

# Tru64 UNIX

---

## Technical Reference for Using Japanese Features

June 2001

**Product Version:** Tru64 UNIX Version 5.1A or higher

This guide describes how to set up your system to use the Japanese features of the Tru64 UNIX operating system and how to use those features.

---

© 2001 Hewlett-Packard Company

Microsoft®, Windows®, and Windows NT® are trademarks of Microsoft Corporation in the United States and other countries. Motif®, OSF/1®, UNIX®, and X/Open® are trademarks of The Open Group in the U.S. and/or other countries.

PostScript® is a registered trademark of Adobe Systems, Inc. TrueType™ is a trademark of Apple Computer, Inc.

All other product names mentioned herein may be the trademarks of their respective companies.

Ichitaro® is a registered trademark of Just System Corporation.

PrintXchange® is a registered trademark of Xerox Corporation.

VJE® is a registered trade mark of VACS Corporation.

WX3 is a trade name of A.I.SOFT, Inc.

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 Tru64 UNIX Japanese Features

1.1	Internationalization Features, Commands and Utilities .....	1-2
1.1.1	Internationalization Features .....	1-2
1.1.2	Codesets .....	1-2
1.1.3	Japanese Locale and Internationalization Commands and Utilities .....	1-2
1.2	Asian Language Processing tty Subsystem .....	1-3
1.2.1	Kana-Kanji Conversion by STREAMS tty .....	1-3
1.2.2	On-Demand Font Loading .....	1-3
1.2.3	Code Conversion .....	1-3
1.3	Wnn Kana-Kanji Conversion System .....	1-4
1.4	Mule Editor .....	1-4
1.5	FD Access Utility .....	1-4
1.6	Kana-Kanji Conversion .....	1-4
1.6.1	Kana-Kanji Conversion Dictionaries and Data File .....	1-4
1.6.2	Conversion Key Definition .....	1-5
1.7	Japanese Printer Filters .....	1-5
1.8	Japanese Advanced Printing Software .....	1-5
1.9	Kanji Code Conversion Utilities .....	1-5
1.10	Conversion Key Definition Library .....	1-5
1.11	User-Defined Character Creation Tools .....	1-6
1.12	Tools for Migration from System V MNLS to Tru64 UNIX .....	1-6
1.13	Japanese Input System .....	1-6
1.14	Japanese Reference Pages .....	1-6
1.15	Japanese Message Catalogs .....	1-6
1.16	Japanese Document Full-Text Searching .....	1-6
1.17	Japanese TrueType Fonts .....	1-7

### 2 Basic Setup of Japanese Environment

2.1	Installing the Japanese Environment .....	2-1
2.2	Setting Up the Internationalization Environment .....	2-3
2.3	Setting up the CDE Language .....	2-4
2.3.1	Setting the CDE Default Language .....	2-5

2.4	Setting Up the Japanese Locale .....	2-5
2.5	Setting Up the Japanese Terminal .....	2-7
2.5.1	Setting the Terminal and Application Codes .....	2-8
2.5.1.1	Setting the Terminal Codeset .....	2-9
2.5.1.2	Setting the Application Codes .....	2-9
2.5.2	Handling of 8-Bit Data .....	2-9
2.5.3	Checking the Current Terminal Settings .....	2-10
2.6	File Code Conversion .....	2-11
2.7	Message Catalogs .....	2-12
2.8	Setting Up the Japanese Reference Pages .....	2-13
<b>3</b>	<b>Setting Up the Japanese Input System</b>	
3.1	dxjim .....	3-1
3.1.1	Starting dxjim .....	3-1
3.1.2	Japanese Input with dxjim .....	3-1
3.2	VJE .....	3-2
3.3	WX3 .....	3-2
<b>4</b>	<b>Setting Up a Japanese Printer</b>	
4.1	Printer Codes .....	4-1
4.2	Setting Up a Japanese Printer .....	4-1
4.3	Setting Up an ESC/P Printer .....	4-3
4.4	Installing and Setting Up Advanced Printing Software .....	4-3
<b>5</b>	<b>Setting Up a Japanese Keyboard</b>	
5.1	Setting Up a CDE Keyboard .....	5-1
5.1.1	Kana Mode .....	5-2
5.1.2	Conversion Keys .....	5-2
5.1.3	Kana LED .....	5-2
5.2	Setting Up a Japanese Keyboard in Console or Single-User Mode .....	5-3
<b>6</b>	<b>Installing and Setting Up Japanese Document Full-Text Searching Tools</b>	
6.1	Installing the midoc Software .....	6-1
6.1.1	Securing the Expanded Destination of the Kit .....	6-2
6.1.2	Installing the midoc Software From the Kit .....	6-2
6.1.3	Setup After Installation .....	6-2
6.1.3.1	Copying the Icon Files .....	6-3

6.1.3.2	Copying the Perl Scripts .....	6-3
6.1.3.3	Editing the Map File .....	6-3
6.1.4	Executing the Search .....	6-4
6.1.5	Uninstalling the midoc Software .....	6-4
6.2	Installing the miman Software .....	6-5
6.2.1	Securing the Expanded Destination of the Kit .....	6-5
6.2.2	Installing the miman Software from the Kit .....	6-5
6.2.3	Setup After Kit Expansion .....	6-6
6.2.4	Uninstalling the miman Software and Other Notes .....	6-7

## 7 Japanese Input with dxjim

7.1	Outline of the Japanese Conversion Features .....	7-1
7.2	Using dxjim .....	7-2
7.2.1	Setting Up dxjim .....	7-3
7.2.2	Input Styles .....	7-4
7.2.3	Using dxjim Resources .....	7-4
7.3	Japanese Input Conversion Keys .....	7-7
7.3.1	Changing the Conversion Key Definition .....	7-14
7.4	Conversion Rules .....	7-14
7.4.1	Romaji-to-Kana Conversion Correspondence Table .....	7-15
7.4.2	Special-Character Conversion Correspondence Table .....	7-18
7.4.3	Symbol Conversion Correspondence Table .....	7-18
7.4.4	Code Input .....	7-20
7.5	Using the Keyboard Kana Mode .....	7-21
7.6	Supporting Japanese Keyboards Equipped with Conversion Keys .....	7-21

## A Character Codes

A.1	Character Sets .....	A-1
A.1.1	7-Bit JIS Kanji Codeset .....	A-3
A.1.2	8-Bit JIS Kanji Codeset .....	A-4
A.1.3	Japanese EUC Codeset .....	A-5
A.1.4	DEC Kanji Codeset .....	A-6
A.1.5	Super DEC Kanji Codeset .....	A-7
A.1.6	Shift JIS Codeset .....	A-9
A.1.7	UTF-8 Codeset .....	A-10

## B Contents of Japanese Locale Definitions

B.1	CHARMAP .....	B-1
B.2	LC_TYPE .....	B-1

B.3	LC_COLLATE .....	B-5
B.4	LC_MESSAGES .....	B-5
B.5	LC_MONETARY .....	B-6
B.6	LC_NUMERIC .....	B-6
B.7	LC_TIME .....	B-7

## Index

### Examples

4-1	Sample /etc/printcap File .....	4-2
6-1	Accessing a CD-ROM Mounted Locally .....	6-4
6-2	Accessing a CD-ROM Mounted on the Web Server .....	6-4

### Figures

2-1	Software Subsets Required to Use the Japanese Environment	2-2
2-2	Terminal Driver (Asian tty) .....	2-8

### Tables

2-1	Japanese Locales That Can Be Selected for the CDE Language Setting Option .....	2-4
2-2	Japanese Locale Names Available for Tru64 UNIX .....	2-6
2-3	Locale Databases .....	2-6
2-4	Available Category Names .....	2-7
2-5	Possible Codeset Names for the Terminal Codesets .....	2-9
2-6	Possible Codeset Names for the Application Codesets .....	2-9
2-7	Possible Codeset Names for use with iconv .....	2-11
2-8	Possible Codeset Names for use with iconv (for Machines Not Manufactured by HP) .....	2-12
2-9	Directories Where Japanese Message Catalogs Exist .....	2-13
2-10	Reference Page Section Numbers .....	2-13
7-1	Japanese Input Conversion Features .....	7-1
7-2	JVMS Conversion Key Definition .....	7-9
7-3	EVEJ Conversion Key Definition .....	7-10
7-4	KEYPAD Conversion Key Definition .....	7-11
7-5	TARO Conversion Key Definition .....	7-12
7-6	CTRL Conversion Key Definition .....	7-13
7-7	Conversion Key Definition of a Japanese Keyboard Equipped with the 変換 (HENKAN) Keys .....	7-22
A-1	Classification of Character Sets Based on the JIS .....	A-1

A-2	ASCII Versus JIS X 0201 LH .....	A-2
A-3	Character Sets Available for Each of the Codesets .....	A-2
A-4	7-Bit JIS Codeset Character Codes .....	A-3
A-5	8-Bit JIS Codeset Character Codes .....	A-4
A-6	Japanese EUC Codeset Character Codes .....	A-6
A-7	DEC Kanji Character Set 1983 .....	A-7
A-8	DEC Kanji Codeset Character Codes .....	A-7
A-9	Super DEC Kanji Codeset Character Codes .....	A-8
A-10	Super DEC Kanji User-Defined Fields .....	A-8
A-11	Encoding of Shift JIS Codes .....	A-9
A-12	Correspondence Between Kanji and User-Defined Characters and Ku-Ten Numbers .....	A-10
A-13	UCS-4 Encoding Ranges and UTF-8 Bit Assignments .....	A-11
B-1	Japanese Codeset Attributes .....	B-1
B-2	Standard Character Classes .....	B-2
B-3	Extended Character Classes .....	B-3
B-4	LC_MESSAGES Definitions .....	B-6
B-5	Definitions of LC_MONETARY .....	B-6
B-6	Definitions of LC_NUMERIC .....	B-7





---

## About This Manual

This guide describes how to set up and use the Japanese features of the HP Tru64 UNIX operating system. It also summarizes related information such as printers, keyboards, and character codes. For details about the individual commands described herein, refer to the reference pages for the commands.

### Audience

This guide is intended for users of the Japanese variant of the Tru64 UNIX operating system.

### Changes

Removed references to obsolete keyboards – LK421.

### Organization

This manual is organized as follows:

<i>Chapter 1</i>	Outlines the Tru64 UNIX Japanese features
<i>Chapter 2</i>	Describes the basic setup required to use the Tru64 UNIX Japanese features
<i>Chapter 3</i>	Describes how to setup the Tru64 UNIX input system
<i>Chapter 4</i>	Describes how to setup Tru64 UNIX Japanese printers
<i>Chapter 5</i>	Describes how to setup the Tru64 UNIX Japanese keyboard
<i>Chapter 6</i>	Describes how to install and set up the full-text search features, midoc and miman
<i>Chapter 7</i>	Describes the Tru64 UNIX standard Japanese input system, dxjim
<i>Appendix A</i>	Describes the Japanese character, terminal, and application codes that can be processed by Tru64 UNIX
<i>Appendix B</i>	Describes the Japanese locales supported by Tru64 UNIX

### Related Documentation

The following Japanese documents are included on the Tru64 UNIX Japanese documentation CD-ROM:

- *DEC Input Server Library*
- *IMLIB Library Reference Manual*
- *User Key Definition User's Guide*
- *Kana-Kanji Conversion by STREAMS tty*
- *Kana-Kanji Conversion by Wnn*
- *User-Defined Characters*

In addition to the Japanese documentation CD-ROM, you can also find translated Tru64 UNIX documentation at the following web site:  
[http://digital.compaq.co.jp/unix/docs/jdoc\\_list.html](http://digital.compaq.co.jp/unix/docs/jdoc_list.html).

## Conventions

The following typographical conventions are used in this manual:

%	A percent sign represents the C shell system prompt.
\$	A dollar sign represents the system prompt for the Bourne and Korn shells.
#	A number sign represents the superuser prompt.
% <b>cat</b>	Boldface type in interactive examples indicates typed user input.
<i>file</i>	Italic (slanted) type indicates variable values, placeholders, and function argument names.
[   ]	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.
{   }	
...	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
:	A vertical ellipsis indicates that a portion of an example that would normally be present is not shown.

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



---

## Tru64 UNIX Japanese Features

The Tru64 UNIX operating system is an internationalized operating system that supports various languages, including Japanese. Most of the Japanese features in Tru64 UNIX are implemented in accordance with internationally standardized specifications. Tru64 UNIX offers the following Japanese features:

- Internationalization features and internationalization commands and utilities
- Asian Language Processing tty Subsystem
- Wnn Kana-Kanji Conversion system
- Mule editor
- FD access utility (FDIO)
- Kana-Kanji Conversion system and library
- Japanese printer filter
- Advanced Printing Software Japanese support kit
- Kanji code conversion utility (`iconv( )`)
- Conversion key definition library (IMLIB)
- User-defined characters (UDC) creation tools
- `dxjim` Japanese input system
- Japanese reference pages
- Japanese message catalogs
- Migration tools for internationalization features from System V MNLS to Tru64 UNIX

Tru64 UNIX also supports the following Japanese features. These features are not shipped as part of the general Tru64 UNIX distribution.

- VJE Japanese input system
- WX3 Japanese input system
- Japanese document full-text searching utilities (`miman` and `midoc`), included on the *Tru64 UNIX Japanese Extensions* CD-ROM

- Japanese TrueType fonts for display of Japanese characters in desktop applications, included on the *Tru64 UNIX Japanese Extensions* CD-ROM

Most Tru64 UNIX user interfaces can be displayed in Japanese.

The remainder of this chapter outlines all of the Japanese features.

## 1.1 Internationalization Features, Commands and Utilities

### 1.1.1 Internationalization Features

Tru64 UNIX internationalized utilities use the interface defined in the *X/Open CAE Screen System Interfaces and Headers, Issue 4 (XPG4)*. You select utilities and features from among multiple languages and codesets by making modifications to the locale settings. (A locale is a defined user environment reflecting a particular language, culture, and custom.)

### 1.1.2 Codesets

The Tru64 UNIX Japanese environment supports the following codesets:

- 7-Bit JIS Kanji
- 8-Bit JIS Kanji
- Japanese EUC
- Shift JIS (Japanese Industrial Standard)
- DEC Kanji
- Super DEC Kanji
- UTF-8

For details about the individual codesets, see Appendix A.

When specifying a Japanese locale, you can specify any of the codesets. Super DEC Kanji is designed to be a super set of DEC Kanji and Japanese EUC. The DEC Kanji character set is detailed in the book *Kanji Code Table* that is included in the Japanese subset of the Tru64 UNIX base system Installation Kit.

### 1.1.3 Japanese Locale and Internationalization Commands and Utilities

Tru64 UNIX supports the following Japanese locales:

- ja\_JP.eucJP or ja\_JP (Japanese EUC)
- ja\_JP.SJIS (Shift JIS)
- ja\_JP.deckanji (DEC Kanji)

- ja\_JP.sdeckanji (Super DEC Kanji)
- ja\_JP.UTF-8 (UTF-8)

By dynamically switching between these locales, Japanese characters can be processed using the internationalization commands and utilities.

## 1.2 Asian Language Processing tty Subsystem

The Asian Language Processing tty Subsystem supports the terminals connected to the system in asynchronous communications mode. For details about the supported terminals, refer to the Japanese Option Hardware section of the Japanese version of the Software Produce Description (SPD). In addition to terminals, the Tru64 UNIX Asian Language Processing tty Subsystem (hereafter, called Japanese tty) supports the following Japanese features:

- Kana-Kanji Conversion
- On-demand font loading
- Conversion between terminal codes and application codes

To use the Asian Language Processing tty Subsystem, you must have selected Japanese line discipline (jdec) using the `stty` command as follows:

```
% stty jdec
```

### 1.2.1 Kana-Kanji Conversion by STREAMS tty

Kana–Kanji conversion allows applications to invoke the standard input routine to input Japanese characters. For details, refer to the document, *Kana-Kanji Conversion by STREAMS tty*. Use the `stty` command to set up Kana-Kanji Conversion with the Japanese tty.

### 1.2.2 On-Demand Font Loading

On-Demand Font Loading (ODL) allows you to load a user-defined character font dynamically to the terminal or printer. Use the `stty` command to set up the ODL feature with the Japanese tty.

### 1.2.3 Code Conversion

The Japanese tty allows code conversion between the terminal and application codes.

Japanese tty code conversion supports the following codes for terminals:

- 7-bit JIS
- 8-bit JIS

- DEC Kanji 1983
- DEC Kanji 1978
- Japanese Extended Unix Code (EUC)
- Shift JIS
- UTF-8

Japanese tty code conversion supports the following codes for applications:

- Japanese EUC
- Shift JIS
- DEC Kanji
- Super DEC Kanji
- UTF-8

### 1.3 Wnn Kana-Kanji Conversion System

The Wnn Kana-Kanji Conversion System is a version of the free software, Wnn 4.109, transported to Japanese Tru64 UNIX. The Wnn Kana-Kanji Conversion System can be used as the front end for Kana-Kanji conversion and contains an extended feature from HP that enables access to a Kana-Kanji Conversion dictionary.

