 36	x	x	
Monotype Century Gothic	Normal	10, 12, 14, 18, 24, 36	x	x	
	Italic	10, 12, 14, 18, 24, 36	x	x	
	Bold	10, 12, 14, 18, 24, 36	x	x	

Table A–8: Bitmap Fonts for *.ISO8859-9 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Century Schoolbook	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	Monotype Courier	Normal	8, 10, 12, 14, 18, 24, 36	x	x
		Italic	8, 10, 12, 14, 18, 24, 36	x	x
		Bold	8, 10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24, 36	x	x
	Monotype Corsiva	Normal	10, 12, 14, 18, 24, 36	x	x
	Monotype Times New Roman	Normal	10, 12, 14, 18, 24, 36	x	x
		Italic	10, 12, 14, 18, 24, 36	x	x
		Bold	10, 12, 14, 18, 24, 36	x	x
		Bold-Italic	10, 12, 14, 18, 24, 36	x	x
	DEC Terminal	Normal	14, 18	x	x
		Double-Width	14, 18	x	x
		Double-Width, Double-Height	28, 36	x	x
		Narrow	14, 18	x	x
		Double-Width, Narrow	14, 18	x	x
		Double-Width, Double-Height, Narrow	28, 36	x	x

Table A–8: Bitmap Fonts for *.ISO8859-9 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Bold	14, 18	x	x
		Double-Width, Bold	14, 18	x	x
		Double-Width, Double-Height, Bold	28, 36	x	x
		Narrow, Bold	14, 18	x	x
		Double-Width, Narrow, Bold	14, 18	x	x
		Double-Width, Double-Height, Narrow, Bold	28, 36	x	x

Table A–9: Bitmap Fonts for *.ISO8859-15 Locales

Language	Typeface	Style	Sizes	75dpi	100dpi
Western European languages, including Catalan, Dutch, English, Finnish, French, German, Icelandic, Italian, Norwegian, Portuguese, Spanish, and Swedish ^a	Adobe Courier	Bold	8, 10, 12, 14, 18, 24	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x
		Medium	8, 10, 12, 14, 18, 24	x	x
		Medium-Italic	8, 10, 12, 14, 18, 24	x	x
	Adobe Helvetica	Bold	8, 10, 12, 14, 18, 24	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x

Table A–9: Bitmap Fonts for *.ISO8859-15 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Medium	8, 10, 12, 14, 18, 24	x	x
		Medium-Italic	8, 10, 12, 14, 18, 24	x	x
	Adobe ITC Avant Garde Gothic	Demi-Bold	8, 10, 12, 14, 18, 24	x	x
		Demi-Bold- Italic	8, 10, 12, 14, 18, 24	x	x
		Italic	8, 10, 12, 14, 18, 24	x	x
		Medium	8, 10, 12, 14, 18, 24		
	Adobe ITC Lubalin Graph	Demi-Bold	8, 10, 12, 14, 18, 24	x	x
		Demi-Bold- Italic	8, 10, 12, 14, 18, 24	x	x
		Italic	8, 10, 12, 14, 18, 24	x	x
		Medium	8, 10, 12, 14, 18, 24	x	x
	Adobe ITC Souvenir	Demi-Bold	8, 10, 12, 14, 18, 24	x	x
		Demi-Bold Italic	8, 10, 12, 14, 18, 24	x	
		Light	8, 10, 12, 14, 18, 24	x	x
		Light-Italic	8, 10, 12, 14, 18, 24	x	x
	Adobe New Century Schoolbook	Bold	8, 10, 12, 14, 18, 24	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x
		Medium	8, 10, 12, 14, 18, 24	x	x
		Medium-Italic	8, 10, 12, 14, 18, 24	x	x
	Adobe Times	Bold	8, 10, 12, 14, 18, 24	x	x

Table A–9: Bitmap Fonts for *.ISO8859-15 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x
		Medium	8, 10, 12, 14, 18, 24	x	x
		Medium-Italic	8, 10, 12, 14, 18, 24	x	x
	Adobe Utopia	Bold	10, 12, 14, 18, 24	x	x
		Bold-Italic	10, 12, 14, 18, 24	x	x
		Regular	10, 12, 14, 18, 24	x	x
		Regular-Italic	10, 12, 14, 18, 24	x	x
	B&H Lucida	Bold	8, 10, 12, 14, 18, 24	x	x
		Bold-Italic	8, 10, 12, 14, 18, 24	x	x
		Medium	8, 10, 12, 14, 18, 24	x	x
		Medium-Italic	8, 10, 12, 14, 18, 24	x	x
	B&H Lucida Bright	Demi-Bold	8, 10, 12, 14, 18, 24	x	x
		Demi-Bold Italic	8, 10, 12, 14, 18, 24	x	x
		Medium	8, 10, 12, 14, 18, 24	x	x
		Medium-Italic	8, 10, 12, 14, 18, 24	x	x
	B&H Lucida Typewriter	Bold	8, 10, 12, 14, 18, 24	x	x
		Medium	8, 10, 12, 14, 18, 24	x	x
	Bitstream Charter	Medium	8, 10, 12, 14, 18, 24	x	x
		Medium-Italic	8, 10, 12, 14, 18, 24	x	x
	Bitstream Terminal	Bold	14, 18, 28, 36 18, 36	 x	x

Table A–9: Bitmap Fonts for *.ISO8859-15 Locales (cont.)

