DIGITAL UNIX

Technical Reference for Using Japanese Features

October 1998

This guide provides the Japanese-specific information and describes the Japanese features supported on the DIGITAL UNIX system.

Software Version: Version 4.0E

Compaq Computer Corporation

October 1998

Digital Equipment Corporation makes no representation that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from DIGITAL or an authorized sublicensor.

No responsibility is assumed for the use or reliability of software on equipment that is not supported by Digital Equipment Corporation or its affiliated companies.

Table of Contents

Preface

Figures

Tables

About This Manual

This guide provides Japanese-specific information, such as character sets and locales, for end users and programmers so that they can use and develop internationalized applications in Japanese locales on the DIGITAL UNIX operating system. Details of the Japanese features are also documented in this guide.

Intended Audience

This guide is intended for new and experienced end users and programmers who are interested in the Japanese variant of the DIGITAL UNIX operating system.

Structure of This Guide

This guide consists of nine chapters:

Chapter 1 Character Sets
Describes the Japanese character sets supported in the DIGITAL UNIX operating system software

Chapter 2 Codesets and Codeset Conversion
Describes the Japanese codesets

Chapter 3 Locales
Describes the Japanese locales

Chapter 4 Local Language Devices
Describes the hardware devices that support the Japanese locales

Chapter 5 Fonts
Provides information on Japanese fonts

Chapter 6 Keyboards
Provides information on Japanese keyboards

Chapter 7 Japanese Input Methods
Describes how to input Japanese characters

Chapter 8 Japanese Printing Support
Introduces Japanese printing support

Chapter 9 Other Japanese Features
Describes other Japanese features

Related Documents

Writing Software for the International Market

Programming for the World: A Guide to Internationalization, Sandra Martin O'Donnell, Prentice Hall, 1994

OSF/Motif User's Guide Revision 1.2, Open Software Foundation, Prentice Hall, Englewood Cliffs, New Jersey 07632

OSF/Motif Style Guide Revision 1.2, Open Software Foundation, Prentice Hall, Englewood Cliffs, New Jersey 07632

X Window System, Third Edition, Robert W. Scheifler and James Gettys, Digital Press

Programmer's Supplement for Release 5 of the X Window System, Version 11, David Flanagan, O'Reilly & Associates, Inc.

The Unicode Standard, Version 2.0, Unicode Inc., Addison Wesley, 1996

Information Technology-Universal Multiple-Octet Coded Character Set, ISO/IEC 10646: 1993

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 shell.
#	A number sign represents the superuser prompt.

`%` **`cat`**

Boldface type in interactive examples indicates typed user input.

`file`

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.

`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.

[RETURN]

In an example, a key name enclosed in a box indicates that you press that key.

Ctrl/*x*

This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example [Ctrl/C]).

1

Character Sets

The DIGITAL UNIX operating system software supports the following Japanese Industrial Standard (JIS) standard character sets:

- JIS X 0201
- JIS X 0208
- JIS X 0212

It also supports the following international standard codesets:

- Unicode
- ISO/IEC 10646

1.1 JIS X 0201

The JIS X 0201 is a national standard for Japanese information interchange. It was first published by the Japanese Industrial Standard Committee in 1976. The current revision is JIS X 0208-1997. It is a single-byte character set and consists of Roman letters and Katakana characters.

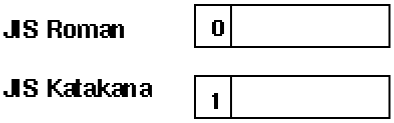
- Roman letters (00-7F)

  The Roman letters defined in JIS X 0201 are the same as those in ASCII, except that the tilde sign (~) is replaced by a horizontal bar (located at the upper part of a character cell), and the backslash (\) is replaced by the Japan currency sign (Yen).

- Katakana characters (A1-DF)

  The Katakana characters are Japanese phonetic symbols.