### 1.4 Mule Editor

Tru64 UNIX supports the multilingual editor Mule 2.3 (free software), which offers Kana-Kanji conversion input and other Japanese features.

### 1.5 FD Access Utility

The FD Access Utility (FDIO) is a utility that you use to access the floppy disk drive built in, or connected to, the Alpha system. FDIO offers read/write capabilities to a floppy disk formatted with MS-DOS and, in the text mode, automatically converts Kanji and line-feed codes during transfer or display of a file.

## 1.6 Kana-Kanji Conversion

### 1.6.1 Kana-Kanji Conversion Dictionaries and Data File

The dictionaries used for Kana-Kanji conversion include dictionaries of proper nouns, such as names of persons and places, as well as a general Japanese dictionary. The dictionaries contain approximately 100,000 words. In addition to a system-wide dictionary set, individual (or learning)



dictionaries are available for each user. You can use the individual dictionaries to save frequently-used information.

A phrase learning data file is also available that stores the results of phrase learning and phrase separation learning for each of user.

## 1.6.2 Conversion Key Definition

Conversion Key Definition (kkseq) defines a Kana-Kanji conversion key sequence for each user.

## 1.7 Japanese Printer Filters

Printer filters enable a Japanese printer to be available as a hardcopy terminal by means of a communications line. Thus, the contents of a Japanese document can be printed using a Tru64 UNIX printer spooler. An output filter program is also available that can print the contents of a Japanese document using a system-supported printer.

## 1.8 Japanese Advanced Printing Software

The Advanced Printing Software Japanese support kit is included on the Tru64 UNIX Worldwide Language Software distribution kit. Advanced Printing Software enables you to print Japanese text files (and Japanese PostScript files if the printer supports PostScript) on Japanese printers. Because the Japanese support kit depends on the standard Advanced Printing Software kit, you must install both the standard kit and the Japanese kit on the system. For details, see Chapter 4.

## 1.9 Kanji Code Conversion Utilities

The Kanji code conversion utility `iconv( )` performs code conversion between the Japanese EUC, Shift JIS, DEC Kanji, Super DEC Kanji, 7-bit JIS, and Unicode codesets. It also supports file code conversion from the IBM Kanji System Character, Fujitsu JEF Code, and Hitachi KEIS codesets. For details, see Section 2.6.

## 1.10 Conversion Key Definition Library

The Conversion Key Definition Library (IMLIB) is a programming library that is used to process definitions of the keys used for Japanese input. When you use IMLIB with an application, the definition process is based on the user-prepared Japanese input key definitions. With IMLIB key definitions, you can describe an input interface more complex than `kkseq`. IMLIB also provides a tool that enables conversion from a `kkseq` key definition file to the IMLIB key definition file. For customizing a key definition, refer to the *User*

*Key Definition User's Guide*. For using IMLIB to develop an application, refer to the *IMLIB Library Reference Manual*.

## 1.11 User-Defined Character Creation Tools

You can create, register, and display user-defined characters using the `cedit( )` utility, which helps you create a user-defined character, or the `cgen( )` utility, which converts the created font into a file used for font loading. For details, refer to the *User-Defined Characters*.

## 1.12 Tools for Migration from System V MNLS to Tru64 UNIX

The Tru64 UNIX operating system includes tools that aid you in converting applications created with the System V MNLS internationalization features into Tru64 UNIX versions. For example, tools are available to output information about migration of the System V MNLS features being used in programs and to convert message source files. For details, refer to the *Migration Guide for Internationalization Features from System V MNLS to Tru64 UNIX*.

## 1.13 Japanese Input System

The Tru64 UNIX operating system provides `dxjim` as the standard Japanese input server in the Common Desktop Environment (CDE). In addition to `dxjim`, Tru64 UNIX supports the PC Front End Processors, VJE and WX3.

See Chapter 3 for details about `dxjim`, VJE, and WX3.

## 1.14 Japanese Reference Pages

You can view Tru64 UNIX Japanese reference pages online.

## 1.15 Japanese Message Catalogs

Messages issued by major commands and utilities can be displayed in Japanese.

## 1.16 Japanese Document Full-Text Searching

Tru64 UNIX provides the Japanese document full-text searching tools (`midoc` and `miman`) for Japanese online documents and reference pages. `MitakeSearch` is used as a search engine for these tools. See Chapter 6 for information on installation and setup.

## 1.17 Japanese TrueType Fonts

Japanese TrueType fonts enable desktop applications, such as the Netscape Browser, to display Japanese characters. The TrueType font kit is included on the *Tru64 UNIX Japanese Extensions* CD-ROM and contains the following Japanese TrueType fonts:

- G2 Sans Serif-U
- Gothic-B
- Hagoromo-M
- Mincho-L
- Heisei Mincho
- Sirius-L II

For information on installing and using the Japanese TrueType fonts, refer to the Release Notes and Installation Guide text files contained in the `./Japanese_TrueType_Fonts/doc/` directory of the *Tru64 UNIX Japanese Extensions* CD-ROM.



# 2

---

## Basic Setup of Japanese Environment

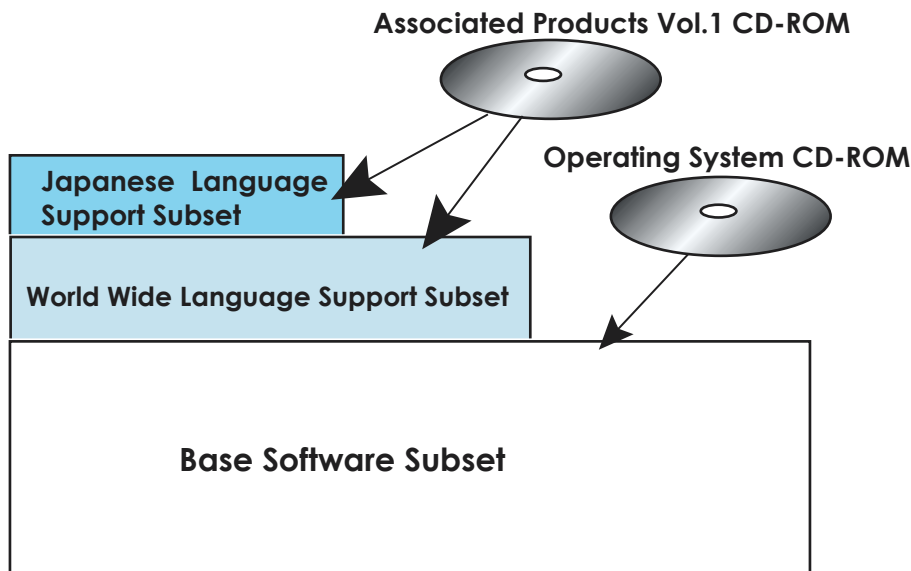
This chapter describes how to perform the following tasks to establish a basic Japanese environment on a Tru64 UNIX system.

- Install the Japanese environment
- Set up the internationalization environment
- Set up the Common Desktop Environment (CDE) language
- Set up the Japanese locale
- Set up the Japanese terminal
- Perform file code conversions with `iconv( )`
- Set up the message catalogs
- Set up the Japanese reference pages

### 2.1 Installing the Japanese Environment

To use the Japanese features that Tru64 UNIX offers, you must install the Worldwide Language Support subsets and Japanese subsets as well as the Tru64 UNIX base software subsets.

**Figure 2–1: Software Subsets Required to Use the Japanese Environment**



The subsets in Figure 2–1 are needed to use the Japanese features and are automatically installed when you select Japanese in the installation procedure. For the detailed steps on the installation procedure, refer to the *Installation Guide* and *Installation Guide — Advanced Topics*.

To use any of the following Japanese features, you must install the associated software kit after installing the Tru64 UNIX Japanese environment:

- **VJE**  
The installation kit for the VJE Japanese input system is contained on a separately orderable CD-ROM. For ordering information, refer to the Worldwide Language Support web page at <http://www.tru64unix.compaq.com/unix/i18n.htm>. For installation information, refer to the *VJE User's Guide*.
- **WX3**  
The Japanese input system WX3 kit is contained on a separately orderable CD-ROM. For ordering information, refer to the Worldwide Language Support web page at <http://www.tru64unix.compaq.com/unix/i18n.htm>. For installation information, refer to the *WX3 User's Guide*.
- **midoc and miman**  
The Japanese document full-text search tool (`midoc`) and reference page full-text search tool (`miman`) are contained on the *Tru64 UNIX Japanese Extensions* CD-ROM. For installation and usage information, refer to Chapter 6.

## 2.2 Setting Up the Internationalization Environment

To use the Tru64 UNIX Japanese features, first set the following environment variables for the internationalization environment:

- **I18NPATH**

Set the directory that will be used as the virtual root directory in the internationalization environment.

- **PATH**

Set `$I18NPATH/bin` at the head of `PATH` to execute the commands included in the Japanese subsets (for example, `jvi`) and to give precedence to the execution of Japanese subsets.

Tru64 UNIX provides the following script files to help you properly set the environment variables:

- `/etc/i18n_profile`

Contains descriptions of the settings you should include in the user's `.profile` file

- `/etc/i18n_login`

Contains descriptions of the settings you should include in the user's `.cshrc` file

- `/etc/i18n_path`

Contains descriptions of the path settings that other commands require to use any of the internationalization commands and utilities

The scripts are template files. The system manager can modify either the `/etc/i18n_profile` or the `/etc/i18n_login` file, as appropriate.

---

**Note**

---

Do not modify the `/etc/i18n_path` file because it depends on the system directory structure.

---

If you use the `/usr/sbin/i18nsetup` command, the contents of the skeleton file are added to the end of each of the user's startup scripts (`$HOME/.profile` and `$HOME/.cshrc`). Use an ASCII editor to reposition these additions within the file.

If you are logged in as superuser (root), you can use the `/usr/sbin/i18nsetup` command to modify the startup script of any user. Consider the following example that modifies the startup files of users `kanai` and `shimada`.

```
# /usr/sbin/i18nsetup kanai shimada
```

If the startup files are properly modified, the commands and utilities for the Tru64 UNIX internationalization environment will be available when you log in to the system, or the settings will be reflected in the current shell environment using `eval` (for `sh`) or `source` (for `csh`).

---

**Note**

---

The descriptions of Japanese environment settings and operations in the remainder of this chapter assume that the environment variable settings for `I18NPATH` and `PATH` described in this section have been made. If the Japanese environment is not set up properly, examine the values of these environment variables.

---

## 2.3 Setting up the CDE Language

To set the language for each user in the CDE, select the appropriate locale from the Language submenu of the login dialog Options when logging in to the system. To set the system language to Japanese, select one of the Japanese locales (listed in Table 2–1) from the Language submenu.

**Table 2–1: Japanese Locales That Can Be Selected for the CDE Language Setting Option**

Language	Locale
Japanese	ja_JP
Japanese (DEC Kanji)	ja_JP.deckanji
Japanese (EUC)	ja_JP.eucJP
Japanese (Super DEC Kanji)	ja_JP.sdeckanji
Japanese (Unicode)	ja_JP.UTF-8
Japanese (Shift JIS)	ja_JP.SJIS

The value of the `LANG` environment variable for CDE sessions is automatically set using the language you select during login. Thus, you do not need to set the `LANG` environment variable in a startup script (such as `.login`).

---

**Note**

---

If you use this method to select a language for a CDE session, you must not define the `LANG` environment variable in the startup script. If there is a mismatch between the language you selected



during login and the value of LANG in the startup script, the session might be unable to start properly.

---

### 2.3.1 Setting the CDE Default Language

If you select only Japanese when you install the Worldwide Language Support software, the default language for the window environment is automatically set to Japanese EUC (ja\_JP.eucJP).

You can use the following procedure to change the default language for the Japanese window environment. This procedure is useful if you want to set a Japanese locale other than EUC for the default language for the window environment or if you selected more than one language when installing the Worldwide Language Support software.

1. If an `/etc/dt/config/Xconfig` file exists, delete it.
2. Create a new `/etc/dt/config/Xconfig` file using the following commands:

```
# cd /etc/dt/config
# ln -s Xconfig.locale-name Xconfig
```

For example, to set the default language to Japanese EUC, use the following command.

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

This setting is enabled for both the login screen and default session languages.

## 2.4 Setting Up the Japanese Locale

To associate the Tru64 UNIX internationalized commands and libraries with a particular Japanese environment, specify the locale that represents the desired combination of language, territory, and codeset.

To specify the locale name representing the locale, use the following format:

*Language\_Territory.Codeset*

For *Language*, *Territory*, and *Codeset*, specify the respective language, territory, and character codeset names defined in the system.

Tru64 UNIX provides the Japanese locales listed in Table 2–2. All of these locales specify the Japanese language currently used in Japan; they differ with respect to the character codeset name used.

**Table 2–2: Japanese Locale Names Available for Tru64 UNIX**

<b>Locale Name</b>	<b>Character Codeset Name Used</b>
ja_JP.deckanji	DEC Kanji
ja_JP.sdeckanji	Super DEC Kanji
ja_JP.eucJP	Japanese EUC
ja_JP.SJIS	Shift JIS
ja_JP	Japanese EUC
ja_JP.UTF-8	UTF-8

At installation, Tru64 UNIX stores the Japanese locale databases in the paths listed in Table 2–3.

**Table 2–3: Locale Databases**

<b>Locale Name</b>	<b>Locale Databases</b>
ja_JP.deckanji	<code>/usr/lib/nls/loc/ja_JP.deckanji</code>
ja_JP.sdeckanji	<code>/usr/lib/nls/loc/ja_JP.sdeckanji</code>
ja_JP.eucJP	<code>/usr/lib/nls/loc/ja_JP.eucJP</code>
ja_JP.SJIS	<code>/usr/lib/nls/loc/ja_JP.SJIS</code>
ja_JP.UTF-8	<code>/usr/lib/nls/loc/ja_JP.UTF-8</code>

To specify a Japanese locale, enter the appropriate locale name listed in Table 2–2 as the value for the LANG environment variable. The commands and utilities of the Tru64 UNIX internationalization environment will be associated with the locale name value set for the LANG environment variable.

In Tru64 UNIX, there are six categories of internationalized commands and libraries that differ from one locale to another. Table 2–4 describes the categories and the settings defined for each. The category names correspond with environment value names. You can independently specify a locale name suitable for each of these environment variables. You can also set up the locales with a variety of environment variable definitions, depending on the processing you want to accomplish.

**Table 2–4: Available Category Names**

Category Name (Environment Variable Name)	Settings
LC_COLLATE	Collating sequence
LC_CTYPE	Character class, character attribute (uppercase, lowercase, etc.)
LC_MESSAGES	Message display format
LC_MONETARY	Monetary format
LC_NUMERIC	Numeric data format
LC_TIME	Date/time format

To assign the same locale name to all these categories, use the `LC_ALL` environment variable.

The system applies the following priority in determining which environment variables control the locale settings:

1. `LC_ALL` environment variable (when a valid locale name has been specified)
2. `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, `LC_MONETARY`, `LC_NUMERIC`, `LC_TIME` environment variables (when a valid locale name has been specified)
3. `LANG` environment variable (when a valid locale name has been specified)

If you specified no locale name or an invalid locale name as an environment variable value, Tru64 UNIX assumes that the C locale has been specified. The C locale is the most basic environment and can process only ASCII characters. With the C locale, Japanese characters cannot be processed.

To view the settings of the current locale, use the `locale` command:

```
% locale
```

For more information on the `locale` command, see `locale(1)`.

## 2.5 Setting Up the Japanese Terminal

To display Japanese characters on the terminal or input them through the keyboard, use the following command to set the terminal driver to the line discipline for Japanese characters:

```
% stty jdec
```

Immediately after execution of this command, the terminal driver is set as follows:

Item	Appropriate stty Option	Settings
Terminal I/O Code	<code>tcode dec</code>	DEC Kanji code
Application code	<code>acode codeset-name</code>	Japanese codeset being set for the locale (eucJP if no locale has been set)
Japanese Input Mode	<code>imode hiragana</code>	Input an 8-bit code as a 2-byte Hiragana code
Kanji Conversion	<code>-ikk</code>	Configuration Off

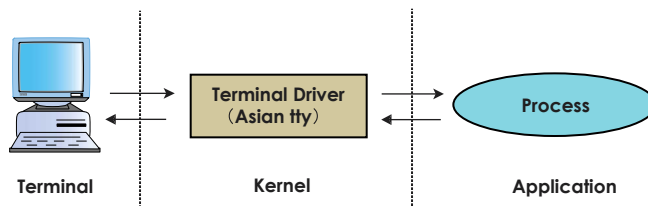
If your environment does not match the above settings, use the `stty` command to establish the appropriate setting. The following sections detail how to use the `stty` command to set handling of terminal and application codes and to set 8-bit data for your environment.