Language	Typeface	Style	Sizes	75dpi	100dpi
		Medium	14,18, 28, 36		x
			18, 36	x	
	DEC Terminal	Bold	10, 20		x
			14, 28	x	
		Medium	10, 20	x	x
	DEC VT330	Bold	14, 28		x
		Medium	14, 28		x

^a The operating system uses bitmap fonts for Latin-9 (ISO 8859–15) to support the same languages as supported for Latin-1 (ISO 8859–1). The Latin-9 fonts include a glyph for the euro character and are therefore used with the *.UTF-8 locales supporting Western European languages.

Table A–10: Bitmap Fonts for *.UTF-8 Locales

Language	Typeface	Style	Sizes	75dpi	100dpi
^a	ETL Fixed	Medium	14, 16, 24	x	x

^a The ETL public domain fonts support about 1000 characters of the Universal Character Set (UCS). Additional composite fonts for UTF-8 locales are created by the `libfr_FGC` font renderer from fonts that are available for the ISO 8859–1 and ISO 8859–15 codesets. Neither the ETL fonts nor those for Latin-1 and Latin-9 encoding support Chinese, Japanese, or Korean characters. Characters in these Asian languages are supported by converting UTF-8 encoding to an encoding for which fonts are available.

In addition to the bitmap fonts described in these tables, the operating system provides two sets of monospaced fonts: a 16x18 pixel set and a 24x24 pixel set. These monospaced fonts support UTF-8 locales with Latin and ISO/IEC 10646 encoding. The bitmap fonts have the following CDE font aliases:

```
-dt-interface-***-***-***-***-***-***-iso8859-1|2|3|4|5|7|8|9|15@mono
-dt-interface-***-***-***-***-***-***-iso10646-1@mono
```


B

Phrase Utility

When entering Chinese or Korean text, you have the option of entering individual characters and words or a string that identifies a phrase. The operating system supports phrase input in the following ways:

- The `dxim` Input Server is a multilingual input server with a menu interface that allows users to activate and use simplified Chinese, traditional Chinese, Korean, and Phrase input methods. The Phrase Input Server provided by `dxim` is different from the operating system Phrase Utility described in this appendix. The `dxim` Phrase Input Server uses a different database and a different user interface. See `dxim(5)` and the `dxim` Input Server online help for a description of the `dxim` Phrase Input Server.
- The `dxhanyuim`, `dxhanziim`, and `dxhangulim` Input Servers (traditional and simplified Chinese and Korean, respectively) each provide a Phrase Input Method that uses the database created by the operating system Phrase Utility. See `dxhanziim(5)`, `dxhanyuim(5)`, `dxhangulim(5)`, the *Technical Reference for Using Chinese Features* and the *Technical Reference for Using Korean Features* for information on these input servers and their use of the Phrase Utility.

This appendix describes the operating system Phrase Utility used by `dxhanyuim`, `dxhanziim`, and `dxhangulim` and includes information on creating and using a phrase database. See the `dxim` Input Server online help for information on creating and using a phrase database with the `dxim` Phrase Input Method.

The Chinese and Korean Input Servers support phrase input by means of a phrase database and one of the following:

- The Software Phrase Input Method (SIM) service

The SIM service, which is enabled through the `-adec` option of the `stty` command, extends support of phrase input to traditional and simplified Chinese terminals. The SIM service loads phrases dynamically to the terminal; therefore, the size of the phrase database is not limited by the memory restrictions of terminal hardware. When using a terminal supported by the SIM service, you press a user-defined key sequence to toggle in and out of phrase input mode. Entering phrase input mode shifts the site of user input to the 26th line of the terminal screen, where you are prompted to enter phrase codes.

- The phrase input mechanism available in the desktop environment
Because terminal emulation windows do not implement the 26th line of a terminal screen, the SIM service does not work correctly in these windows. In a windows desktop environment, phrase input, along with other kinds of input methods, is supported by the input method server for the Chinese and Korean languages. Therefore, you enter phrases by invoking the Input Method application and selecting the phrase item.

Using the Chinese and Korean input method servers, you can enter individual phrases and their codes for conversion and reuse as described in the Technical Reference manuals for those languages. You can also use the Phrase Utility to create and maintain a phrase database as described in this appendix.

Table B–1 lists and describes basic terms associated with phrase input.