Figure 1-1: JIS X 0201 Encoding



1.2 JIS X 0208

The JIS X 0208 is a national standard for a primary set of graphic characters for Japanese information interchange. It was first published by the Japanese Industrial Standard Committee in 1978. The current revision is JIS X 0208-1997.

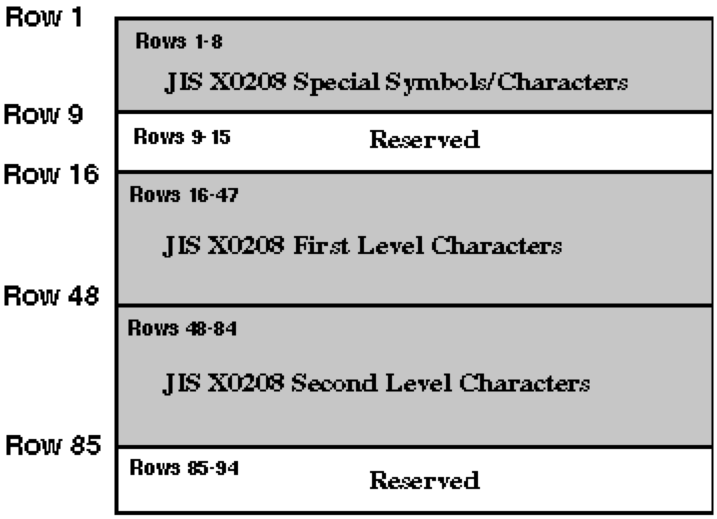
JIS X 0208 has defined a total of 6,879 characters which include those in Table 1-1.

Table 1-1: Characters Defined in JIS X 0208

Rows	Characters
1 and 2	Special symbols
3	Numerals and Roman alphabets
4	Hiragana characters
5	Katakana characters
6	Greek alphabets
7	Russian alphabets
8	Symbols for drawing graphs, diagrams, and lines
16-47	First level Kanji
48-84	Second level Kanji

The JIS X 0208 code table is divided into 94 rows, numbered from 1 to 94. Each row has 94 columns, also numbered from 1 to 94.

Figure 1-2: JIS X 0208 Character Set


Row 1
Rows 1-8
JIS X0208 Special Symbols/Characters
Row 9
Rows 9-15
Reserved
Row 16
Rows 16-47
JIS X0208 First Level Characters
Row 48
Rows 48-84
JIS X0208 Second Level Characters
Row 85
Rows 85-94
Reserved

1.3 JIS X 0212

The JIS X 0212 is a national standard for a supplemental set of graphic characters for Japanese information interchange. JIS X 0212 was published by the Japanese Industrial Standard Committee in 1990, and JIS X 0212-1990 is the current revision.

Support for JIS X 0212 characters in DIGITAL UNIX is limited. The DIGITAL UNIX operating system provides no fonts or input methods for JIS X 0212 characters. However, the character data encoded by the codesets which contain the JIS X 0212 character set can be processed properly by internationalized library functions.

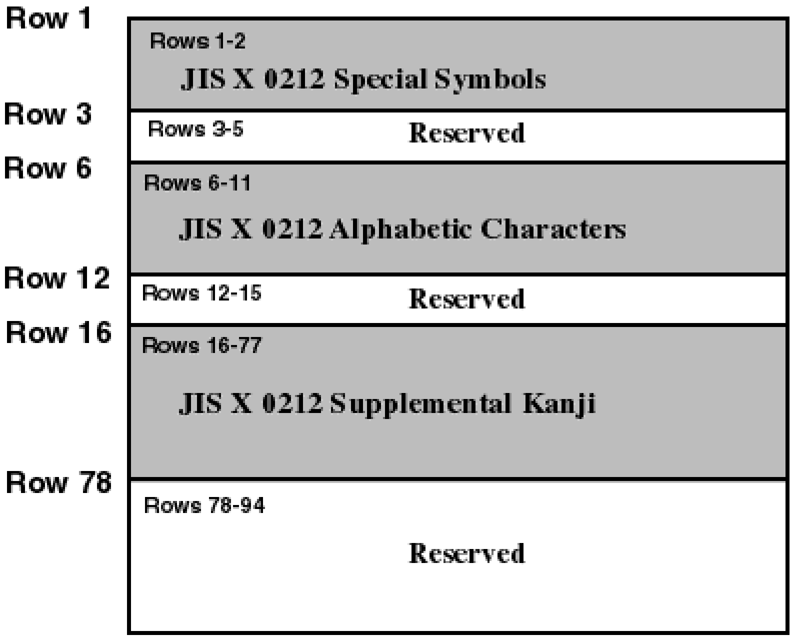
JIS X 0212 has defined a total of 6,067 characters which include those in Table 1-2.