## 2.5.1 Setting the Terminal and Application Codes

You can independently set Kanji codes for each of the terminal and application codes. Various combinations are available.

The application program uses the terminal driver `atty`<sup>1</sup> to communicate with the Tru64 UNIX terminal. The terminal driver `atty` is incorporated in the kernel (see Figure 2-2).

**Figure 2-2: Terminal Driver (Asian tty)**



Tru64 UNIX allows you to select the respective character codes interchanged between the terminal, terminal driver, and application. Throughout this document, the character codes that the terminal uses for communication with the host are called terminal codes and the character codes that the application processes are called application codes.

<sup>1</sup> Terminal driver `atty` (abbreviation of "Asian tty driver") is that extended version of the standard terminal driver that enables communication between the terminal and any of the locales offered by Tru64 UNIX for Asian countries. It also enables Kanji conversion, read and display of user-defined characters, and other features.

### 2.5.1.1 Setting the Terminal Codeset

To set a character codeset other than DEC Kanji for the terminal codes, use the `stty` command with the `-tcode` option:

```
% stty tcode codeset-name
```

Table 2-5 lists the codeset names that you can set.

**Table 2-5: Possible Codeset Names for the Terminal Codesets**

Codeset Name	Associated Mnemonic
DEC Kanji	dec
Shift JIS	SJIS
Japanese EUC	eucJP
7-bit JIS	jis7
8-bit JIS	jis8
UTF-8	UTF-8

### 2.5.1.2 Setting the Application Codes

To set the application codeset, use the `stty` command with the `-acode` option:

```
% stty acode codeset-name
```

Table 2-6 lists the codeset names that you can set.

**Table 2-6: Possible Codeset Names for the Application Codesets**

Codeset Name	Associated Mnemonic
DEC Kanji	deckanji
Super DEC Kanji	sdeckanji
Shift JIS	SJIS
Japanese EUC	eucJP
UTF-8	UTF-8

## 2.5.2 Handling of 8-Bit Data

The Tru64 UNIX system interprets 8-bit data (that is, data with the eighth bit on) that was input through the terminal in one of two ways:

1. The data is interpreted as a 1-byte Kana character, and:

- Passed to the application as a 1-byte, or half-width Kana character without any conversion
- Converted into the 2-byte, or full-width Hiragana code associated with the Kana character of interest
- Converted into the 2-byte, or full-width Katakana code associated with the Kana character of interest

2. The data is interpreted as part of a Kanji code of two or more bytes.

You can use the `stty` command with the `-imode` option to specify one of the following:

- 1-byte, or half-width Kana code

To pass the 8-bit data input as a 1-byte Kana character to the application, use the following `stty` command:

```
% stty imode hankaku
```

Unless a DEC Kanji code that does not support a 1-byte Kana character has been set for the application code, the 1-byte Kana character represented by this code is converted into the associated 2-byte Katakana code.

- 2-byte, full-width Hiragana code

To convert a Kana character represented by the 1-byte (8-bit) input data into the 2-byte Hiragana code, use the following `stty` command:

```
% stty imode hiragana
```

- 2-byte, or full-width Kana character

To convert a Kana character represented by the 1-byte (8-bit) input data into the 2-byte Katakana character, use the following `stty` command:

```
% stty imode katakana
```

- part of character of two or more bytes

To handle the 1-byte (8-bit) input data as part of a Kanji code when DEC Kanji has been set for the terminal codeset, use the following `stty` command:

```
% stty imode kanji
```

### 2.5.3 Checking the Current Terminal Settings

To view the current terminal settings, execute the `stty` command without flags or options:

```
% stty
atty disc;speed 9600 baud; -parity hupcl
erase = ^?
```

```
brkint -inpck -istrip icrnl -ixany onlcr
echo echoe echok
imode=hiragana
tcode=dec acode=deckanji
%
```

The settings displayed correspond to the `stty` options. The options prefixed by a hyphen (-) are currently disabled; the others are currently enabled. See `stty(1)` for a full description of options.

## 2.6 File Code Conversion

The Tru64 UNIX `iconv` command converts file codes. The command uses the following format:

```
iconv -f input-code -t output-code file-name ...
```

The `iconv` command converts the input codeset to the output codeset and writes the resulting data to standard output or to the specified filename(s). You can specify any codeset name in Table 2–7.

**Table 2–7: Possible Codeset Names for use with `iconv`**

Codeset Name	Associated Mnemonic
DEC Kanji	deckanji
Super DEC Kanji	sdeckanji
Japanese EUC	eucJP
Shift JIS	SJIS
7-bit JIS	JIS7
7-bit JIS <sup>a</sup>	jiskanji7
ISO-2022-JP Japanese Mail Code	ISO-2022-JP
ISO-2022-JP Extended Japanese Mail Code	ISO-2022-JPext
UCS-4	UCS-4
UCS-2	UCS-2
UTF-8	UTF-8

<sup>a</sup> JIS7 and jiskanji7 are different in how they represent the JIS X0201 Katakana character set. For details, see `jiskanji(5)`.

For example, the following command converts the `foobar.txt` Shift JIS coded text file into the `foobar.sdc` Super DEC Kanji text file:

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

The user-defined character areas vary from codeset to codeset. When you use `iconv` for code conversion, make sure that the code values assigned to the character codes of the user-defined characters are also converted. For

information on the user-defined character areas of the respective codesets and the character code correspondence between the codesets, see `udc(5)`.

In addition to the codesets listed in Table 2–7, Tru64 UNIX supports file code conversion into codesets used with machines other than those manufactured by HP. Table 2–8 lists these codesets.

**Table 2–8: Possible Codeset Names for use with `iconv` (for Machines Not Manufactured by HP)**

Codeset Name	Associated Mnemonic
IBM Kanji System Character	<code>ibmkanji</code>
Fujitsu JEF Code	<code>JEF</code>
Hitachi KEIS	<code>KEIS</code>

For details on the various codesets, see `iconv_ibmkanji(5)`, `iconv_JEF(5)`, and `iconv_KEIS(5)`.

## 2.7 Message Catalogs

Programs output various messages during execution. When software contains message character strings embedded in the program, the contents of the output cannot vary according to the operating environment.

Instead of embedding message character strings in the program during execution, internationalized Tru64 UNIX commands and libraries reference a file containing logged message characters and output its contents in message form. A message catalog is the mechanism that references a file during execution and outputs the appropriate message.

---

### Note

---

The contents of files containing messages have been compiled by the `genocat` command in binary form. Thus, they cannot be displayed using the `cat` command. To view the file contents, use the `dspscat` command.

---

During installation, Tru64 UNIX stores the Japanese locale message catalogs in directories. Table 2–9 describes the directory path associated with each of the Kanji codes specified with a locale name<sup>2</sup>.

---

<sup>2</sup> For some components, the message file is code-converted during execution. For this reason, the message file of each component does not necessarily exist in the directory of every codeset.



**Table 2–9: Directories Where Japanese Message Catalogs Exist**

Kanji Code	Directory of Japanese Message Catalog
Japanese EUC	<code>/usr/lib/nls/msg/ja_JP</code>
DEC Kanji	<code>/usr/lib/nls/msg/ja_JP.deckanji</code>
Super DEC Kanji	<code>/usr/lib/nls/msg/ja_JP.sdeckanji</code>
Japanese EUC	<code>/usr/lib/nls/msg/ja_JP.eucJP</code>
Shift JIS	<code>/usr/lib/nls/msg/ja_JP.SJIS</code>
UTF-8	<code>/usr/lib/nls/msg/ja_JP.UTF-8</code>

## 2.8 Setting Up the Japanese Reference Pages

Tru64 UNIX provides Japanese reference pages that describe the commands and libraries available in the Japanese environment. Reference pages are documents you can view online using the `man` command. For example:

```
% man n title
```

The document associated with `title` is output to the terminal screen. The value `n` is the section number of the reference page. The reference pages are classified into the sections listed in Table 2–10.

**Table 2–10: Reference Page Section Numbers**

Classification of Contents	Associated Section Number
Commands	1
System Calls	2
Library Routines	3
File Formats	4
nroff macros, etc.	5
Special Files	7
System Management Commands	8

If the `LANG` environment variable is set for use in the Japanese environment, the `man` command searches for the directory that contains the simplified online operation file of the Japanese reference page (`/usr/share/ja_JP.deckanji/man` if `LANG` is `ja_JP.deckanji`) and displays the Japanese reference page.

If the `LANG` environment variable is set to `ja_JP.SJIS`, the `man` command searches for the `/usr/share/ja_JP.eucJP/man` directory and the codeset converter converts the format into `SJIS`.

When you need to view an English reference page, use the `man` command with the `-M` option as follows:

```
% man -M /usr/share/man title
```

For details about the `man` command, see `man(1)`.

---

## Setting Up the Japanese Input System

The Tru64 UNIX operating system provides the Japanese input server `dxjim` as the standard Japanese input system for the window environment. Tru64 UNIX also supports the optional Japanese input systems VJE and WX3.

### 3.1 `dxjim`

The `dxjim` Japanese input server meets the internationalized input method XIM standard, which was adopted in X11R6. The input server also supports the R6 XIM protocol, which became the de facto standard in X11R6, as well as the earlier DECXim protocol.

Because it is based on OSF/Motif Version 1.2.5, the Tru64 UNIX Japanese window environment allows you to input Japanese characters with either the `XmText` or the `DXmCSText` widget.

To enable Japanese input, you must start `dxjim` on the display server that will display the applications.

#### 3.1.1 Starting `dxjim`

To use the session manager and applications for Japanese input, the Japanese input server must be running.

A single Japanese input server supports the Japanese input features for multiple applications on the same display. Also, network applications from a remote node running on the local display can receive Japanese input through the local Japanese input server.

To start `dxjim`, select Japanese on the CDE login screen and log in to the system. Alternatively, you can start `dxjim` explicitly using the following command:

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

#### 3.1.2 Japanese Input with `dxjim`

To input Japanese characters:

1. Start `dxjim`.
2. Start the application on the display where `dxjim` is running.

3. Focus on the character input field (that is, bring the input field to the foreground). The dxjim title changes to represent the input style.
4. Input Japanese characters.

The dxjim input server supports three input styles: on-the-spot, over-the-spot (default), and root-window.

For more information on using dxjim, see Chapter 7.

## 3.2 VJE

VJE for Tru64 UNIX implements the basic functionality of PC Front End Processor (FEP) VJE as the X Window System input server on Tru64 UNIX. VJE is an input method that offers Japanese input conversion service for applications running on Tru64 UNIX.

The VJE installation kit is supplied on a separate CD-ROM. For ordering information, refer to the Worldwide Language Support web page at <http://www.tru64unix.compaq.com/unix/i18n.htm>. For information on installing and using the kit, refer to the *VJE User's Guide*.

## 3.3 WX3

WX3 for Tru64 UNIX implements the basic functionality of PC FEP WX3 as the X Window System input server on Tru64 UNIX.

To use WX3 for Tru64 UNIX, you must have a WX3 license in addition to the operating system license.

The WX3 kit is supplied on a separate CD-ROM. For ordering information, refer to the Worldwide Language Support web page at <http://www.tru64unix.compaq.com/unix/i18n.htm>. For information on installing and using the kit, refer to the *WX3 User's Guide*.

---

## Setting Up a Japanese Printer

The Tru64 UNIX operating system provides a printing function that uses Advanced Printing Software as well as the conventional printing function that uses the `lpr` command. Section 4.1, Section 4.2, and Section 4.3 describe printer operation for the `lpr` command. Section 4.4 describes Advanced Printing Software.

### 4.1 Printer Codes

In an internationalized environment, multiple Japanese text files that use various Kanji codes can reside on the file system. If your printer uses a codeset different from that of the file, you must convert character codes to codes that are suitable for the printer. In this document, a codeset used for a file is called the file codeset and a codeset used for a serial printer is called the printer codeset.

To output file contents to the printer, use the `lpr` command. When you issue the command, it references the Character Codeset section of the current user's locale. The `lpr` command also examines the `/etc/printcap` file, finds the `ya` entry for the printer to use, and references the codeset which is specified there. If the file codeset and printer codeset are different, `iconv( )` converts the file codeset to the printer codeset. The resulting data is output, as a file, to a directory called the printer spool. In addition, the control file used for printer control is written to the printer spool.

The `lpr` command then sends a print request to the printer spool daemon, `lpd`. Upon receipt of the request, `lpd` sends the output file placed under the spool directory to the printer by means of the printer filter that matches the output device. For more information, see `lpr(1)` and `lpd(8)`.

### 4.2 Setting Up a Japanese Printer

To set up a Japanese printer, you must use either the `/usr/sbin/lprsetup` command or the Printer Configuration utility. To perform either of these operations, you must have superuser privileges.

You can issue the `lprsetup` command from the command line.

To start the Printer Configuration utility, select Configure line printer from the SysMan Menu. Alternatively, select Printer Configuration from the Custom Setup subutility of the System Setup utility.

If you use the Printer Configuration utility to set up the Japanese printer, use the following procedure:

1. From the CDE application manager, select System Management ⇒ System Settings ⇒ Printer icons to start the Printer Configuration utility.
2. In Available Printer Types, click on the name of the printer you want to set up. Japanese printer names end with “ja”.

Example: DEClaser2300

3. Click Configure.
4. In the dialog box, enter the Printer Aliases and the name of the device to which the printer is connected.
5. Click Advanced and use the dialog box to set up detailed conditions such as flag page suppression, or to enable the printer to print user-defined characters in Printer ODL.
  - a. To set the Printer ODL feature, select the SoftODL cache size (ys): field and enter the size value (for example, 256).
  - b. Select the Country-specific parameters (ya): field and add the ODL-style settings.

For example, if you want to use a font size 40 x 40 with ln05ja, enter the following:

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

- c. Click OK.
6. Click Commit to save the settings. Click Close.

#### Example 4–1: Sample /etc/printcap File

---

```
lp0|0|ln05:\
:af=/usr/adm/lp0acct:\
:br#9600:\
:ct=dev:\
:fc#0177777:\
:fs#023:\
:if=/usr/lbin/ln05jaof:\
:lf=/usr/adm/lp0err:\
:lp=/dev/tty01:\
:mx#0:\
```

### Example 4–1: Sample /etc/printcap File (cont.)

---

```
:of=/usr/lbin/ln05jaof:\
:pl#66:\
:pw#80:\
:sd=/usr/spool/lpd0:\
:xc#0177777:\
:xf=/usr/lbin/xf:\
:xs#044000:\
:ya="plocale=ja_JP.sdeckanji odlstyle=normal-40x40":\
:ys#256:
```

---

For more information on the Tru64 UNIX printer support internationalization features, see `il8n_printing(5)`.

## 4.3 Setting Up an ESC/P Printer

Tru64 UNIX provides two configuration files (`escp_a4_10cpi` and `escp_a4_12cpi`) for the `pcfof` filter that enable you to use the ESC/P printer. You can use the configuration files to set the ESC/P printer initialization codes and to configure the settings required to print 10 or 12 CPI (characters per inch) data.

Select either the `escp_a4_10cpi` or `escp_a4_12cpi` configuration file with the `lprsetup` command to automatically set up the ESC/P printer in the `/etc/printcap` file. To edit the `/etc/printcap` file manually, set the `if` and `of` entries as follows:

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

In addition, you must set `ja_JP.eucJP` as the printer locale for the `ya` entry, as follows:

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

## 4.4 Installing and Setting Up Advanced Printing Software

Advanced Printing Software is a distribution client/server printing system that is based on Xerox PrintXchange technology. Advanced Printing Software transmits and receives print jobs to and from the `lpr` and `lpd` print subsystems by using the inbound and outbound gateways.

The *Associated Products Volume 1* CD-ROM in the present release contains the standard version kit of Advanced Printing Software. The Japanese kit is supplied on the Worldwide Software Language Support kit. To use Advanced Printing Software to output data to the Japanese printer, you must install both of the kits on the system.

The Japanese kit of Advanced Printing Software contains environments that enable Japanese text or Japanese PostScript files to be output to Japanese printers. (The printer must support Japanese PostScript.) For details on installing and using the Japanese kit of Advanced Printing Software, refer to the Japanese documentation for Advanced Printing Software.