Table B–1: Phrase Input Definitions

Term	Description
Phrase	The string for the phrase that you want to retrieve. Each phrase is a string of any characters in the codeset of the current locale and can be a maximum of 80 bytes in length.
Phrase code	The keyword that you enter to retrieve a phrase. Each phrase code is a string of up to 8 ASCII alphanumeric characters.
Class	A group of logically related phrases. Each class has an identifier that is a string of up to 8 ASCII characters.
Database	<p>A set of two files: the phrase data file <code>phrase.dat</code> and the class data file <code>class.dat</code>. If a phrase database is moved from one directory to another, the two data files must be moved together.</p> <p>There are two types of phrase databases: system and user. The system database is shared by all users on the system and is maintained by the system administrator. User databases are defined and maintained by individual users.</p> <p>Pathnames for the system and user phrase database directories are set in the <code>/var/i18n/conf/cp_dirs</code> file. By default, this file sets the pathname for the system phrase database directory to be <code>/var/i18n/sim</code> and for the user phrase database directory to be <code>\$HOME/.sim</code>. See the <i>Writing Software for the International Market</i> manual for more information on the <code>cp_dirs</code> file.</p> <p>Phrase database files are locale specific and reside in locale directories subordinate to the default path. For example, an individual user might create and maintain the following sets of files to support two different locales:</p> <pre> \$HOME/.sim/zh_TW.big5/phrase.dat \$HOME/.sim/zh_TW.big5/class.dat \$HOME/.sim/zh_TW.dechanyu/phrase.dat \$HOME/.sim/zh_TW.dechanyu/class.dat </pre>

B.1 Enabling the SIM Service

Table B-2 describes the `stty` command options that enable and set certain characteristics for Chinese phrase input through traditional and simplified Chinese terminals. These options do not apply to terminal emulation windows, which use mechanisms other than SIM and support Chinese and Korean Input Server phrase input.

Table B-2: The `stty` Options Used for the SIM Service

stty Option	Description
<code>sim</code>	Enables the SIM service.
<code>-sim</code>	Disables the SIM service.
<code>simall</code>	Displays current SIM service settings.
<code>simclass class</code>	Sets the current class name for locating the appropriate phrase in the phrase database. Classes identify subsets of information in the phrase database and are defined by using the Phrase Utility.
<code>simdb path</code>	Sets the path for the phrase database.
<code>simkey key</code>	Sets the toggle key for entering phrase input mode.
<code>simmode dmode</code>	Sets the display location for the phrase. The two supported settings are <code>offspot</code> (the default) and <code>onspot</code> . Use <code>onspot</code> with terminal emulators, such as <code>xterm</code> and <code>dterm</code> , that do not support the 26th display line.

B.2 Creating and Maintaining a Phrase Database

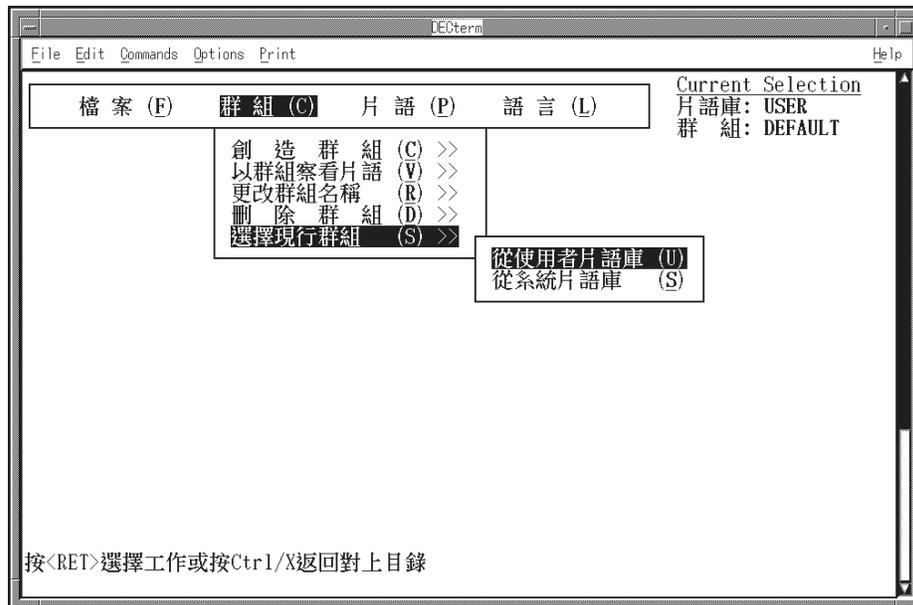
You can create or maintain a phrase database by using the Phrase Utility. On workstations, you invoke this utility with the following command:

```
% phrase
```

If you are a nonprivileged user, the `phrase` command assumes that you are using a private phrase database. If you are superuser, the `phrase` command assumes that you are using the system-wide phrase database. You can change these defaults by using the Phrase Utility Menu interface.

On startup, the Phrase Utility displays a full-screen, menu-driven interface. (Figure B-1 illustrates the display as it would appear on an Asian language system.)

Figure B-1: User Interface Screen of the Phrase Utility



The Phrase Utility is a curses application. To navigate the Phrase Utility user interface, use the following guidelines:

- To select a menu and menu items, use the arrow keys.
- To activate the selected menu or menu item, press either the Return key or the space bar.
- To select and activate in one operation, press the key for the underlined letter in the name of a menu or menu item, depending on your current level in the menu hierarchy.
- To return to a higher level of the menu hierarchy without activating a selection, press Ctrl/x.

If you press Ctrl/x when a menu is not activated, the Phrase Utility exits.

The Phrase Utility screen includes the following:

- A menu bar (upper-left corner of the screen)
- An area that specifies the current phrase database and class (to the right of the menu bar)
- Two lines for warning and informational messages (bottom of screen)
- A large area for menu expansion and user dialog (center of screen)

The following list describes the Phrase Utility menus and the operations they perform. The list is followed by guidelines and restrictions on the use of these menus.