Table 1-2: Characters Defined in JIS X 0212

Rows	Characters
1 and 2	Special symbols
6-11	Alphabetic characters
16-77	Supplemental Kanji

The JIS X 0212 code table is divided into 94 rows, numbered from 1 to 94. Each row has 94 columns, also numbered from 1 to 94.

Figure 1-3: JIS X 0212 Character Set



1.4 Unicode

The Unicode Standard, Version 2.0 specifies a universal character set (UCS) that contains definitions for 38,885 characters and also includes a Private Use Area for vendor- or user-defined characters. The main features of this character set are:

- All characters are treated as 16-bit units.
- Some nonspacing diacritic marks are defined, which can be combined with base characters in Unicode. For example, you can encode the character à as the 16-bit character a plus the 16-bit nonspacing grave accent. The result is a 32-bit composed sequence. The Unicode standard provides rules for ordering characters so that parsing of character sequences is unambiguous.
- Characters have properties, such as base, numeric, spacing, combination, and directionality.
- The relationship between Unicode characters and the glyphs in the native language script that users see, type, or print is not necessarily one-to-one. A glyph may be mapped to a single character or a composite sequence. Conversely, more than one glyph can be mapped to a character.

- The ISO 8859-1 character set occupies the first 256 code positions (and the ASCII character set the first 128 positions) of the UCS.

1.5 ISO/IEC 10646

The ISO/IEC 10646 standard, which is specified in *Information Technology-Universal Multiple-Octet Coded Character Set, ISO/IEC 10646*, allows characters to be specified as either 32-bit units or like Unicode, as 16-bit units. In their 32-bit form, the 16-bit character values in Unicode are zero-extended.

2

Codesets and Codeset Conversion

DIGITAL UNIX fully supports the following Japanese codesets by including locales and codeset conversion support:

- DEC Kanji
- Japanese EUC (Extended UNIX Code)
- Super DEC Kanji
- Shift JIS

It also provides codeset conversion support for the following codesets:

- JIS Kanji
- ISO-2022-JP
- Extended ISO-2022-JP
- UCS-4
- UTF-8

2.1 DEC Kanji

DEC Kanji is the codeset currently used by all DIGITAL Japanese products. Thus, software supporting this codeset can exchange data with existing Japanese products. This codeset is denoted as `deckanji` in the DIGITAL UNIX system.

DEC Kanji is formed by the following character sets:

- ASCII or JIS X 0201 Roman letters
- JIS X 0208
- User-Defined Characters (UDC)

DEC Kanji uses a combination of single-byte data and two-byte data to represent ASCII characters, symbols, and ideographic characters.

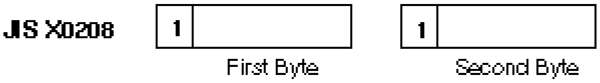
2.1.1 ASCII or JIS X 0201 Roman Letter Code

All ASCII characters or JIS X 0201 Roman letters can be represented in the form of single-byte 7-bit data in DEC Kanji. That is, the most significant bit (MSB) of these characters is always set off.

2.1.2 JIS X 0208 Code

Each JIS X 0208 character is represented by a two-byte code in DEC Kanji. The MSB of both bytes is always set on to distinguish it from an ASCII/JIS Roman character or a user-defined character.