# 5

## Setting Up a Japanese Keyboard

The Tru64 UNIX system supports the following Japanese keyboards (in this table, Yes means that the keyboard has a Conversion key or Kana LED):

Keyboard	Language	Conversion Key	Kana LED
LK411-AJ/LK41W-AJ Japanese LK411	ANSI	Yes	Yes
LK411-JJ/LK41W-JJ Japanese LK411	JIS	Yes	Yes
PCXAJ-AA Japanese PC106	JIS	Yes	No
LK97W-AJ Japanese PC109	JIS	Yes	No

- LK41 W-AJ and LK41 W-JJ are equivalent to LK411-AJ and LK411-JJ, respectively. Set up the LK41 W-AJ as you would set up the LK411-AJ; likewise, set up the LK41 W-JJ as you would the LK411-JJ.
- LK97W-AJ is a version of PCXAJ-AA with three additional keys for Windows. The three additional keys do not work on Tru64 UNIX. Set up the LK97W-AJ as you would set up the PCXAJ-AA.

Check the back of the keyboard for the model number.

### 5.1 Setting Up a CDE Keyboard

To use the Japanese keyboard, you must set it up for CDE sessions as described in this section.

To set up the keyboard for each user, use the Set Up Keyboard action from the Desktop Application of the Application Manager. Select the keyboard name of your Japanese keyboard from the keyboard list<sup>1</sup>.

For details about setting the keyboard, refer to the *CDE Companion*.

For details about setting the system default keyboard, see Section 5.2.

<sup>1</sup> If you specify a keyboard name that contains esc, the ' key, located at the top left corner, can be used as the Escape key. In this case, "" and "~" move to the < key.

---

### Note

---

The LK41W-AJ Japanese keyboard must be used with the same settings as the LK411-AJ. The LK41W-JJ Japanese keyboard must be used with the same settings as the LK411-JJ. The LK97W-AJ Japanese keyboard must be used with the same settings as the PCXAJ-AA.

---

## 5.1.1 Kana Mode

After you have set up the Japanese keyboard, turn on Kana mode to type Kana characters through the keyboard. When you type a keyboard key in Kana mode, it inputs the Kana character (rather than the ASCII character) displayed on the keytop.

On the PCXAJ-AA and LK97W-AJ keyboards, turn on Kana mode by pressing the right **Ctrl**<sup>2</sup>. On other Japanese keyboards, turn on Kana mode by pressing the **カナ** (KANA) key, located at the bottom left corner of the keyboard. Repeating the same operation returns the keyboard to ASCII mode.

To type Kana characters, you must be in the Japanese environment.

## 5.1.2 Conversion Keys

Some Japanese keyboards have the **無変換** (MU-HENKAN), **前候補/変換** (MAE-KOUHO/HENKAN), and **カタカナ/ひらがな** (KATAKANA/HIRAGANA) keys at both sides of the **Space** bar. These keys are called conversion keys and can be used for Kana-Kanji conversion input from a Japanese input server.

## 5.1.3 Kana LED

When the Kana LED on a Japanese keyboard is lit, it indicates that Kana mode is on.

In earlier versions of Tru64 UNIX, the keyboard indicator (`kb_indicator`) is controlled the Kana LED. In Version 4.0 or higher, support of XKB enables the Kana LED to blink properly without the `kb_indicator`.

---

<sup>2</sup> The PCXAJ-AA and LK97W-AJ keyboards each have two **Ctrl** keys.

## 5.2 Setting Up a Japanese Keyboard in Console or Single-User Mode

In Tru64 UNIX Version 4.0 and later, the settings for the Japanese keyboard have been modified to improve ease of use and the console firmware has been modified to resolve the following problems:

- When the JIS layout Japanese keyboard on the console firmware was used, a character not matching the one displayed on the keytop was occasionally entered.
- When the JIS layout Japanese keyboard was used in the operating system single-user mode, a character not matching the one displayed on the keytop was occasionally entered.
- Though the default keyboard setting was unnecessary when the English keyboard was used, it was necessary when the Japanese keyboard was used.

The latest (modified) console firmware CD-ROM is included in the installation kit for this version.

Use the following procedure to set up the Japanese keyboard with the console firmware:

1. Update the console firmware.  
For details, refer to the console firmware document.
2. Set the console firmware variables `kbd_hardware_type` and `language`. The `kbd_hardware_type` variable indicates the keyboard type. The `language` variable indicates the keyboard layout and language.

Specify LK411 (for the LK41\* Japanese keyboard) or PCXAL (for the PCXAJ-AA and LK97W-AJ keyboards) as the value for the `kbd_hardware_type` variable.

Specify 50 (for the JIS layout Japanese keyboard) or 52 (for the ANSI layout Japanese keyboard) as the value for the `language` variable.

The following table lists the variable values by keyboard:

Keyboard	<code>kbd_hardware_type</code>	Language
LK411-AJ/LK41W-AJ	LK411	52
LK411-JJ/LK41W-JJ	LK411	50
PCXAJ-AA	PCXAL	50
LK97W-AJ	PCXAL	50

The following example sets up the LK41W-JJ keyboard:

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

You can enter more than one command on a line, for example:

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

3. Initialize the console firmware with the following command:

```
>>> init
```

Alternatively, you can initialize the console firmware by powering off the system and powering it on again.

If you replace the keyboard, power off the system first.

4. Confirm that the keyboard variable values have been changed by using the `show` command.

```
>>> show
      :
kbd_hardware_type      LK411
language                50
language_name          Japanese (JIS)
      :
```

# 6

---

## Installing and Setting Up Japanese Document Full-Text Searching Tools

The Tru64 UNIX operating system offers two Japanese document full-text searching tools; MitakeSearch for online documents (`midoc`) and MitakeSearch for reference pages (`miman`). These tools use the HP Japanese full-text searching system, MitakeSearch, and can execute full-text search for any character string in an online document or reference page.

- The `midoc` tool searches a Japanese HTML document using the Web browser interface.
- The `miman` tool can search either a Japanese or English reference page using a command-line or Web interface. In comparison to the operating system `apropos` command, which executes a search with one of the keywords registered in the `whatis` database, `miman` executes a full-text search in a reference page.

Because the `midoc` and `miman` tools execute Web browser searches by means of a CGI script, you must install the tools on the system and set them up for the HTTP server. This chapter describes installation of the `midoc` and the `miman` tools and their required settings.

The `midoc` and `miman` kits are contained on the *Tru64 UNIX Japanese Extensions* CD-ROM.

For information on using the `miman` and `midoc` tools with the Browser interface, click the Help icon or button in the appropriate search window and refer to the online help.

For information on using the `miman` tool with the command-line interface, see `miman(1)`.

For information on MitakeSearch, the Japanese full-text searching system, refer to the following Web page:

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

### 6.1 Installing the `midoc` Software

Use the `setld` command to install MitakeSearch for Online Documents (`midoc`). The expanded destination of the kit is fixed to the `/var/opt/midoc` directory and 27 MB of free space is required.

The following sections describe the procedure for installing the `midoc` software.

### 6.1.1 Securing the Expanded Destination of the Kit

Make sure that the `/var/opt` directory has at least 27 MB of free space. Expand the distribution kit under the `/var/opt/midoc` directory.

If the directory does not have sufficient free space, use the following steps to make a directory for `midoc` on a separate disk or partition and establish a symbolic link from the `/var/opt/midoc` directory.

1. Log in to the system as a root user.
2. Make the `/usr/local/midoc` directory:  

```
% mkdir -p /usr/local/midoc
```
3. Set root for the owner of the `/usr/local/midoc` directory:  

```
% chown root /usr/local/midoc
```
4. Set 755 for the permission mode of the `/usr/local/midoc` directory:  

```
% chmod 755 /usr/local/midoc
```
5. Link the `/var/opt/midoc` directory to the `/usr/local/midoc` directory:  

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

### 6.1.2 Installing the `midoc` Software From the Kit

The `midoc` kit is configured as a single unit with the `IOSMIDOCnnn` subset name, where `nnn` represents the version number of the `midoc` software. To install the `midoc` software, mount the *Tru64 UNIX Japanese Extensions* CD-ROM under the `/mnt` directory, and use the following `setld` load command:

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

From the installation dialogue, select the `midoc` software subset for installation. For information on using the `setld` command, see the *Installation Guide* and `setld(8)`.

### 6.1.3 Setup After Installation

The `midoc` tool assumes that a search is executed using the Web browser by means of a CGI script. After `midoc` is installed, you must configure the HTTP server as well as the `midoc` software. The following sections describe how to configure HTTP and `midoc`.

### 6.1.3.1 Copying the Icon Files

Copy the following files, which are stored under the `/var/opt/midoc/cgi-bin` directory, to the HTTP server's icon directory. Use the `http://host-name/icons/ URL` to access the icon directory. If the icon directory does not exist, use the `mkdir` command to create one.

- `midoc-c.gif`
- `midoc-next.gif`
- `midoc-prev.gif`
- `midoc-t.gif`
- `midoc-help.gif`
- `midoc-p.gif`
- `midoc-search.gif`
- `midoc-x.gif`

After the files are copied, set the file protection to allow user access.

### 6.1.3.2 Copying the Perl Scripts

Copy the following Perl scripts, which are stored under the `/var/opt/midoc/cgi-bin` directory, to the CGI directory. Use the `http://host-name/cgi-bin/ URL` to access the CGI directory.

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

After the scripts are copied, set the file protection to allow user access.

If your system is using `MitakeSearch` or `miman`, `libjconv.pl` already exists. These files are identical and can be used interchangeably.

### 6.1.3.3 Editing the Map File

Edit the map file (`/var/opt/midoc/etc/urlmap.cgi`) to enable the URL resulting from a `midoc` search to be mapped to a URL that can be displayed.

In general, the URL of the document registered in the `midoc` database must be similar to the mount point for the *Tru64 UNIX Japanese Extensions* CD-ROM (`/usr/share/doclib/online`), except that the URL host name is registered by means of the `file://` protocol in `localhost`. Thus, all documents have been registered by means of a URL beginning with `file://localhost/usr/share/doclib/online/`.

The edit you make to the URL map file depends on whether the CD-ROM is mounted locally or mounted on the Web Server.

Consider the following two examples. In either case, the separator must be one `TAB` without inserting a blank.

If the *Tru64 UNIX Japanese Extensions* CD-ROM is mounted under the local system's `/mnt` directory and the `midoc` user executing the search accesses it as a local file, edit the URL map file as follows:

---

#### Example 6–1: Accessing a CD-ROM Mounted Locally

---

Before edit:

```
file://localhost/usr/share/doclib/online/TABhttp://hostname/path/
```

After edit:

```
file://localhost/usr/share/doclib/online/TABfile:/mnt/
```

---

If the *Tru64 UNIX Japanese Extensions* CD-ROM is mounted on the Web server and is accessed by means of the URL `http://hostname/document/`, edit the URL map file as follows:

---

#### Example 6–2: Accessing a CD-ROM Mounted on the Web Server

---

Before edit:

```
file://localhost/usr/share/doclib/online/TABhttp://hostname/path/
```

After edit:

```
file://localhost/usr/share/doclib/online/TABhttp://hostname/document/
```

---

If the files on the CD-ROM have been copied to disk with the same directory configuration, make the same edits to the URL map file.

### 6.1.4 Executing the Search

To execute the search CGI script, access the HTTP server that was set up in Section 6.1.3.3 by using the following URL:

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

If the appropriate document cannot be displayed even by clicking the URL in the Search Results window, recheck the settings of the URL map file (`/var/opt/midoc/etc/urlmap.cgi`).

### 6.1.5 Uninstalling the `midoc` Software

To uninstall the `midoc` software, use the `setld` command:

```
% setld -d IOSMIDOCnmm
```



Where *nnn* is the version number of the `midoc` software subset. If you do not know the complete subset name for the installed `midoc` software, you can use `setld -i | grep IOSMIDOC` to display the name.

## 6.2 Installing the `miman` Software

Use the `setld` command to install `MitakeSearch for Man Pages (miman)`. The expanded destination of the kit is fixed to the `/var/opt/miman` directory and 77 MB of free space is required.

The following sections describe the procedure for installing the `miman` software.

### 6.2.1 Securing the Expanded Destination of the Kit

Make sure that the `/var/opt` directory has at least 77 MB of free space. Expand the distribution kit under the `/var/opt/midoc` directory.

If the directory does not have sufficient free space, use the following steps to make a directory for `miman` on a separate disk or partition and establish a symbolic link from the `/var/opt/miman` directory.

For example, if you want the kit expanded under the `/usr/local/miman` directory because `/usr/local` has sufficient free space, use the following procedure.

1. Log in to the system as a root user.
2. Make the `/usr/local/miman` directory:  

```
% mkdir -p /usr/local/miman
```
3. Set root for the owner of the `/usr/local/miman` directory:  

```
% chown root /usr/local/miman
```
4. Set 755 for the permission mode of the `/usr/local/miman` directory:  

```
% chmod 755 /usr/local/miman
```
5. Link the `/var/opt/miman` directory to the `/usr/local/miman` directory:  

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

### 6.2.2 Installing the `miman` Software from the Kit

The `miman` kit is configured as a single unit with the `IOSMIMANnnn` subset name, where *nnn* represents the version number of the `miman` software. To install the `miman` software, mount the *Tru64 UNIX Japanese Extensions* CD-ROM under the `/mnt` directory, and use the following `setld` load command:

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

From the installation dialogue, select the miman software subset for installation. For information on using the `setld` command, see the *Installation Guide* and `setld(8)`.

### 6.2.3 Setup After Kit Expansion

You can set up the miman software for use with a command-line interface or with a Web interface.

- For command-line use, no setup is necessary. You can use `miman` immediately after expanding the kit. For information on using `miman` as a command, see `miman(1)`.
- For Web use, you must configure the HTTP server as explained in this section.
- A miman search database is installed when you install the kit. For information about adding or deleting content from the database, see `mimanctl(1)`.

To use the Search CGI and Schedule Management CGI Clients, execute the following steps after installing `MitakaeSearch for Man Pages`.

1. Copy the following files, stored under the `/var/opt/miman/cgi-bin` directory, to the HTTP server's icons directory. Use the `http://host-name/icons/ URL` to access the icon directory. If the icon directory does not exist, use the `mkdir` command to create one.

- `miman-c.gif`
- `miman-go.gif`
- `miman-help.gif`
- `miman-m.jpg`
- `miman-p.gif`
- `miman-x.gif`

After the files are copied, set the file protection to allow user access to the files.

2. Copy the following Perl scripts, stored under the `/var/opt/miman/cgi-bin` directory, to the CGI directory. Use the `http://host-name/cgi-bin/ URL` to access the CGI directory.

- `miman.pl`
- `miman2html.pl`
- `libjconv.pl`

After the scripts are copied, set the file protection to allow user access.

To enable the Search CGI Client, specify the following URL in the Web browser:

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

## 6.2.4 Uninstalling the miman Software and Other Notes

- To uninstall the miman software, use the `setld` command. After uninstalling it, some files and directories you created might remain. Delete them manually.

```
% setld -d IOSMIMANnnn
```

- Where *nnn* is the version number of the miman subset. If you do not know the complete subset name for the installed miman software, you can use `setld -i | grep IOSMIMAN` to display the name.

In some rare cases, an error can occur during database update and destroy the database. If the database is destroyed, reinstall the kit. Before the reinstallation, be sure to uninstall the miman software. Reinstallation causes user-registered data to be lost.



---

## Japanese Input with dxjim

This chapter describes the dxjim Japanese input server, which is the standard Japanese input server on the Tru64 UNIX operating system.

### 7.1 Outline of the Japanese Conversion Features

The Tru64 UNIX window environment allows you to input Japanese characters in text input fields. To input Japanese characters, type a Roman (Romaji) or Kana character string that represents the phones (YOMI) and then use the 変換 (HENKAN) key to convert it into a string of Kanji, Hiragana, or Katakana characters. Table 7-1 lists the Japanese input conversion features.

**Table 7-1: Japanese Input Conversion Features**

Feature	Description
Kanji Conversion, Next Phrase Candidate	Converts the characters into Kanji characters. Pressing more than once displays the next candidate.
Hiragana Conversion, Phrase Hiragana Conversion	Converts the characters into Hiragana characters.
Katakana Conversion, Phrase Katakana Conversion	Converts the characters into a Katakana characters.
Full-Width Conversion	Converts full-width alphanumeric characters into half-width characters converts full-width symbols into half-width symbols.
Half-width Conversion	Converts a half-width alphanumeric character or symbol into the full-width one.
Symbol Conversion, Code Input	Converts one to three alphanumeric characters or symbols into special symbols (see Section 7.4.3), or converts Kanji codes into Kanji characters (see Section 7.4.4).
Uppercase Conversion	Converts lowercase characters into uppercase characters.
Lowercase Conversion	Converts uppercase characters into lowercase characters.