- File Menu
 - Override the default path for the phrase database with which you want to work.
 - Exit from the Phrase Utility and save any changes made to the database.
- Class Menu
 - Create a class.
 - View phrases in the selected class.
 - Rename a class.
 - Delete a class.
 - Choose (change) the current class.
- Phrase Menu
 - Create a phrase within the selected class. If you do not explicitly choose a class, the Default class is assumed.
 - Modify a phrase.
 - Delete a phrase.
- Language Menu
 - Choose the language in which screen text and messages appear.

The following guidelines and restrictions apply to the use of the Phrase Utility:

- Creating and maintaining phrases
 - Phrases are maintained within the context of a phrase class. If you do not choose a class, the phrase is assumed to be in class Default. Otherwise, the phrase applies to the last class name you chose.
 - When you choose options that manipulate phrase definitions, a two-part window appears. The left side of the window displays phrase codes, while the right side of the window displays phrases. You enter phrase names and definitions in an area below the two-part display window.
 - Choose your phrase name carefully. The phrase name is the code that the system uses to invoke the phrase. After you define a phrase name, you cannot modify it. To redefine a phrase name, you must delete and reenter the entire phrase definition. Phrase names must

be unique within a given class, but you can use the same phrase name in different phrase classes.

- The phrase itself can contain up to 80 bytes of data, which correspond roughly to 80 columns on the screen. All 80 bytes of data appear in the user input area; however, the display window provides fewer than 80 columns to display the phrase. As a result, long phrase definitions are truncated at the right boundary of the display window. In such cases, the right angle bracket (>) appears in the rightmost position to indicate that the phrase definition contains more data. This truncation is a restriction of the display window and does not apply to the phrase when it is invoked.
- Creating and maintaining classes
 - Classes are created and maintained within the context of a particular database. If you have not explicitly specified a database, the class operation applies to your default database.
 - Class names must be unique within a database.
 - Creating a new class causes that class to be the selected class and then automatically invokes the function to create new phrases for the class.
 - There are no restrictions on the number of phrases in a class when phrases are retrieved through Asian terminals or through the input method window in the CDE environment.
- Using multiple phrase databases
 - Phrase databases are locale specific. You cannot invoke the Phrase Utility without setting the LANG environment variable to a locale; however, you can create phrase databases for any locale.

Be sure that the LANG environment variable is set to the locale you want to create phrases for before invoking the Phrase Utility. Otherwise, you will be working with (or creating) phrase databases for a locale different from the one you want.
 - You can copy phrase definitions to your private database from the system-wide database and from databases of other users (assuming their file protections allow you read access). If you choose to copy phrases from another user's database, you are prompted for the absolute path of the database from which you want to copy. If the specified database is accessible to you, all its phrase definitions are listed and you choose the ones you want to copy.
 - You must own a database to create, delete, or modify classes in that database.

- Nonprivileged users can perform write operations on their private databases. Only the superuser can perform write operations on the system-wide database.

B.3 Using a Phrase Database

How you use a phrase database depends on whether you are using the hardware input method or the SIM service.

B.3.1 Phrase Input Supported Through the SIM Service

You use the SIM service on the traditional and simplified Chinese terminals or in a terminal emulation window on a workstation.

Before you can use a phrase database through SIM service, you must use the `stty` command to perform the following tasks:

- Make sure your locale is set to one that supports the appropriate Asian codeset and that your terminal line discipline is set to `adec`
- Start the SIM service with the following command

```
% stty sim
```

- Define the key sequence for toggling in and out of phrase input mode

The following example sets this key sequence to be `Ctrl/b`:

```
% stty simkey Ctrl/b
```

Define a key sequence that you do not already use at the command line or in other applications. For example, do not define the key sequence to be `Ctrl/c` (abort operation) or `Ctrl/z` (suspend operation).

If you do not want to use phrases from the class `Default` or from your default phrase database, use the `stty` command to perform the following tasks:

- Specify the phrase class that the SIM service or specialized terminal software will use to interpret phrase codes. For example:

```
% stty simclass CORP
```

- Specify the database that specialized terminal software will access

The SIM service always searches your private phrase database first for a phrase name and, if the name is not found, then searches the system-wide phrase database. However, terminals that support the hardware phrase input method can load phrases from only one database at a time. Therefore, a nonprivileged user using the terminal hardware input method might enter the following command:

```
% stty simdb /var/i18n/sim
```

When the terminal setup is complete, you can perform the following actions to retrieve a phrase:

1. Press the key sequence specified for the `simkey` option of the `stty` command, for example `Ctrl/b`.
At the bottom of your screen, you are prompted to enter a phrase code.
2. Type the phrase code and press either the Return key or the space bar.
The phrase is returned to the screen or, if the phrase code was not found, an error message is displayed.

When you want to exit from phrase input mode, press the `simkey` key sequence again.

While in phrase input mode, the characters that you enter are subject to the following rules:

- Phrase codes must be uppercase letters or numbers. If you enter lowercase characters, they are converted to uppercase.
- A space or Return character that is entered when the phrase code buffer is empty is sent directly to the application from which you entered phrase input mode.

This behavior means that you do not have to exit from phrase mode to enter a space or new line between phrases.