Figure 2-1: Representation of a JIS X 0208 Character in DEC Kanji



The first byte of a two-byte code determines its row number, while the second determines its column number. The following formula illustrates the code of a JIS X 0208 character in relation to its row and column numbers:

1st byte = A0 + row number

2nd byte = A0 + column number

For example, if a character is positioned at the first column of the 36th row, its encoding value can be calculated as follows:

1st byte = A0 (hex) + 36 = C4 (hex)

2nd byte = A0 (hex) + 01 = A1 (hex)

In this case, the character code is C4A1.

2.1.3 User–Defined Character Code

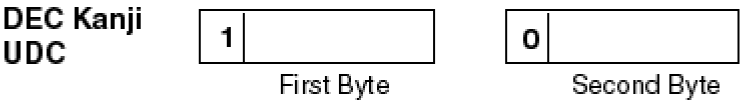
In addition to the ASCII or JIS Roman Code and the JIS X 0208 Code, DEC Kanji provides an area of 2,914 positions for user-defined characters. This UDC code range is shown in Table 2-1.

Table 2-1: DEC Kanji UDC Code Range

Area Usage	Row Range	Number of Characters	Code Range
User Area	1-31	2,914	A121-BF7E
DEC Reserved	32-94		

A UDC is also represented by a two-byte code, just like a JIS X 0208 character. However, the MSB of the second byte is set off to distinguish it from a JIS X 0208 character, as shown in Figure 2-2.

Figure 2-2: Representation of a UDC in DEC Kanji



The following formula illustrates the code of a UDC in relation to its row and column numbers:

1st byte = A0 + row number

2nd byte = 20 + column number

For example, if a UDC is positioned at the first column of the 16th row, its encoding value can be calculated as follows:

1st byte = A0 (hex) + 16 = B0 (hex)

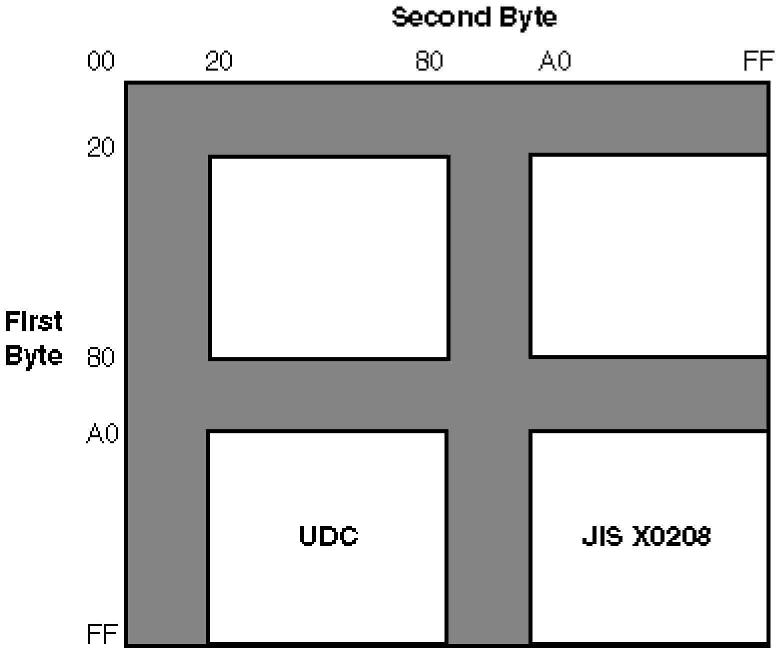
2nd byte = 20 (hex) + 01 = 21 (hex)

In this case, the character code is B021.