**Table 7–1: Japanese Input Conversion Features (cont.)**

Feature	Description
Move to Next Phrase	Moves to the next phrase if the resulting character string from Kanji Conversion contains two or more phrases (that is, the next phrase is regarded as the target to convert).
Move to Previous Phrase	Moves to the previous phrase if the resulting character string from Kanji Conversion contains two or more phrases (that is, the previous phrase is regarded as the target to convert).
Reduce Phrase	Reduces the phrase to convert.
Enlarge Phrase	Enlarges the phrase to convert.
Previous Phrase Candidate	Returns to the preceding candidate for Kanji Conversion.
Determine Full Text	Determines the conversion of the entire phone you entered.
Determine Phrase	Determines the conversion of the phrase that is the current target to convert. (If the text contains two or more phrases, the next phrase is regarded as the target to convert.)
Cancel Conversion, No Conversion	Returns to the initial phones immediately after they are converted (the phones can be modified). Alternatively, determines the input, with the initially typed characters as they are, immediately after the phones are typed.

## 7.2 Using dxjim

To use the dxjim input server, first start the application on the display where dxjim is running. Then, focus on the character input field (that is, bring it to the foreground). The dxjim title changes to represent the input style and you can now input Japanese characters.

The dxjim input server supports three input styles: over-the-spot (default), on-the-spot, and root-window.

Under the over-the-spot style, Japanese Conversion occurs in the application window and allows you to perform Japanese input without considering dxjim. However, because the character string currently being converted is displayed in the application window, an attempt to insert a character between existing characters might hide part of the existing string until the conversion is determined. Also, if you proceed with the process immediately after character input (for example, by pressing the **OK** button), the characters for which the conversion has been determined are not included.

In this case, you have to determine the conversion by pressing **Ctrl/G** or **Ctrl/N** for the JVMS conversion key before proceeding with the process.

Under the on-the-spot style, Japanese Conversion occurs in the application window and allows you to perform Japanese input without considering dxjim. However, unlike the over-the-spot style, no part of an already typed character string is hidden when you attempt to insert a character between existing characters.

Under the root-window style, you must consider dxjim. That is, the root-window style displays in the dxjim window, any character string you typed in the application window. Because the entire character string currently being converted is displayed in the dxjim window, you must be aware of the dxjim window until the Japanese Conversion is determined.

For information on setting up the input styles, see Section 7.2.2.

## 7.2.1 Setting Up dxjim

When running, the dxjim input server displays either the character string 漢字変換 (Kanji Conversion) or "ASCII" at the bottom left field of the dxjim window. While 漢字変換 remains displayed there, the application can input Japanese characters using dxjim. When "ASCII" is displayed, Japanese input does not work. Immediately after startup of the application, dxjim is in the 漢字変換 mode.

The key used to change the mode from "ASCII" to 漢字変換 is called the Start key. The key used for reverse mode (from 漢字変換 to "ASCII") is called the End key.

By default, the Start and End key is Shift-Space (that is, pressing the space bar while holding down the **Shift** key). To change the Start and End key settings, use the Start Key: Settings dialog box that displays when you select the Set Key menu from the Options menu. Changes to the Start and End keys made in the Start Key: Settings dialog box are effective for all applications.

The dxjim input server uses the user key definition library IMLIB. The dxjim input server uses the key definition of the user who has started the input server. For information on definitions of the conversion and user keys, and IMLIB, refer to the *User Key Definition User's Guide*.

To change the conversion key, select Key Definition from the Options menu. Using this method, you can change only the definition of the conversion key; you cannot change the manner of echoing. The conversion key change applies only to the widget you used for input of Japanese characters while the change was in effect.

If the application uses root-window input style, a preedit area is created in the dxjim window. The key used to determine the preedit for input to the application is called the Determination key. The Determination key default is the `Return` key.

To change the Determination key, use the Start key: Settings dialog box. The new Determination key applies to all applications that use the root-window input style.

## 7.2.2 Input Styles

The input styles supported by dxjim (on-the-spot, over-the-spot, and root-window) are listed as input styles for text window input in the X11R6.3 standard. (X11R6.3 also lists a few preedit modes, which are not supported by dxjim). The dxjim input server can also display the mode of the connected client.

You can use the `XmText( )` and `DXmCSText( )` widgets to set the input style using the `VendorShell( )` widget's `XmNpreeditType( )` resource. For this resource, specify one of the following values:

- `OnTheSpot`
- `OverTheSpot`
- `Root`
- `OffTheSpot`

The `XmNpreeditType( )` resource allows you to specify two or more of these values, separated by commas.

The default input style is `OverTheSpot`. Under the `OverTheSpot` style, the character string currently being converted is initially overwritten on the existing text. After the conversion is determined, the string is inserted into the entire text.

## 7.2.3 Using dxjim Resources

Using resources, you can modify dxjim settings. The resources available for dxjim are listed in this section. The class name for dxjim is `DXjim` and the default resource file is `/usr/i18n/lib/X11/app-defaults/DXjim`.

- `x`  
Specifies the x coordinate of the top left corner of the dxjim window.

Setting:	Use an integer that represents the number of pixels counted from the left margin
Initial setting:	Bottom left corner of the screen



- `y`

Specifies the `y` coordinate of the top left corner of the `dxjim` window.

Setting: Use an integer that represents the number of pixels counted from the top margin

Initial setting: Bottom left corner of the screen

- `startupKanjiMode`

Enables or disables Japanese input immediately after startup of the text widget. If `startupKanjiMode` is on, you enter Japanese input mode immediately after startup of the text widget.

Setting: on or off

Initial setting: on

- `startKey`

Sets the Start key. You can also set `startKey` in the Set Key dialog box by selecting Set Key from the Options menu.

Setting: See Section 7.2.1

Initial setting: Shift<Key>space

- `endKey`

Sets the End key. You can also set `endKey` in the Set Key dialog box by selecting Set Key from the Options menu.

Setting: See Section 7.2.1

Initial setting: Shift<Key>space

- `root_commitKey`

Sets the Determination key. You can also set `root_commitKey` in the Set Key dialog box by selecting Set Key from the Options menu.

Setting: See Section 7.2.1

Initial setting: <Key>Return

- `kp_numlock`

If the `kp_numlock` setting is `False`, the keypad keys cannot be used to input preedit strings. The preedit is determined by pressing any keypad key during preedit. If the `kp_numlock` setting is `True`, the keypad keys, except the PF1 to PF4 and Enter, can be used to input preedit strings.

Setting: True or False  
Initial setting: False

- `icon_state`

If the `icon_state` setting is Normal, the dxjim window is restored by starting preedit with the dxjim window iconized. If the `icon_state` setting is Iconic, automatic restoration of the dxjim window never takes place. If the `icon_state` setting is Withdrawn, the dxjim window does not appear and the root preedit style is not available. To use an icon to start dxjim, set up the resource as follows:

```
DXjim*initialState: IconicState
```

Setting: Iconic, Withdrawn, or Normal  
Initial setting: Normal

- `use_root_resources`

Determines whether to use the root preedit style resources (`root_fontset`, `root_background`, `root_foreground`, `root_highlight`, and `root_highlight_fore`). If the `use_root_resources` setting is False, the setting defined with the Xset-ICValues of the client will be used.

Setting: True or False  
Initial setting: True

- `root_background`

Specifies the background color that is used for preedit string areas of the root preedit style when the `use_root_resource` resource is True.

Type: Pixel  
Initial setting: white

- `root_foreground`

Specifies the foreground color that is used for preedit string areas of the root preedit style when the `use_root_resource` resource is True.

Setting: Color setting. A named color is available.  
Initial setting: black

- `root_highlight`

Specifies the background color of conversion candidate character strings that is used for preedit string areas of the root preedit style when the `use_root_resource` resource is True.

Setting: Color setting. A named color is available.  
Initial setting: black

- `root_highlight_fore`

Specifies the foreground color of conversion candidate character strings that is used for preedit string areas of the root preedit style when the `use_root_resource` resource is `True`.

Setting: Color setting. A named color can also be used.  
Initial setting: white

- `root_fontset`

Specifies the font set that is used for preedit strings of the root preedit style when the `use_root_resource` resource is `True`.

Setting: Color setting. A named color can also be used.  
Initial setting: See the following:

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

- `root_width`

Specifies the width of preedit areas of the root preedit style.

Setting: The number of pixels (integer) that represents the width  
Initial setting: Determined dynamically

## 7.3 Japanese Input Conversion Keys

Tru64 UNIX provides definitions for the conversion keys used for text input. Japanese VMS (JVMS) is the default conversion key definition unless otherwise specified. See Section 7.3.1 for information on changing the conversion key definition.

The conversion key definitions are:

- JVMS (default)

The standard Japanese VMS conversion key definition. The JVMS conversion keys are listed in Table 7-2.

- EVEJ

The conversion key definition for the EVEJ editor and used in the EVEJ keypad mode of Japanese EVE. The EVEJ conversion keys are as listed in Table 7-3.

- **KEYPAD**

The key definition that uses the VT-series terminal numeric keypad. The KEYPAD conversion keys are as listed in Table 7-4.

- **TARO**

The conversion key definition that is similar to that of the word processing software Ichitaro. The TARO conversion keys are as listed in Table 7-5.

- **CTRL**

The key definition that uses the Control key. The CTRL conversion keys are as listed in Table 7-6.

**Table 7-2: JVMS Conversion Key Definition**

Function	Keystrokes
Kanji Conversion or Next Phrase Candidate	Ctrl+Space
Kanji Candidate List	Ctrl+[ is a key located at the top left corner of the keyboard
Hiragana Conversion of Phrase Hiragana Conversion	Ctrl+L
Katakana Conversion or Phrase Katakana Conversion	Ctrl+K
Half-Width Kana Conversion	Ctrl+G Ctrl+K
Full-Width Conversion	Ctrl+F
Half-Width Conversion	Ctrl+G Ctrl+F
Symbol Conversion or Code Input	Ctrl+]
Uppercase Conversion	N/A
Lowercase Conversion	N/A
Move to Next Phrase	Ctrl+P
Move to Previous Phrase	Ctrl+G Ctrl+P
Reduce Phrase	Ctrl+
Enlarge Phrase	Ctrl+G Ctrl+
Previous Phrase Candidate	Ctrl+G Ctrl+Space or Ctrl+G Ctrl+L
Determine Full Text	Ctrl+G Ctrl+N
Determine Phrase	N/A
Cancel Conversion or No Conversion	Ctrl+N

**Table 7-3: EVEJ Conversion Key Definition**

Function	Keystrokes
Kanji Conversion or Next Phrase Candidate	Ctrl+Space
Kanji Candidate List	Ctrl+' is a key located at the top left corner of the keyboard
Hiragana Conversion or Phrase Hiragana Conversion	Ctrl+H
Katakana Conversion or Phrase Katakana Conversion	Ctrl+K
Half-Width Kana Conversion	PF1 Ctrl+K
Full-Width Conversion	Ctrl+F
Half-Width Conversion	Ctrl+E
Symbol Conversion or Code Input	PF1 Z or PF1 z
Uppercase Conversion	N/A
Lowercase Conversion	N/A
Move to Next Phrase	Ctrl+P
Move to Previous Phrase	PF1 Ctrl+P
Reduce Phrase	Ctrl+A
Enlarge Phrase	PF1 Ctrl+A
Previous Phrase Candidate	PF1 Ctrl+Space
Determine Full Text	PF1 Ctrl+N
Determine Phrase	N/A
Cancel Conversion or No Conversion	Ctrl+N

**Table 7-4: KEYPAD Conversion Key Definition**

Function	Keystrokes <sup>a</sup>
Kanji Conversion or Next Phrase Candidate	ⓀⓅ7
Kanji Candidate List	ⓀⓉⓇ+Ⓜ is a key located at the top left corner of the keyboard
Hiragana Conversion or Phrase Hiragana Conversion	ⓀⓅ1
Katakana Conversion or Phrase Katakana Conversion	ⓀⓅ2
Half-Width Kana Conversion	ⓀⓅ6
Full-Width Conversion	ⓀⓅ3
Half-Width Conversion	ⓀⓅ9
Symbol Conversion or Code Input	ⓀⓅ8
Uppercase Conversion	N/A
Lowercase Conversion	N/A
Move to Next Phrase	ⓀⓅ5
Move to Previous Phrase	ⓀⓅ1 ⓀⓅ5
Reduce Phrase	ⓀⓅ4
Enlarge Phrase	ⓀⓅ1 ⓀⓅ4
Previous Phrase Candidate	ⓀⓅ1 ⓀⓅ7
Determine Full Text	N/A
Determine Phrase	N/A
Cancel Conversion or No Conversion	ⓀⓅ0

<sup>a</sup> ⓀⓅ0 to ⓀⓅ9 represent keys 0 to 9 on the numeric keypad.

**Table 7-5: TARO Conversion Key Definition**

Function	Keystrokes
Kanji Conversion or Next Phrase Candidate	Space
Kanji Candidate List	Ctrl+[ is a key located at the top left corner of the keyboard
Hiragana Conversion or Phrase Hiragana Conversion	F11
Katakana Conversion or Phrase Katakana Conversion	F12
Full-Width Conversion Full-Width Conversion does not work after Kanji, Katakana, or Hiragana Conversion. In these cases, use Cancel Conversion before using Full-Width Conversion.	F14
Half-Width Conversion Half-Width Conversion does not work after Kanji, Katakana, or Hiragana Conversion. In these cases, use Cancel Conversion before using Half-Width Conversion.	F13
Symbol Conversion or Code Input	Ctrl+]
Uppercase Conversion	N/A
Lowercase Conversion	N/A
Move to Next Phrase	↓
Move to Previous Phrase	N/A
Reduce Phrase	←
Enlarge Phrase	→
Previous Phrase Candidate	↑
Determine Full Text	Return
Determine Phrase	↓
Cancel Conversion The TARO Conversion Key Definition does not contain the むん (MU-HENKAN, No Conversion) key.	DELETE



**Table 7–6: CTRL Conversion Key Definition**

Function	Keystrokes
Kanji Conversion or Next Phrase Candidate	Ctrl+K
Kanji Candidate List	Ctrl+' ' is a key located at the top left corner of the keyboard
Hiragana Conversion or Phrase Hiragana Conversion	Ctrl+G
Katakana Conversion or Phrase Katakana Conversion	Ctrl+N
Half-Width Kana Conversion	Ctrl+H
Full-Width Conversion	Ctrl+E
Half-Width Conversion	Ctrl+T
Symbol Conversion or Code Input	Ctrl+F
Uppercase Conversion	N/A
Lowercase Conversion	N/A
Move to Next Phrase	Ctrl+P  r
Move to Previous Phrase	Ctrl+P  l
Reduce Phrase	Ctrl+A
Enlarge Phrase	Ctrl+X
Previous Phrase Candidate	Ctrl+L
Determine Full Text	N/A
Determine Phrase	N/A
Cancel Conversion or No Conversion	Ctrl+B

### 7.3.1 Changing the Conversion Key Definition

By default, dxjim uses the JVMS Conversion Key Definition.

To change the conversion key definition you use for applications, use the following steps. The steps use the example of changing the key definition to EVEJ.

1. Define the IM\_PROFILE environment variable as follows:

```
% setenv IM_PROFILE im_profile_evej (When csh is in use)
$ IM_PROFILE=im_profile_evej
$ export IM_PROFILE (When sh or ksh is in use)
```

2. To use the JVMS, TARO, CTRL, or KEYPAD Conversion Key Definition, change im\_profile\_evej (shown in the example in step 1) to one of the following:
  - JVMS: im\_profile\_jvms
  - TARO: im\_profile\_taro
  - CTRL: im\_profile\_ctrl
  - KEYPAD: im\_profile\_keypad
3. To use the new key definition, restart the application.

If you include the definition in the user's .cshrc (or .profile if sh is in use), you will be able to use the selected key definition for subsequent sessions.

For details on key definitions, refer to the *User Key Definition User's Guide*.

## 7.4 Conversion Rules

This section defines the following conversion rules:

- Romaji-to-Kana Conversion (Section 7.4.1)  
Shows the correspondence between Roman and Kana characters that are used in typing the phones in Roman characters.
- Special-Character Conversion (Section 7.4.2)  
Shows the correspondence between symbols and special characters that are used in converting a character into a special character using Katakana Conversion.
- Symbol Conversion (Section 7.4.3)  
Shows the correspondence between the alphanumeric characters and symbols and the special characters that are used in converting a string of one to three alphanumeric characters or symbols into a special character.

- Code Input (Section 7.4.4)

Describes how to convert Kanji codes into the Kanji characters using Symbol Conversion.