- If you enter printable characters other than alphanumeric ones, the bell rings to signal that they are invalid characters for a phrase code.
- Control key sequences other than the one used to toggle in and out of phrase mode are sent directly to the application from which you entered phrase input mode.

This behavior means that control sequences such as `Ctrl/z` and `Ctrl/c` are handled as you would expect for the system command line, editor, or other application in which the phrases are being entered.

- Pressing a function or arrow key produces undefined results.

B.3.2 Phrase Input from the Input Options Application

When phrase input is supported by your language setting and the associated input method server is running, your desktop environment includes an Input Options window. Click on the Options button in this window to perform the following tasks:

- Choose the phrase database (user or system)
- Choose the phrase class within the database
- Start phrase input

To start phrase input, choose Input Method Customization from the Input Options Menu and, in the pop-up dialog box, choose Phrase.

Index

A

application

connecting to input server, 2–3

ASCII characters

in DEC Kanji, 2–15

Asian codeset

activate with Configure
International Software, 1–2

Asian environment

use of operating system, 2–1

Asian language

adjusting system variables, 1–11
input methods, 2–1
selecting locale for, 1–4
table of bitmap fonts, A–1
terminal emulation, 3–5
tuning X server, 3–1
UDC databases, 1–11

Asian language support

command line editing, 2–9

Asian printer

lprsetup command, 1–18

Asian terminal driver

atty, 2–6
character deletion, 2–10
defining with Configure
International Software, 1–2
history function, 2–10
mail message codeset conversion,
1–28
word deletion, 2–10

Asian tty settings

defining with Configure
International Software, 2–8
wwconfig command, 2–8

atty driver, 2–6

(*See also* Asian terminal driver)

B

bdf font format

restrictions when tuning X server
cache, 3–3

bitmap fonts

Asian languages, A–1
Eastern European languages, A–5
European languages, A–18
Greek language, A–12
Hebrew language, A–14
Lithuanian language, A–8
Russian language, A–10
Turkish language, A–16
UTF-8 locales, A–21
Western European languages, A–3

C

cache size

calculating, 3–3

CDE

Asian language support, 3–1
font alias files, 3–4
locale assumptions, 1–5
setting locale, 1–5
starting input method server, 2–5

CDE application

font files for, 3–4

characters

user-defined in Asian languages,
1–11

charset field in mail headers

- use by comsat server, 1–28
- Chinese**
 - bitmap font, A–1
 - simplified
 - dxhanziim, 2–4
 - dxim support, 2–3
 - input method server, 2–3
 - Postscript font, 1–24
 - Postscript font, 1–24
 - traditional
 - dxhanyuim, 2–4
 - dxim support, 2–3
 - input method server, 2–3
 - Postscript font, 1–24
- Chinese characters**
 - Motif requirement for non-ASCII fonts, 3–4
 - setting the font path, 3–4
- class.dat**
 - phrase database file, B–2
- client/server display environment**
 - font installation requirements, 3–4
- codeset**
 - application to terminal conversion, 2–9
 - conversion for data files, 1–29
 - conversion of, 1–28, 1–29
 - by mailx and MH, 1–26
 - for print jobs, 1–15
 - euro symbol support, 1–7
 - for mail interchange, 1–27
 - user application, 1–27
- command line editing**, 2–9
 - character entry under history mode, 2–11
 - commands for, 2–10
 - in history mode, 2–10
- Common Desktop Environment**
 - (*See* CDE)
- Compose key**, 1–5
- Compose sequences**
 - differences with mode switching, 1–6
- comsat server**

- codeset conversion, 1–28
- Configure International Software utility**, 1–2
 - Asian terminal driver support, 1–2
 - configuring an Asian environment, 2–1
 - defining Asian tty features, 2–8
 - defining Wnn, 1–2
 - removing fonts, 1–3
 - removing locales, 1–3
 - selecting input method, 1–3
 - setting login language, 1–3
 - system access to WLS tools, 1–2
 - user access to WLS tools, 1–2
 - viewing installed keymaps, 1–3
- csh.login**
 - Asian and Hebrew setting, 1–11

D

- data file**
 - converting codesets, 1–29
- dec option of stty command**, 2–7
- dense code locale**, 1–3
 - setting with Configure International Software, 1–3
- dtimsstart command**
 - setting preediting style, 2–2
- dtterm command**
 - setting preediting style, 2–2
- dxhangulim**, 2–4
 - (*See also* Korean Input Server)
 - Korean input method, 2–3
 - supported preediting styles, 2–4
- dxhanyuim**, 2–4
 - supported preediting styles, 2–4
 - traditional Chinese input method, 2–3
- dxhanziim**, 2–4
 - simplified Chinese input method, 2–3
 - supported preediting styles, 2–4
- dxim**
 - activating input methods, 2–4

- activating input servers, 2–4
- connecting application to, 2–3
- customizing input methods, 2–4
- customizing invocation keys, 2–4
- functions of, 2–4
- input method servers, 2–3
- Phrase input method, 2–3
- supported preediting styles, 2–3
- supporting multiple applications, 2–4

dxjim

- supported preediting styles, 2–4

dxkeyboard command, 1–5

- to set keyboard maps, 1–6

dxkeycaps command

- to edit keyboard mappings, 1–6

E

Eastern European languages

- table of bitmap fonts, A–5

Emacs editor

- multilingual, 1–30

euro currency symbol, 1–6

- displaying, 1–7

- entering, 1–7

- locales for assigning, 1–10

- locales that support, 1–7

- printing, 1–7

- table of supporting locales, 1–8

European languages

- table of bitmap fonts, A–18

excode definition

- in .mailrc file, 1–27

- in .mh_profile file, 1–27

EXCODE environment variable,

- 1–27

F

file command, 1–30

file name

- character set support, 1–31
- multibyte characters in, 1–31

font

- bitmap, 3–4

- displaying installed, 3–5

- for Chinese, A–1

- for Japanese, A–1

- for Korean, A–1

- for Thai, A–1

- CDE alias setup, 3–4

- compressed, 3–3

- exceeding printer memory, 1–21

- for codesets, 1–21

- for remote display, 3–4

- PostScript

- embedding in print job, 1–13

- PostScript outline, 1–21

- removing with Configure

- International Software, 1–3

- restrictions when tuning X server

- cache, 3–3

- setting style and size for local

- language printer, 1–15

- tuning X server cache for

- ideographic, 3–2

- use by print filter, 1–16

font cache

- optimal values for, 3–1

G

graphical application

- selecting locale for, 1–4

Greek language

- table of bitmap fonts, A–12

H

Hankaku characters

- in the Japanese language, 2–11

Hebrew language

- table of bitmap fonts, A–14

Hiragana characters
in the Japanese language, 2-11
history mode, 2-10
Hold Screen key, 1-30

I

i18n_profile
default setting, 1-11
i18nconfig
(See Configure International
Software utility)
I18NPATH
Asian language setting, 1-11
default setting, 1-11
iconv command, 1-29
ideographic characters
tuning X server cache for, 3-2
ideographic language
tuning X server, 3-1
input method, 2-1
phrase, B-2
preediting styles for, 2-1
setting with Configure International
Software, 1-3
Wnn, 1-2
input method server
connecting to terminal window, 3-5
connecting with an application, 2-3
dxim, 2-3
Japanese, 2-3, 2-4
Korean, 2-3, 2-4
running local, 2-3
running remote, 2-3
simplified Chinese, 2-3, 2-4
starting under CDE, 2-5
traditional Chinese, 2-3, 2-4
input mode state
toggling on and off, 2-5
input server
using phrase input, B-8
interactive style
(See preediting style)

internationalization
setup steps, 1-1
invocation key
customizing with dxim, 2-4
ISO-2022-JP, 1-27
ISO8859-1 codeset
PostScript font, 1-21
ISO8859-15 codeset
PostScript fonts, 1-24
ISO8859-2 codeset
PostScript font, 1-21
ISO8859-5 codeset
PostScript fonts, 1-22
ISO8859-7 codeset
PostScript fonts, 1-22
ISO8859-8 codeset
PostScript fonts, 1-23
ISO8859-9 codeset
PostScript fonts, 1-23

J

Japanese
bitmap font, A-1
Hankaku characters, 2-11
Hiragana characters, 2-11
input method server, 2-3
Kana-Kanji characters, 2-11
Kanji characters, 2-11
Katakana characters, 2-11
PostScript fonts, 1-25
Japanese character entry
customizing terminal for, 2-12
Japanese characters
multiple types in a single session,
2-12
Japanese Input Server, 2-4
jdec option of stty command, 2-7
caution on kernel debugging, 2-7
jfile command, 1-30
JSYKKSEQ environment variable,
2-12

K

Kana-Kanji conversion, 2–11, 2–12
changing key map for, 2–12
dictionaries used with, 2–12
displaying key map for, 2–12

Kanji characters
in the Japanese language, 2–11

Katakana characters
in the Japanese language, 2–11

kb_indicator command, 2–5

kernel debugging
caution when using stty command, 2–7

kernel linking
defining with Configure
International Software, 1–2

keyboard
alternate function key, 1–10
determining character mapping, 1–6
determining state of mode switch, 2–5
dxkeyboard command, 1–6
Keyboard Indicator utility, 2–5
Keyboard Options, 1–5
selecting for different languages, 1–5
setting from the command line, 1–6
setting in CDE, 1–5
system login default, 1–6

keyboard command
changing system-default keyboard, 1–6

keyboard entry
converting with dxim, 2–3

Keyboard Indicator
erasing server status, 2–5
if language is Hebrew, 2–6
stopping, 2–5

Keyboard Manager

(*See* Keyboard Indicator)

keyboard map
example of choosing, 1–7
viewing with Configure
International Software, 1–3

keyboard map files, 1–3

keyboard mapping
editing, 1–6

keyboard setting
as a global attribute, 1–6

keyboard type
requirements for setting, 1–5

kkcd daemon, 2–12

Korean
bitmap font, A–1
input method server, 2–3
PostScript fonts, 1–25

Korean input methods
support with dxim, 2–3

Korean Input Server, 2–4
(*See also* dxhangulim)

L

LANG environment variable
and man command search path, 1–28
set locale with, 1–4
use by mailx and MH, 1–27