### 7.4.1 Romaji-to-Kana Conversion Correspondence Table

Roman character <sup>a</sup>					Hiragana character				
a	i	u	e	o	あ	い	う	え	お
ka	ki	ku	ke	ko	か	き	く	け	こ
qa	qi	qu	qe	qo	くあ	くい	く	くえ	くお
sa	si	su	se	so	さ	し	す	せ	そ
ta	ti	tu	te	to	た	ち	つ	て	と
na	ni	nu	ne	no	な	に	ぬ	ね	の
ha	hi	hu	he	ho	は	ひ	ふ	へ	ほ
fa	fi	fu	fe	fo	ふあ	ふい	ふ	ふえ	ふお
ma	mi	mu	me	mo	ま	み	む	め	も
ya	yi	yu	ye	yo	や	い	ゆ	え	よ
ra	ri	ru	re	ro	ら	り	る	れ	ろ
la	li	lu	le	lo	ら	り	る	れ	ろ
wa	wi	wu	we	wo	わ	ゐ	う	ゑ	を
n					ん				
xa	xi	xu	xe	xo	あ	い	う	え	お

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

Roman character <sup>a</sup>					Hiragana character				
rya	ryi	ryu	rye	ryo	りゃ	りい	りゅ	りえ	りよ
lya	lyi	lyu	lye	lyo	りゃ	りい	りゅ	りえ	りよ
gya	gyi	gyu	gye	gyo	ぎゃ	ぎい	ぎゅ	ぎえ	ぎよ
gwa	gwi	gwu	gwe	gwo	ぐわ	ぐい	ぐ	ぐえ	ぐお
zya	zyi	zyu	zye	zyo	じゃ	じい	じゅ	じえ	じよ
jya	jyi	jyu	jye	jyo	じゃ	じい	じゅ	じえ	じよ
dya	dyi	dyu	dye	dyo	ぢゃ	ぢい	ぢゅ	ぢえ	ぢよ
dha	dhi	dhu	dhe	dho	でゃ	でい	でゅ	でえ	でよ
bya	byi	byu	bye	byo	びゃ	びい	びゅ	びえ	びよ
pya	pyi	pyu	pye	pyo	ぴゃ	ぴい	ぴゅ	ぴえ	ぴよ

<sup>a</sup>Supplements:

- Either uppercase or lowercase characters may be used for Roman characters.
- To type a double consonant character ("っ"), type the succeeding consonant character twice.
- If a "ん" is followed by a vowel, insert an apostrophe ("'") before the vowel.

## 7.4.2 Special-Character Conversion Correspondence Table

Before Conversion	After Conversion
;	、 (Touten)
.	。 (Kuten)
+	・ (Nakaten)
-	— (Chouon (Long Vowel))
¥	＼ (Reverse Slant)
'	' (Left Single Quotation Mark)
—	～ (Swung Dash)
@	◎ (Double Circle)
*	※ (Kome-jirushi)
=	＝ (Geta)
[	「 (Left Kagi-Kakko)
]	」 (Right Kagi-Kakko)
<	< (Right Angle Bracket)
>	> (Left Angle Bracket)

## 7.4.3 Symbol Conversion Correspondence Table

- One-character conversion

Before conversion:										
1	2	3	4	5	6	7	8	9	0	-
After conversion:										
┌	┐	└	┘	+	+	┌	┐	└	┘	—

- Two-Character Conversion

Before Conversion:

F1 F2 F3 F4 F5 F6 F7 F8 F9 F0 F-

After Conversion:

⊥ ⊥ ⊃ ⊂ ⊕ ⊖ ∫ ∫ ∫ ∫ ∫

Before Conversion:

T1 T2 T3 T4 T5 T6 T7 T8 T9 T0 T-

After Conversion:

⊥ ⊥ ⊃ ⊂ ⊕ ⊖ ∫ ∫ ∫ ∫ ∫

Before Conversion:

Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y0 Y-

After Conversion:

⊥ ⊥ ⊃ ⊂ ⊕ ⊖ ∫ ∫ ∫ ∫ ∫

Before Conversion:

,, ^I /] /^ || ,. .. '' ([ ]) << >> [[ ]]

After Conversion:

” 全 々 々 々 || … … “ [ ] 《 》 ∫ ∫

Before Conversion:

[( )] +- XX :- =/ <= >= OO .; O> O+ /\_ |\_

After Conversion:

[ ] ± × + ≠ ≤ ≥ ∞ ∴ ♂ ♀ ∠ ⊥

Before Conversion:

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

After Conversion:

∩ ∂ ∇ ≡ ≐ ≪ ≫ √ ∞ ∞ ∴ ∫ ∫ ∫ ∈

Before Conversion:  
 )- ( \_ )\_ (( )) CU CA /¥ ¥/ -| => == V- ]-  
 After Conversion:  
 ⊃ ⊆ ⊇ ⊂ ⊃ ∪ ∩ ∧ ∨ ¬ ⇒ ⇔ ∇ ∃

Before Conversion:  
 .1 '1 '2 .C C/ L- .A %% SS >< () <> [] <¥  
 After Conversion:  
 - ' " ° € £ Å % § ☆ ○ ◇ □ △

Before Conversion:  
 </ =| >- ><- |^ |v ## |. .^ +1 +2 ]| C)  
 After Conversion:  
 ∇ ㄒ → ← ↑ ↓ # ♭ ♯ † ‡ ¶ ○

- Three-Character Conversion

Before Conversion:  
 K>< K() K<> K[] K<¥ K</  
 After Conversion:  
 ★ ● ◆ ■ ▲ ▼

### 7.4.4 Code Input

Type the Kanji code for symbol conversion to input a Kanji character using a Kanji code. Two types of Kanji codes are available:

- DEC Kanji
- Ku-Ten

If you use DEC Kanji code, attach an “X” to the beginning.

For example, XA1B9 → 記号変換 → 々

If you use a Ku-Ten code, attach a “J” to the beginning.

For example, J125 → 記号変換 → 々



For the members of the DEC Kanji and Ku-Ten codes, see the book *Kanji Code Table* that is included in the Japanese version of the Tru64 UNIX Installation Kit.

## 7.5 Using the Keyboard Kana Mode

Tru64 UNIX supports Kana keyboards. After the Japanese Keyboard is setup on the system, a Kana keyboard is automatically available.

A Kana keyboard has two modes and you can toggle between the two. Kana Mode lets you type Kana characters. ASCII Mode lets you type alphabetic characters and digits. The default mode is ASCII.

To toggle between the ASCII and Kana Modes on the PCXAJ keyboard, press the **Ctrl** key located at the right.

---

### Note

---

The PCXAJ-AA is equipped with two **Ctrl** keys.

---

To toggle between the ASCII and Kana Modes on any other Japanese keyboard, press the **かな** (KANA) key.

On a North American keyboard, the **Compose** or **Comp** key functions as the **かな** (KANA) key.

## 7.6 Supporting Japanese Keyboards Equipped with Conversion Keys

Tru64 UNIX supports the following Japanese keyboards, which are equipped with keys dedicated to Japanese input (for example, **変換** (HENKAN), **無変換** (MU-HENKAN)):

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

For details about setting up the keyboards, refer to the Release Notes.

Any of the keyboards equipped with the **変換** (HENKAN) key allow you to use the keystrokes listed in Table 7-7, as well as the keystrokes based on the conversion key definition (JVMS, EVEJ, or KEYPAD).

**Note**

For the TARO Conversion Key Definition, the keystrokes listed in Table 7-7 are not available.

**Table 7-7: Conversion Key Definition of a Japanese Keyboard Equipped with the 変換 (HENKAN) Keys**

Function	Keystrokes <sup>a</sup>
Kanji Conversion or Next Phrase Candidate	変換
Kanji Candidate List	Alt+変換
Hiragana Conversion or Phrase Hiragana Conversion	ひらがな
Full-Width Katakana Conversion or Full-Width Phrase Katakana Conversion	カタカナ; カタカナ is to press the カタカナ/ひらがな key while holding down the Shift key
Full-Width Conversion	Alt+ひらがな
Half-Width Katakana Conversion or Half-Width Phrase Katakana Conversion	Ctrl+ひらがな or Ctrl+Shift+ひらがな
Half-Width Conversion	Ctrl+Alt+ひらがな
Symbol Conversion or Code Input	Alt+無変換
Move to Next Phrase	Shift+↓
Move to Previous Phrase	Shift+↑
Reduce Phrase	Shift+←
Enlarge Phrase	Shift+→
Previous Phrase Candidate	前候補; 前候補 is to press the 前候補/変換 key while holding down the Shift key
No Conversion	無変換
Cancel Conversion	Shift+無変換

<sup>a</sup> Ctrl+ひらがな is to press the ひらがな key while holding down the Ctrl key.

# A

---

## Character Codes

In Japan, there are several Kanji codesets, which are character codesets that contain character codes representing several thousand Kanji and non-Kanji characters. Thus, you must explicitly specify in the locale name the suitable Kanji codeset for the data you want to process. This chapter describes the Tru64 UNIX Japanese character sets and the following supported codesets:

- 7–Bit JIS Kanji
- 8–Bit JIS Kanji
- Japanese Extended UNIX Code (EUC)
- DEC Kanji
- Super DEC Kanji
- Shift JIS
- UTF-8

### A.1 Character Sets

A set of correspondences between characters and codes is called a character set. The Japanese Industrial Standard (JIS) specifies the character sets listed in Table A–1. Each character in these character sets is represented by a 1- or 2-byte character code. However, only the seven bits of each byte of a character are used; a 0 (zero) is contained in the most significant bit (MSB).

**Table A–1: Classification of Character Sets Based on the JIS**

Class	JIS Standard Number	Character Code
JIS Roman Character	JIS X 0201 LH	0xxxxxxx
JIS Katakana Characters	JIS X 0201 RH	0xxxxxxx
JIS Kanji Characters	JIS X 0208 <sup>a</sup>	0xxxxxxx 0xxxxxxx
JIS Supplemental Kanji Characters	JIS X 0212	0xxxxxxx 0xxxxxxx

<sup>a</sup> Unless otherwise noted, Tru64 UNIX refers to the 1983 JIS Kanji character set.

In addition to the character set based on the Japanese Industrial Standard (JIS), the ASCII character set, which contains 1-byte character codes defined by the American National Standards Institute (ANSI), is used widely in

Japan. The ASCII character set is almost the same as JIS X 0201H, except for the graphic characters assigned to the two codes listed in Table A-2.

**Table A-2: ASCII Versus JIS X 0201 LH**

Character Code	ASCII	JIS X 0201 LH
0x5c	Back Slash (\)	Yen Sign (¥)
0x7e	Tilde (~)	Overline ( ¯ )

A set of two or more of these character sets is called a codeset. However, within a codeset, character codes belonging to different character sets overlap and characters cannot be uniquely associated with codes. For this reason, some code extension is required for the character set included in the codeset.

The International Standards Organization (ISO) specifies ISO 2022, which defines the system of character codes common to ISO countries. The Japanese Industrial Standard translated ISO 2022 into the Japanese version numbered JIS X 0202. The following are the main codesets that were code-extended based on JIS X 0202:

- 7-Bit JIS Kanji
- 8-Bit JIS Kanji
- Japanese EUC

The following codesets were code-extended specific to Japan, but are not based on JIS X 0202:

- DEC Kanji
- Super DEC Kanji
- Shift JIS

Table A-3 lists the character sets available for each of the codesets. In the table, Yes means that the codeset contains characters included in each character set.

**Table A-3: Character Sets Available for Each of the Codesets**

Codeset	ASCII or JIS X 0201 LH	JIS X 0201 RH	JIS X 0208	JIS X 0212
7-Bit JIS	Yes	Yes	Yes	Yes
8-Bit JIS	Yes	Yes	Yes	Yes
Japanese EUC	Yes	Yes	Yes	Yes
DEC Kanji	Yes	No	Yes	No

**Table A-3: Character Sets Available for Each of the Codesets (cont.)**

Codeset	ASCII or JIS X 0201 LH	JIS X 0201 RH	JIS X 0208	JIS X 0212
Super DEC Kanji	Yes	Yes	Yes	Yes
Shift JIS	Yes	Yes	Yes	No

For details about each JIS standard, refer to the appropriate JIS table.

In ISO/IEC 10646, the International Standards Organization has defined the Universal Character Set (UCS), which represents all characters in the world by a single codeset. The Japanese Industrial Standard has translated the UCS into JIS X 0221.

Tru64 UNIX supports UTF-8 encoding based on ISO/IEC 10646.

### A.1.1 7-Bit JIS Kanji Codeset

In the 7-bit JIS Kanji codeset, all the characters included in the JIS X 0201 LH, 0202 RH, 0208, and 0212 character sets are encoded into 7-bit character codes. Throughout this document, the 7-Bit JIS Kanji codeset is called the 7-Bit JIS Codeset.

**Table A-4: 7-Bit JIS Codeset Character Codes**

Character Set	Character Code
JIS X 0201 LH	0xxxxxxx
JIS X 0201 RH	0xxxxxxx
JIS X 0208	0xxxxxxx 0xxxxxxx
JIS X 0212	0xxxxxxx 0xxxxxxx
C0 Control Character	000xxxx

The characters encoded as listed in Table A-4 overlap the codes of the other characters, and so, cannot be identified uniquely. For the 7-Bit JIS Codeset, the respective characters are identified by means of the code extension specified in JIS X 0202.

The codes are processed as follows:

- All the codes appearing after the Kanji in sequence ("ESC \$ B" by default) are processed as JIS X 0208 Kanji characters.
- All the codes appearing after the Kanji out sequence ("ESC(B" by default) are processed as ASCII characters.
- All the codes appearing after Supplemental Kanji in sequence ("ESC\$(D" by default) are processed as JIS X 0212 Supplemental Kanji characters.

- All the codes appearing after Kanji out sequence ("ESC(B" by default) are processed as ASCII characters.
- All the codes from SO (0x0e) to SI (0x0f) are processed as JIS X 0201 RH Katakana characters.

You also have the option to use the Kana in sequence ("ECS(I" by default), instead of SO (0x0e) and SI (0x0f), to represent JIS X0201 RH Katakana characters.

---

**Note**

---

The 7-Bit JIS Codeset can be used for the terminal codes, but cannot be used to specify a locale name.

---

For details about the 7-Bit JIS Codeset, see JIS7(5).

### A.1.2 8-Bit JIS Kanji Codeset

In the 8-Bit JIS Kanji codeset, all the characters included in the JIS X 0201 LH, 0208, and 0212 character sets are encoded into 7-bit character codes and all the characters included in the JIS X 0201 RH character set are encoded into 8-bit character codes. Throughout this document, the 8-Bit JIS Kanji codeset is called the 8-Bit JIS Codeset. Table A–5 summarizes encoding of the 8-Bit JIS Codeset characters.

**Table A–5: 8-Bit JIS Codeset Character Codes**

Character Set	Character Code
JIS X 0201 LH	0xxxxxxx
JIS X 0201 RH	1xxxxxxx
JIS X 0208	0xxxxxxx 0xxxxxxx
JIS X 0212	0xxxxxxx 0xxxxxxx
C0 Control Character	000xxxx

The codes are processed as follows:

- All the codes appearing after the Kanji in sequence ("ESC( B" by default) are processed as JIS X 0208 Kanji characters.
- All the codes appearing after the Kanji out sequence ("ESC\$B" by default) are processed as ASCII characters.
- All the codes appearing after Supplemental Kanji in sequence ("ESC\$(D" by default) are processed as JIS X 0212 Supplemental Kanji characters.

- All the codes appearing after Kanji out sequence ("ESC(B" by default) are processed as ASCII characters.
- Data with a 1 in the MSB is processed as a JIS X 0201 RH Katakana character.
- SI and SO, used for the 7-Bit JIS Codeset, are ignored.

---

**Note**

---

The 8-Bit JIS Codeset can be used for the terminal codes, but cannot be used to specify a locale name.

---

For details about the 8-Bit JIS Codeset, see JIS8(5).

### A.1.3 Japanese EUC Codeset

The Extended UNIX Code (EUC) is a coding scheme that was extended by AT&T Bell Laboratories and made available worldwide. The Japanese EUC codeset is the result of applying the EUC to Japanese. The extension is based on the UNIX Internal Coding Scheme for Japanese that was defined in the UNIX System Japanese Features Proposal, proposed in 1985 by the AT&T Japanese UNIX System Consultative Committee.

The EUC has four codesets:

- CS0 — Any character belonging to CS0 is encoded so that the MSB is always set to 0 (zero). In any EUC worldwide, CS0 is defined as the ASCII character set.
- CS1 — Any character belonging to CS1 is encoded so that the MSB of each byte of the code is always set to 1. In the Japanese EUC Codeset, CS1 is JIS X 0208.
- CS2 — Any character belonging to CS2 is encoded so that, following the SS2 code (0x8f), the MSB of each byte of the code is always set to 1. In the Japanese EUC Codeset, CS2 is JIS X 0201 RH.
- CS3 — Any character belonging to CS3 is encoded so that, following the SS3 code (0x8f), the MSB of each byte of the code is always set to 1. In the Japanese EUC Codeset, CS3 is JIS X 0212.

Table A–6 summarizes encoding of the Japanese EUC Codeset characters.

**Table A–6: Japanese EUC Codeset Character Codes**

Code Set	Corresponding Character Set	Encoding
CS0	ASCII	0xxxxxxx
CS1	JIS X 0208	1xxxxxxx 1xxxxxxx
CS2	JIS X 0201RH	SS2 1xxxxxxx
CS3	JIS X 0212	SS3 1xxxxxxx 1xxxxxxx
C0 Control Character		000xxxxx
C1 Control Character		100xxxxx

The following fields of the undefined JIS X 0208 and 0212 fields can be assigned as the user-defined character fields by the vendor or user:

- 85th to 94th Ku fields of the undefined JIS X 0208 fields
- 78th to 94th Ku fields of the undefined JIS X 0212 fields

---

**Note**

---

JIS reserves the 78th to 84th fields of the undefined JIS X 0212 fields for future extension. Thus, it is possible for the JIS to assign characters to these fields that will collide with user-defined characters.

---

For details about the Japanese EUC Codeset, see `eucJP(5)`.

#### A.1.4 DEC Kanji Codeset

The DEC Kanji Codeset consists of the following character sets:

- ASCII or JIS X 0201 LH
- DEC Kanji Character Set 1983
- User-defined fields in DEC Extended Kanji Character Set

An ASCII character or JIS X 0201 LH is represented by a 1-byte character code. The MSB of each byte is always 0 (zero).

DEC Kanji Character Set 1983 consists of a total of 6,877 characters, which include non-Kanji characters and Level-1 and Level-2 Kanji characters (see Table A–7). One character is represented by 2 bytes, and the MSB of each byte is always 1. The Ku-Ten numbers of this character set are the same as JIS X 0208-1983.



**Table A–7: DEC Kanji Character Set 1983**

<b>Ku Number</b>	<b>Assignment</b>	<b>Characters</b>
1st to 8	Non-Kanji characters including special characters, digits, Roman characters, Hiragana characters, Katakana characters, Greek characters, Russian characters, and ruled lines	524
9th to 15th	JIS-reserved fields	
16th to 47th	Level-1 Kanji characters	2,965
48th to 84th	Level-2 Kanji characters	3,388
85th to 94th	JIS-reserved fields	

In the DEC Kanji Codeset, codes of 8,836 user-defined characters can be assigned to user-defined fields in the DEC Extended Kanji Character Set. Any character belonging to this extended character set contains a 1 in the MSB in the first byte and a 0 (zero) in the MSB in the second byte. The 32nd to 94th Ku fields are reserved by HP. Therefore, exceptional or user-defined characters can be assigned to the 1st to the 31st Ku fields in the DEC Extended Kanji Character Set.

Table A–8 summarizes DEC Kanji Codeset character codes.

**Table A–8: DEC Kanji Codeset Character Codes**

<b>Character Set</b>	<b>Character Code</b>
ASCII or JIS X 0201 LH	0xxxxxxx
JIS X 0208	1xxxxxxx 1xxxxxxx
DEC Extended Kanji Character Set	1xxxxxxx 0xxxxxxx
C0 Control Character	000xxxxx
C1 Control Character	100xxxxx

For details about the DEC Kanji Codeset, see the Kanji Code Test and `deckanji(5)`.

### A.1.5 Super DEC Kanji Codeset

The Super DEC Kanji Codeset is an extended version of the DEC Kanji Codeset that enables CS2 and CS3 of the Japanese EUC to be handled.

The Super DEC Kanji Codeset consists of the following character codesets:

- ASCII or JIS X 0201 LH
- DEC Kanji Character Set 1983
- DEC Extended Kanji Character Set
- JIS X 0201 RH
- JIS X 0212-1990

The ASCII (or JIS X 0201 LH) codeset, DEC Kanji Character Set 1983, and DEC Extended Kanji Character Set are encoded in the same manner as the DEC Kanji Codeset.

Any character belonging to JIS X 0201 RH is represented by the 1-byte character data that follows SS2 (0x8e). The MSB of this byte is always 1.

Any character belonging to JIS X 0212-1990 is represented by the 2-byte character data that follows SS3 (0x8f). The MSB of this byte is always 1.

Table A-9 summarizes the Super DEC Kanji Codeset character codes.

**Table A-9: Super DEC Kanji Codeset Character Codes**

Character Set	Character Code
ASCII or JIS X 0201 LH	0xxxxxxx
JIS X 0208	1xxxxxxx 1xxxxxxx
DEC Extended Kanji Character Set	1xxxxxxx 0xxxxxxx
JIS X 0212 RH	SS2 1xxxxxxx
JIS X 0212	SS3 1xxxxxxx 1xxxxxxx
C0 Control Character	000xxxxx
C1 Control Character	100xxxxx

In the Super DEC Kanji Codeset, exceptional or user-defined characters can be assigned to fields ranging from the first field to the undefined fields in the DEC Extended Kanji Character Set, JIS X0208, and JIS X 0212 (see Table A-10).

**Table A-10: Super DEC Kanji User-Defined Fields**

Character Set	Assignable Ku Fields
JIS X 0208	85th to 94th
JIS X 0212	78th to 94th
DEC Extended Kanji Character Set	1st to 94th

For details about the Super DEC Kanji Codeset, see `sdeckanji(5)`.

## A.1.6 Shift JIS Codeset

The Shift JIS Codeset is widely used in the personal computer world. Table A-11 summarizes encoding of its character sets.

**Table A-11: Encoding of Shift JIS Codes**

Shift JIS Character Set	Encoding
C0 Control Characters	0x00 to 0x1f
Space	0x20
JIS X 0201 LH	0x21 to 0x7e
Del	0x7f
Undefined	0x80
First byte of JIS X 0208 Kanji character	0x81 to 0x9f
Undefined	0xa0
JIS X 0201 RH	0xa1 to 0xdf
First byte of JIS X 0208 kanji character	0xe0 to 0xef
First byte of user-defined character	0xf0 to 0xfc
Undefined	0xfc to 0xff

Table A–12 summarizes the correspondence between the codes and Kanji characters and the correspondence between the user-defined characters and Ku-Ten numbers.

**Table A–12: Correspondence Between Kanji and User-Defined Characters and Ku-Ten Numbers**

<b>Kanji and User-Defined Characters</b>	<b>Ku-Ten Numbers</b>
0x8140 ... 0x819c	JIS X 0208, 1st Ku, 1st Ten to 1st Ku, 94th Ten (except 0x817f)
0x819f ... 0x81fc	JIS X 0208, 2nd Ku, 1st Ten to 2nd Ku, 94th Ten
0x8240 ... 0x829e	JIS X 0208, 3rd Ku, 1st Ten to 3rd Ku, 94th Ten (except 0x827f)
0x8140 ... 0x819c	JIS X 0208, 4th Ku, 1st Ten to 4th Ku, 94th Ten
⋮	⋮
0x9f9f ... 0x9ffc	JIS X 0208, 62nd Ku, 1st Ten to 62nd Ku, 94th Ten (except 0x817f)
0xe040 ... 0xe09e	JIS X 0208, 63rd Ku, 1st Ten to 63rd Ku, 94th Ten (except 0xe07f)
⋮	⋮
0xef40 ... 0xef9e	JIS X 0208, 93rd Ku, 1st Ten to 93rd Ku, 94th Ten (except 0x817f)
0xef9f ... 0xeffc	JIS X 0208, 94th Ku, 1st Ten to 94th Ku, 94th Ten (except 0xe07f)
0xf040 ... 0xf0fc	User-defined characters (except 0xf07f)
⋮	⋮
0xfc40 ... 0xfcfc	User-defined characters (except 0xfc7f)

In the Shift JIS Codeset, user-defined characters can be assigned to 0xf040 to 0xfcfc (unless the second byte is one of 0x00 to 0x3f, 0x7f, and 0xfd to 0xff).

For details about Shift JIS Codeset, see `shiftjis(5)`.

### A.1.7 UTF-8 Codeset

The UTF-8 codeset encodes the UCS-4 (4 octets) and UCS-2 (2 octets) data defined in ISO/IEC 10646 into a single codeset to represent the universal characters. UTF-8 codes are represented as variable-length codes in UCS interchange format and represent any ASCII code by one byte. In addition,

the standard gave consideration to not using an ASCII code for a multibyte variable-length code field.

**Table A–13: UCS-4 Encoding Ranges and UTF-8 Bit Assignments**

Encoding Range (Hex)	UTF-8 Octet String (Binary)
0000 0000-0000 007F	0xxxxxxx
0000 0080-0000 07FF	110xxxxx 10xxxxxx
0000 0800-0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000-001F FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
0020 0000-03FF FFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
0400 0000-7FFF FFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

For details about UTF-8, see Unicode(5).



# B

---

## Contents of Japanese Locale Definitions

The Tru64 UNIX operating system offers the following Japanese locales: `ja_JP.eucJP`, `ja_JP.SJIS`, `ja_JP.deckanji`, `ja_JP.sdeckanji`, and UTF-8. The contents of the definitions of the four `ja_JP.*` Japanese locales are identical, except for the codeset-dependent definitions. The Japanese locales meet the definitions in the UI-OSF Japanese Environment Installation Conventions (1993).

This chapter describes the definitions of `CHARMAP`, `LC_CTYPE`, `LC_COLLATE`, `LC_MESSAGES`, `LC_MONETARY`, `LC_NUMERIC`, and `LC_TIME` environment variables. For details on the meanings of items and keywords defined in the locale, see `charmap(4)` and `locale(4)`.

### B.1 CHARMAP

The `charmap` file referenced during creation of a locale database exists for each of the codesets. It describes the correspondence between the symbolic names and code values that are used for the codeset attributes and locale definitions. Table B-1 lists the attributes of the Japanese codesets that are displayed with the `locale -k CHARMAP` command. The attributes are listed by `charmap` file keyword. In the table, `code_set_name` is the value of the locale in program execution. To obtain the appropriate value, invoke `n1_langinfo(CODESET)`.

**Table B-1: Japanese Codeset Attributes**

Attributes	Japanese EUC	Shift JIS	DEC Kanji	Super DEC Kanji	UTF-8
<code>code_set_name</code>	"eucJP"	"SJIS"	"deckanji"	"sdeckanji"	"UTF-8"
<code>mb_cur_max</code>	3	2	2	3	3
<code>mb_cur_min</code>	1	1	1	1	1

### B.2 LC\_TYPE

The `LC_TYPE` environment variable contains the definitions for the character class that determines the action of the `isw*( )` function. It also contains the character conversion information that determines the action of `tow*( )` function.

There are two categories of character classes: standard and extended. The standard character classes are specified in standard specifications such as the Programming Language C and POSIX standards. These classes always exist in a system conforming to the standard of XPG4 or POSIX.

The extended character classes include required character classes that are also defined for each of the locales. These classes include those character classes required for the Japanese locales that are not specified in the standard specification.

Table B-2 summarizes the contents of the definitions of the standard character classes. Table B-3 summarizes the contents of the definitions of the extended character classes. (For UTF-8, no extended character classes are undefined.) To make a decision for an extended character class by using a program, combine the `wctype( )` and `iswctype( )` functions.

**Table B-2: Standard Character Classes**

Character Class Name	Contents of Definition
upper	ASCII 'A' to 'Z', JIS X 0208 Roman uppercase characters, JIS X 0208 Greek uppercase characters, JIS X 0208 Russian uppercase characters, JIS X 0212 Greek uppercase characters with diacritical marks, JIS X 0212 Cyrillic alphabetic uppercase character, JIS X 0212 Latin Alphabetic lowercase characters, JIS X 0212 Latin Alphabetic uppercase characters with diacritical marks
lower	ASCII 'a' to 'z', JIS X 0208 Roman lowercase characters, JIS X 0208 Greek lowercase characters, JIS X 0208 Russian lowercase characters, JIS X 0212 Greek lowercase characters with diacritical marks, JIS X 0212 Cyrillic alphabetic lowercase characters, JIS X 0212 Latin Alphabetic lowercase characters, JIS X 0212 Latin Alphabetic lowercase characters with diacritical marks
alpha	Characters defined in upper and lower
digit'	ASCII '0' to '9'
space	ASCII ' ', '\t' (0/9), '\n' (0/10), '\v' (0/11), '\f' (0/12), '\r (0/13)', and space in JIS X 0208 (1st Ku, 1st Ten)
cntrl	C0 and C1 control characters (except SS2 and SS3 for eucJP and sdeckanji)



**Table B-2: Standard Character Classes (cont.)**

Character Class Name	Contents of Definition
punct	<ul style="list-style-type: none"> <li>• ASCII '!' (2/1) to '/' (2/15), ':' (3/10) to '@' (4/0), '[' (5/11) to '^' (6/0), '`' (7/11) to '~' (7/14)</li> <li>• JIS X 0208 ' ` ' (1st Ku, 2nd Ten) to ' ! ' (1st Ku, 10th Ten), ' ^ ' (1st Ku, 13th Ten) to ' _ ' (1st Ku, 18th Ten), ' " ' (1st Ku, 23rd Ten), ' &lt; ' (1st Ku, 26th Ten), ' → ' (1st Ku, 29th Ten), to ' ◇ ' (1st Ku, 94th Ten)</li> <li>• JIS X 0201 RH Kuten (2/1), left parenthesis (2/2), right parenthesis (2/3), Touten (2/4), Nakaten (2/5)</li> <li>• JIS X 0202 diacritical marks (2nd Ku, 15th Ten to 25th Ten), description symbols (2nd Ku, 34th Ten; 2nd Ku, 36th Ten), unit symbol (2nd Ku, 80th Ten), ordinary symbols (2nd Ku, 35 Ten; 75th to 79th Ten, 81st Ten)</li> </ul>
graph	<ul style="list-style-type: none"> <li>• Characters included in ASCII upper, lower, digit, and punct</li> <li>• All characters (including undefined fields) other than JIS X 0208 space</li> <li>• All JIS X 0212 RH characters (2/1 to 5/15)</li> <li>• All JIS X 0212 characters (including undefined fields)</li> <li>• All characters in the user-defined character fields</li> </ul>
print	<ul style="list-style-type: none"> <li>• All characters included in graph</li> <li>• ASCII ' ' (2/0)</li> <li>• JIS X 0208 space (1st Ku, 1st Ten)</li> </ul>
xdigit	ASCII '0' to '9', 'a' to 'f', 'A' to 'F'
blank	ASCII '\t', '\n' (0/9)

**Table B-3: Extended Character Classes**

Character Class Name	Contents of Definition
ascii	All ASCII characters
line	JIS X 0208 ruled lines (8th Ku, 1st Ten to 8th Ku, 32nd Ten)
jdigit	JIS X 0208 digits ('0' to '9')
paren	<ul style="list-style-type: none"> <li>• ASCII '(', ')', '[', ']', '{', '}'</li> <li>• JIS X 0208 aggregation signs (1st Ku, 38th Ten to 1st Ku, 59th Ten)</li> <li>• JIS X 0201 RH left parenthesis (2/2), right parenthesis (2/3)</li> </ul>

**Table B-3: Extended Character Classes (cont.)**

Character Class Name	Contents of Definition
jparen	<ul style="list-style-type: none"> <li>JIS X 0208 aggregation signs (1st Ku, 38th Ten to 1st Ku, 59th Ten)</li> <li>JIS X 0201 RH left parenthesis (2/2), right parenthesis (2/3)</li> </ul>
jisx0201	All JIS X 0201 characters
jisx0201r	All JIS X 0201 RH characters
jisx0208	All JIS X 0208 characters (JIS-defined fields only)
jisx0212	All JIS X 0212 characters (JIS-defined fields only)
udc	User-defined characters
vdc	Vendor-defined characters (undefined because this does not exist in Tru64 UNIX)
gaiji	User- and vendor-defined characters
jhira	JIS X 0208 'ぁ' (4th Ku, 1st Ten) to 'ん' (4th Ku, 83rd Ten), 'ゝ' (1st Ku, 11th Ten), 'ゞ' (1st Ku, 12th Ten), 'ゝ' (1st Ku, 21st Ten), 'ゞ' (1st Ku, 22nd Ten), 'ゝ' (1st Ku, 28th Ten)
jkata	JIS X 0208   'ァ' (5th Ku, 1st Ten) to 'ヶ' (5th Ku, 86th Ten), 'ヰ' (1st Ku, 11th Ten), 'ヱ' (1st Ku, 12th Ten), 'ヰ' (1st Ku, 19th Ten), 'ヱ' (1st Ku, 20th Ten), 'ヰ' (1st Ku, 28th Ten)
jhankana	JIS X 0201 RH Katakana (2/6 to 5/13), voiced sound symbol (5/14), semivoiced sound symbol (5/15)
jkanji	<ul style="list-style-type: none"> <li>JIS X 0208 Kanji (16th to 84th JIS-defined characters), '仝' (1st Ku, 24 Ten), '々' (1st Ku, 25 Ten), '〇' (1st Ku, 27 Ten)</li> <li>JIS X 0212 Kanji (16th to 77th Ku JIS-defined characters)</li> </ul>
jspace	JIS X 0208 space (1st Ku, 1st Ten)
english	JIS X 0208 'A' to 'Z', 'a' to 'z'
number	JIS X 0208 '0' to '9'
special	JIS X 0208 special characters (1st to 2nd Ku JIS-defined characters)
phonogram	JIS X 0208 'ぁ' (4th Ku, 1st Ten) to 'ん' (4th Ku, 83rd Ten), 'ァ' (5th Ku, 1st Ten) to 'ヶ' (5th Ku, 86th Ten)
ideogram	All JIS X 0208 characters (including undefined JIS fields)

The LC\_TYPE environment variable defines the correspondence between uppercase and lowercase characters referenced from the toupper( )

and `tolower( )` functions. The following list describes the alphabetic characters defined in `toupper( )` and `tolower( )`.