.lang_device_name file
and mail message codeset
conversion, 1–28

language support
Asian support location, 1–11
default location, 1–11

LC_MONETARY
assigning euro symbol, 1–10

Lithuanian language
table of bitmap fonts, A–8

locale
default POSIX (C), 1–4
default setting for, 1–4

- dense code, 1–3
- @euro, 1–10
- euro symbol support, 1–7
- for Asian languages, 1–4
- for graphical applications, 1–4
- removing with Configure
 - International Software, 1–3
- setting, 1–4
- setting with Configure International Software, 1–4
- switching between dense code and Unicode, 1–4
- Unicode, 1–3
- use by print filter for validation, 1–16
- login language**
 - setting default with Configure International Software, 1–3
- login operation**
 - keyboard setting for, 1–6
- lp command**
 - local language printer support, 1–21
- lpc command**
 - local language printer support, 1–21
- lpd printer daemon**
 - local language printer support, 1–21
- lpq command**
 - local language printer support, 1–21
- lpr command**
 - local language printer support, 1–21
- lprm command**
 - local language printer support, 1–21
- lprsetup command**
 - Asian printers, 1–18
- lpstat command**
 - local language printer support, 1–21

M

- mail header**
 - charset field, 1–27
 - charset field in, 1–27
- mail message**
 - codeset conversion of, 1–26
 - notification of incoming, 1–28
- mail-codesets file**
 - use by comsat server, 1–28
 - use by mailx and MH, 1–27, 1–28
- .mailrc file**
 - defining application codeset in, 1–27
 - defining mail interchange codeset in, 1–27
- mailx command, 1–26**
 - determining application codeset, 1–27
- man command, 1–28**
 - and reference page translations, 1–28
 - conversion of codesets, 1–29
- manpage**
 - (*See* reference page)
- MH (mail handler) system, 1–26**
- MH command**
 - determining application codeset, 1–27
- .mh_profile file**
 - defining application codeset in, 1–27
 - defining mail interchange codeset in, 1–27
- mode switch LED, 2–5**
- monospaced font**
 - in UTF-8 locales, A–21
- more command, 1–31**
- Motif application**
 - displaying Chinese characters, 3–4
- mule command, 1–30**
- multibyte characters**
 - command line editing, 2–9
 - deleting, 2–10

in file and user names, 1–31
setting terminal line discipline for,
2–6

N

native language environment

working in, 1–1

nroff command

print filter for Asian language
output, 1–14

O

Off-the-Spot

preediting style, 2–2

On-the-Spot

preediting style, 2–2

operating system

use in Asian environment, 2–1

Over-the-Spot

preediting style, 2–2

P

PATH

Asian language setting, 1–11
default setting, 1–11

pcf font format

restrictions when tuning X server
cache, 3–3

pcfof print filter, 1–12

phrase database, B–2

changing the default, B–3
classes in, B–2
defining classes for, B–6
defining phrases for, B–5
effect of locale setting on, B–6
files in, B–2
guidelines and restrictions for, B–5
maintaining, B–3
private, B–3

SIM search path, B–7

system-wide, B–3

using, B–7

phrase input

dxhangulim Input Server, B–1

dxhanyuim Input Server, B–1

dxhanziim Input Server, B–1

dxim Input Server, B–1

rules for, B–8

using with input server, B–8

phrase input method

customizing for Asian terminals,
B–3

in terminal emulation windows,
B–2

on terminals, B–1

support with dxim, 2–3

Phrase Utility

dxim differences, 2–3

guidelines for using, B–5

invoking, B–3

menu interface, B–3

menus, B–4

screen interface, B–4

phrase.dat

phrase database file, B–2

plocale command

and font selection, 1–16

use with country-specific printers,
1–17

use with non-default fonts, 1–17

PostScript font

embedding in print job, 1–13

for ISO8859–1 codeset, 1–21

for ISO8859–15 codeset, 1–24

for ISO8859–2 codeset, 1–21

for ISO8859–5 codeset, 1–22

for ISO8859–7 codeset, 1–22

for ISO8859–8 codeset, 1–23

for ISO8859–9 codeset, 1–23

for Japanese, 1–25

for Korean, 1–25

- for simplified Chinese, 1–24
- for Thai, 1–25
- for traditional Chinese, 1–24
- for Unicode codeset, 1–24
- support for low-resolution bitmap, 1–13
- support for outline, 1–13
- support for TrueType, 1–13

preediting style, 2–2

- none, 2–4
- Off-the-Spot, 2–2
- on dxhangulim, 2–4
- on dxhanyuim, 2–4
- on dxhanziim, 2–4
- on dxjim, 2–4
- On-the-Spot, 2–2
- Over-the-Spot, 2–2
- Root Window, 2–2
- setting, 2–2
- supported by dxim, 2–3

print filter

- for Asian-language text files, 1–14
- for local language printers, 1–14
- for PostScript files, 1–12, 1–14
- generic internationalized, 1–12
- lines on a page, 1–16
- pcfof, 1–12
- specifying SoftODL font and style, 1–17
- using print filter of validating text, 1–16
- wwpssof, 1–13

printcap file

- printer characteristics, 1–15
- symbols for local language printers, 1–15

printcap symbol

- ya symbol keywords, 1–16

printer, 1–21

- (*See also* print filter, printcap file)
- and nonspacing characters, 1–17
- setting up, 1–18