- ASCII Roman characters (lowercase 'a' to 'z' <==> uppercase 'A' to 'Z')
- JIS X 0208 Roman characters (lowercase 'a' to 'z' <==> uppercase 'A' to 'Z')
- JIS X 0208 Greek characters (lowercase 'α' to 'ω' <==> uppercase 'Α' to 'Ω')
- JIS X 0208 Russian characters (lowercase 'а' to 'я' <==> uppercase 'А' to 'Я')
- JIS X 0212 Greek alphabet with diacritical marks
- JIS X 0212 Cyrillic alphabet
- JIS X 0212 Latin alphabet
- JIS X 0212 Latin alphabet with diacritical marks

### B.3 LC\_COLLATE

The `LC_COLLATE` environment variable contains the definitions for the character collating sequence. The collating sequence for the characters defined in the Japanese locales uses the following order:

1. ASCII characters
2. JIS X 0201 RH characters
3. JIS X 0208 characters (including undefined JIS fields)
4. JIS X 0221 characters (including undefined JIS fields)
5. User-defined characters of DEC Extended Kanji fields

In each character set, the characters are sequenced in ascending order of the codes.

### B.4 LC\_MESSAGES

When you enter a character string to a question that forces a selection of either YES (affirmative) or NO (negative), the `LC_MESSAGES` environment variable defines which character string is affirmative and which is negative. Use a regular expression for this definition. In the Japanese locales, the following definitions have been adopted:

- Any character string beginning with 'y' or 'Y' in the ASCII or JIS X 0208 alphabet is determined as YES.
- Any character string beginning with 'n' or 'N' in the ASCII or JIS X 0208 alphabet is determined as NO.

Table B-4 summarizes the contents of the definitions of `LC_MESSAGES`.

**Table B-4: LC\_MESSAGES Definitions**

Keyword	Contents of Definition
yesexpr	"^[yY£ù£Ù]"
noexpr	"^[nN£ì£Î]"

## B.5 LC\_MONETARY

The LC\_MONETARY environment variable contains the definitions of the characters and rules that are used to represent an amount of money. Table B-5 summarizes the contents of the LC\_MONETARY definitions.

**Table B-5: Definitions of LC\_MONETARY**

Keyword	Contents of Definition
int_curr_symbol	"JPY "
currency_symbol	"¥"
mon_decimal_point	""
mon_thousands_sep	","
mon_grouping	3
positive_sign	""
negative_sign	"-"
int_frac_digits	0
frac_digits	0
p_cs_precedes	1
p_sep_by_space	0
n_cs_precedes	1
n_sep_by_space	0
p_sign_posn	1
n_sign_posn	4

## B.6 LC\_NUMERIC

The LC\_NUMERIC environment variable contains the definitions of the characters and rules that are used to represent numeric values other than an amount of money. Table B-6 summarizes the contents of the LC\_NUMERIC definitions.

**Table B-6: Definitions of LC\_NUMERIC**

<b>Keyword</b>	<b>Contents of Definition</b>
decimal_point	."
thousands_sep	","
grouping	3

## **B.7 LC\_TIME**

The `LC_TIME` environment variable contains the definitions of the character strings and rules that are used to represent date and time information. The following table summarizes the contents of the `LC_TIME` definitions. The contents of the definitions of each keyword are referenced from the conversion specification for the `strftime( )` function listed in the table.

Keyword	strftime Function Conversion Specification	Contents of Definition
abday	%a	"日";"月";"火";"水";"木";"金";"土"
day	%A	"日曜日";"月曜日";"火曜日";"水曜日";"木曜日";"金曜日";"土曜日"
abmon	%b	"1月";"2月";"3月";"4月";"5月";"6月";"7月";"8月";"9月";"10月";"11月";"12月"
mon	%B	"1月";"2月";"3月";"4月";"5月";"6月";"7月";"8月";"9月";"10月";"11月";"12月"
d_t_fmt	%c	"%Y年%m月%d日 %H時%M分%S秒"
d_fmt	%x	"%Y年%m月%d日"
t_fmt	%X	"%H時%M分%S秒"
am_pm	%p	"午前";"午後"
t_fmt_ampm	%r	"%p%H時%M分%S秒"
era	-	<ul style="list-style-type: none"> <li>• "+:2:1990/01/01:+*:平成:%EC%Ey年";\</li> <li>• "+:1:1989/01/08:1989/12/31:平成:%EC元年";\</li> <li>• "+:2:1927/01/01:1989/01/07:昭和:%EC%Ey年";\</li> <li>• "+:1:1926/12/25:1926/12/31:昭和:%EC元年";\</li> <li>• "+:2:1913/01/01:1926/12/24:大正:%EC%Ey年";\</li> <li>• "+:1:1912/07/30:1912/12/31:大正:%EC元年";\</li> <li>• "+:2:1869/01/01:1912/07/29:明治:%EC%Ey年";\</li> <li>• "+:1:1868/09/08:1868/12/31:明治:%EC元年";\</li> <li>• "+:1:1/1/1:1868/09/07:西暦:%EC%Ey年";\</li> <li>• "-:1:-1/12/31:-*:紀元前:%EC%Ey年"</li> </ul>
era_year	%EY	"%EC%Ey年"
era_d_fmt	%Ex	"%EY%m月%d日"
era_d_t_fmt	%Ec	"%EY%m月%d日 %H時%M分%S秒"

---

# Index

## Numbers and Special Characters

---

- 7-Bit JIS Kanji codeset**, A-3
  - processing sequence, A-3
- 8-Bit JIS Kanji codeset**, A-4
  - processing sequence, A-4
- 8-bit data**
  - handling, 2-9

## A

---

- Advanced Printing Software**
  - setting up, 4-3
- Application codes**, 2-8, A-1
  - setting, 2-8
- Application codeset**
  - setting, 2-9
- ASCII character set**, A-1
- atty terminal driver**, 2-8

## C

---

- C locale**, 2-7
- CDE**
  - changing default language, 2-5
  - Japanese input server (dxjim), 1-6
  - LANG environment variable, 2-4
  - setting up Japanese keyboard, 5-1
  - setting up printer, 4-2
  - setting user language, 2-4
- cedit utility**
  - user-defined characters, 1-6
- cgen utility**
  - files for font loading, 1-6
- Character code**, A-1

- Character set**, A-1
- CHARMAP**, B-1
- Code conversion**
  - and user-defined characters, 2-11
- Code input**, 7-20
- Codeset**, 1-2
  - 7-Bit JIS Kanji, A-3
  - 8-Bit JIS Kanji, A-4
  - DEC Kanji, A-6
  - Japanese EUC, A-5
  - overlapping characters, A-2
  - Shift JIS, A-9
  - specifying, 1-2
  - Super DEC Kanji, A-7
  - superset relationships, 1-2
  - UTF-8, A-10
- Codeset names**
  - list of, 2-11
- Collating sequence**
  - LC\_COLLATE definition, B-5
  - order used by Japanese locales, B-5
- Commands**
  - categories that differ by locale, 2-6
- Common Desktop Environment**
  - ( *See* CDE )
- Conversion**
  - Kanji code, 7-15
  - Romaji to Kana, 7-14
  - special character, 7-14
  - symbol, 7-14
- Conversion key**
  - changing the definition, 7-14
  - CTRL, 7-8
  - definition of Japanese keyboards, 7-22
  - definitions, 7-7

EVEJ, 7-7  
for Japanese input, 7-7  
JVMS, 7-7  
KEYPAD, 7-8  
TARO, 7-8

**Conversion rules**

for characters and symbols, 7-14

**CTRL Conversion key**, 7-8

**CTRL Conversion Key Definition**,  
7-13

---

## D

**Date**

LC\_TIME definition, B-7

**DEC Kanji codeset**, A-6

character sets, A-6

**dxjim**

changing determination key, 7-4  
entering Japanese characters, 3-1  
input server, 7-1  
input styles, 7-2  
Japanese input server, 3-1  
starting, 3-1  
using, 7-2

**dxjim input server**, 1-6

changing conversion key, 7-3  
changing start and end keys, 7-3  
setting up, 7-3

**dxjim input style**

OverTheSpot default, 7-4  
setting with XmText, 7-4

**DXmCSText**

entering Japanese characters in  
OSF/MOTIF, 3-1

---

## E

**Environment variables**

for international environment, 2-3  
I18NPATH, 2-3  
LANG, 2-6  
LC\_ALL, 2-7  
MANPATH, 2-13

PATH, 2-3  
script files for, 2-3  
**/etc/i18n\_login**, 2-3  
**/etc/i18n\_path**, 2-3  
**/etc/i18n\_profile**, 2-3

**EUC**

codesets, A-5  
setting alternative Japanese locale,  
2-5

**EVEJ Conversion key**, 7-7

**EVEJ Conversion Key Definition**,  
7-10

**EVEJ key**, 7-7

**Extended character classes**, B-2

**Extended UNIX Code**

( See EUC )

---

## F

**FDIO utility**

accessing floppy disk drives, 1-4

**File code conversion**, 2-11

**Font loading**

cgen utility, 1-6

---

## I

**iconv function**

converting file codes, 2-11  
format, 2-11

**IMLIB library**

key definitions, 1-5

**Input server**

dxjim, 1-6, 3-1, 7-1  
starting dxjim, 3-1

**Installation**

subsets for Japanese environment,  
2-1

**Installation kits**

for Japanese features, 2-2  
full-text search tools, 2-2  
VJE input system, 2-2  
WX3 input system, 2-2

**Internationalization environment**



setting up, 2-3  
virtual root directory, 2-3  
**isw\* function**  
LC\_TYPE definition, B-1

## J

---

**Japanese characters**  
input of, 7-1  
**Japanese codeset attributes**, B-1  
**Japanese environment**  
basic setup, 2-1  
installation subsets, 2-1  
installing, 2-1  
**Japanese EUC codeset**, A-5  
**Japanese features**  
installation subsets for, 2-2  
**Japanese locale definitions**, B-1  
**Japanese tty**  
( See tty subsystem )  
**JVMS conversion key**, 7-7  
**JVMS Conversion Key Definition**,  
7-8  
**JVMS key**, 7-7

## K

---

**Kana characters**  
setting up keyboard, 5-2  
**Kana keyboard**, 7-21  
**Kana mode**, 7-21  
**Kana-Kanji conversion**, 1-3  
dictionaries used with, 1-4  
keyboard settings, 5-2  
kkseq (sequency key), 1-5  
phrase learning data file, 1-4  
**Kanji code conversion**, 1-5, 7-15  
**Kanji codes**, 4-1, A-1  
for terminals and applications, 2-8  
symbol conversion, 7-20  
**Key conversion**  
IMLIB library, 1-5

## Keyboard

console firmware, 5-3  
conversion keys, 5-2  
Japanese input conversion keys,  
7-21  
Kana mode, 5-2  
Kana modes, 7-21  
list of supported, 5-1  
setting up Japanese, 5-1  
single user mode, 5-3  
**KEYPAD Conversion key**, 7-8  
**KEYPAD Conversion Key  
Definition**, 7-11  
**KEYPAD key**, 7-7

## L

---

**LANG**  
environment variable under CDE,  
2-4  
**Language**  
changing system default, 2-5  
setting system to Japanese, 2-4  
**LC\_ALL**  
environment variable, 2-7  
**LC\_COLLATE**, B-5  
**LC\_CTYPE**, B-1  
**LC\_MESSAGES**, B-5  
**LC\_MONETARY**, B-6  
**LC\_NUMERIC**, B-6  
**LC\_TIME**, B-7  
**Libraries**  
categories that differ by locale, 2-6  
**Locale**  
basic default, 2-7  
categories of commands and  
libraries, 2-6  
setting up Japanese, 2-5  
specifying Japanese, 2-6  
specifying name, 2-5  
view settings, 2-7  
**Locale database**, 2-6

## Locale definitions

- CHARMAP, B-1
- LC\_COLLATE, B-5
- LC\_MESSAGES, B-5
- LC\_MONETARY, B-6
- LC\_NUMERIC, B-6
- LC\_TIME, B-7
- LC\_TYPE, B-1

## Locales

- dynamically switching between, 1-3
- list of available Japanese, 2-5
- supported on Tru64 UNIX, 1-2

## login file

- settings, 2-3

## lpr command, 4-1

## lprsetup command, 4-1

## M

---

### MANPATH

- environment variable, 2-13

### Message catalogs, 2-12

- installed directory paths, 2-12

### midoc, 6-1

- installation kit for, 2-2
- installing, 6-1
- setup, 6-2
- text searching tool, 1-6
- web browser requirements, 6-1

### miman, 6-1

- installation kit for, 2-2
- installing, 6-5
- reference page searching tool, 1-6
- setup, 6-6
- web browser requirements, 6-1

### MitakeSearch for online documents

- ( See midoc )

### MitakeSearch for reference pages

- ( See miman )

### Money

- units defined in LC\_MONETARY, B-6

## Mule editor, 1-4

## N

---

### Numbers

- LC\_NUMERIC definitions, B-6

## O

---

### ODL

- using with tty subsystem, 1-3

### On-Demand Font Loading, 1-3

- ( See also ODL )

### On-the-spot

- dxjim input style, 7-2

### OSF/MOTIF

- entering Japanese characters, 3-1

### Over-the-spot

- dxjim input style, 7-2

## P

---

### Path

- settings for, 2-3

### PATH

- for international environment, 2-3

### Printer

- Advanced printing software, 4-3
- Advanced Printing Software, 1-5
- as a terminal, 1-5
- configuration utility, 4-1
- filters for, 1-5
- lprsetup command, 4-1
- outputting a file, 4-1
- setting up Japanese, 4-1
- using ESC/P, 4-3

### Printer codes, 4-1

## R

---

### Reference pages

- setting up Japanese, 2-13
- viewing English, 2-14

### Romaji to Kana conversion, 7-14

correspondence table, 7–15

**Root directory**

for international environment, 2–3

**Root-window**

dxjim input style, 7–2

---

## S

**Shift JIS codeset**, A–9

**Special character conversion**,  
7–14

correspondence table, 7–18

**Standard character classes**, B–2

**Startup scripts**, 2–3

**stty command**

acode option, 2–9

changing settings, 2–8

imode option, 2–10

jdec to set line discipline, 2–7

tcode option, 2–9

viewing current settings, 2–10

**Super DEC Kanji codeset**, A–7

character codesets, A–8

**Symbol conversion**, 7–14

correspondence table, 7–18

**SysMan menu**

printer configuration, 4–1

**System V MNLS**

migration to Tru64 UNIX, 1–6

---

## T

**TARO Conversion key**, 7–8

**TARO Conversion Key Definition**,  
7–12

**TARO key**, 7–7

**Terminal**

setting up Japanese, 2–7

**Terminal codes**, 2–8, A–1

setting, 2–8

**Terminal codeset**

setting, 2–9

**Terminal driver**, 2–8

settings, 2–8

**Terminal settings**

checking current, 2–10

**Terminals**

tty subsystem, 1–3

**Time**

LC\_TIME definition, B–7

**tow\* function**

LC\_TYPE definition, B–1

**Tru64 UNIX**

Japanese features, 1–1

locales supported, 1–2

migration from System V MNLS,  
1–6

supported codesets, A–1

text searching tools, 1–6

**tty subsystem**, 1–3

code conversion support, 1–3

features of, 1–3

selecting Japanese line discipline,  
1–3

using with ODL, 1–3

---

## U

**User profile**

settings, 2–3

**User-defined characters**

and code conversion, 2–11

credit utility, 1–6

**/usr/sbin/i18nsetup**, 2–3

**UTF-8 codeset**, A–10

---

## V

**VJE input system**

installation kit for, 2–2

---

## W

**Wnn Kana-Kanji conversion**, 1–4

**WX3 input system**  
installation kit for, 2-2

entering Japanese characters in  
OSF/MOTIF, 3-1

## **X**

---

**XmText**