- support for local languages, 1–12

printer characteristics

- printcap file, 1–15

printer locale

- specifying, 1–17

profile component

- of the .mailrc file, 1–27
- of the .mh_profile file, 1–27

pseudo terminal driver

- defining protocol with Configure International Software, 1–2

R

reference page

- location of translated files, 1–28
- processing non-English, 1–28

remote display

- font requirements for Asian languages, 3–4

rlogin command, 1–30

Root Window

- preediting style, 2–2

Russian language

- table of bitmap fonts, A–10

S

sd symbol in /etc/printcap file, 1–15

sendmail utility

- configuring for 8-bit data, 1–26

server

- starting starting input method, 2–3

Session Manager option

- setting preediting style, 2–2

SIM

- enabling service, B–3
- Software Input Method, B–1
- stty command options, B–3

simplified Chinese input method

- support with dxim, 2–3

simplified Chinese Input Server, 2–4

(*See also dxhanziim*)

simplified Chinese PostScript fonts, 1–24

SoftODL database
 default pathname, 1–17
 pathname specification, 1–17

SoftODL font and style
 default style and size, 1–17
 pathname specification, 1–17

SoftODL service
 printer settings used by
 character replacement method, 1–15
 size of printer cache, 1–15

Software Input Method
 (*See SIM*)

stty command
 acode option, 2–9
 adec option, 2–7
 clause option, 2–12
 dec option, 2–7
 enhancements for Asian languages, 2–6
 erase option, 2–10
 esc.alw option, 2–12
 henkan option, 2–12
 history option, 2–10
 ikk options, 2–12
 imode option, 2–12
 isc options, 2–15
 jdec option, 2–7
 jinkey option, 2–12
 jx options, 2–12
 kin option, 2–12
 kkmap option, 2–12
 kkseq option, 2–12
 knj.bsl options, 2–12
 knj.sp options, 2–12
 kout option, 2–12
 locale setting, 2–7
 multibyte codeset support, 2–6

options for phrase databases, B–3, B–7

options for phrase input, B–3

reorder options, 2–15

setting terminal code, 2–9

sim option, B–3, B–7

simall option, B–3

simclass option, B–3, B–7

simdb option, B–3, B–7

simkey option, B–3, B–7

special character handling, 2–6

table of options for Japanese input, 2–12

tcode option, 2–9

tdec option, 2–7

thistory options, 2–15

werase option, 2–10

stty utility
 codeset conversion, 2–9
 enabling command line editing, 2–10
 enabling Japanese input method, 2–12
 enhancements for local languages, 2–6
 local language enhancements, 2–9
 terminal line discipline, 2–6

symbol support
 requirements for, 1–7

T

terminal
 converting to supported application codeset, 2–9
 phrase input on, B–1

terminal emulation
 for Asian language, 3–5

terminal interface features, 2–6

terminal line discipline
 displaying current setting, 2–8
 restoring default, 2–7

setting, 2–6

terminal window
input method server connection,
3–5

text file
print filters for Asian language,
1–14

text morphing
enabling, 1–17

Thai
bitmap font, A–1
enabling text morphing, 1–17
PostScript fonts, 1–25
printers and nonspacing characters,
1–17

Thai language
printing characters on
one-and-one-half lines, 1–17

Thai terminal, 2–15
setting line discipline to tdec, 2–15

traditional Chinese input methods
support with dxim, 2–3

traditional Chinese Input Server,
2–4
(*See also* dxhanyuim)

**traditional Chinese PostScript
fonts**, 1–24

tty terminal driver
Thai terminal support, 2–15

tty driver
enhancements for local languages,
2–6
Kana-Kanji conversion, 2–12

Turkish language
table of bitmap fonts, A–16

U

UDC database, 1–11

Unicode
PostScript fonts, 1–24

Unicode locale, 1–3
setting with Configure International
Software, 1–3

user name
multibyte characters in, 1–31

user-defined characters
SoftODL printer settings, 1–15

UTF-8 locale
monospaced fonts, A–21
table of bitmap fonts, A–21

UTX support options
add with Configure International
Software, 1–2

V

vi command, 1–31

W

Western European languages
table of bitmap fonts, A–3

WLS, 1–1
configuring, 1–2
installing language support, 1–1
post-installation language support,
1–1

WLS tools
setting access with Configure
International Software, 1–2

Wnn
defining with Configure
International Software, 1–2
Japanese input method, 1–2

Worldwide Language Support
(*See* WLS)

wwconfig command
defining Asian tty features, 2–8
deleting Asian tty settings, 2–8
listing tty setting, 2–8

wwpsof print filter, 1–13
affect of locale setting, 1–13
order of search for fonts, 1–13
printer memory for fonts, 1–21

X

X server

- cache performance, 3-1
- configuration file, 3-2
- formula for cache size, 3-3
- tuning cache parameters, 3-2

X server configuration file

- cs option, 3-2
- cu option, 3-2
- tuning font cache in, 3-2

xlsfonts command, 3-5

Y

ya symbol in /etc/printcap file,
1-15

yp symbol in /etc/printcap file,
1-15

ys symbol in /etc/printcap file,
1-15

yt symbol in /etc/printcap file,
1-15