2.1.4 Two–Byte Code Space

Figure 2-3 illustrates the division of the two-byte code space and the position of JIS X 0208 and User-Defined Characters in DEC Kanji:

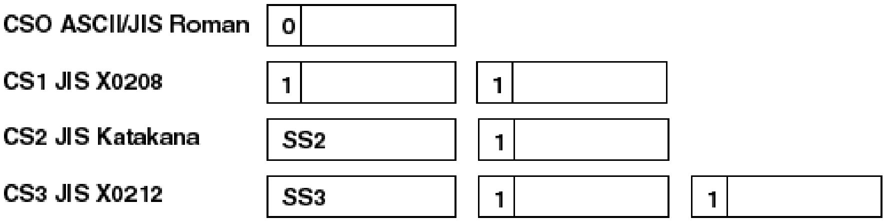
Figure 2-3: Two-Byte Code Space for DEC Kanji



2.2 Japanese EUC

Extended UNIX Code (EUC) is an encoding method that allows up to four character sets to be combined in a single data stream. Japanese EUC, denoted as `eucJP`, is the EUC codeset for representing Japanese data.

Figure 2-4: Encoding of Japanese EUC



CSO is called the primary character set while CS1 through CS3 are the supplementary character sets. The MSB of the primary character set must be off while the MSB of all bytes in the supplementary character sets must be on. This scheme is used to determine the character set to which a character belongs.

The representation of ASCII/JIS Roman and JIS X 0208 characters is similar to that of DEC Kanji. In addition, two more character sets, JIS Katakana and JIS X 0212, are encoded in Japanese EUC by making use of the Single-Shift 2 (SS2) and Single-Shift 3 (SS3) control characters.

Japanese EUC provides two areas for defining a UDC, as shown in Table 2-2.

Table 2-2: Japanese EUC UDC Code Range

Area Usage	Row Range	Number of Characters	Code Range
JIS X 0208	85-94	940	F5A1-FEFE
JIS X 0212	78-94	1,598	SS3 + EEA1-FEFE

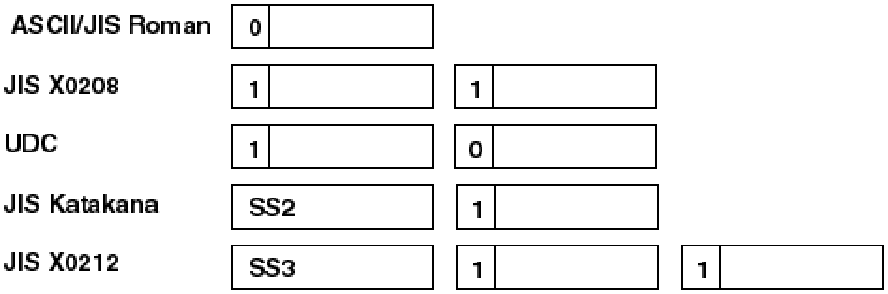
Note

JIS X 0212 characters (JIS Supplementary Kanji) are not supported in this release of the DIGITAL UNIX operating system.

2.3 Super DEC Kanji

Super DEC Kanji, denoted as `sdeckanji`, is an extension to DEC Kanji which supports the CS2 (JIS Katakana) and CS3 (JIS X 0212) character sets as encoded in Japanese EUC. It is a superset of both DEC Kanji and Japanese EUC. Data encoded in both DEC Kanji and Japanese EUC can be handled with this unified codeset. This codeset was invented to ease the transition from DEC Kanji to Japanese EUC. Figure 2-5 illustrates the encoding of Super DEC Kanji.

Figure 2-5: Encoding of Super DEC Kanji



Super DEC Kanji provides three areas for defining UDCs, as shown in Table 2-3.

Table 2-3: Super DEC Kanji UDC Code Range

Area Usage	Row Range	Number of Characters	Code Range
JIS X 0208	85-94	940	F5A1-FEFE
JIS X 0212	78-94	1,598	SS3 + EEA1-FEFE
UDC	1-94	8,836	A121-FE7E

2.4 Shift JIS

Shift JIS, denoted as `SJIS`, is a popular codeset which is widely used in the PC market.

Shift JIS codes use a combination of single-byte data and two-byte data to represent characters defined in JIS X 0201 and JIS X 0208. To allow the characters defined in these standards to be encoded in a single codeset, the first byte of each JIS X 0208 character is encoded in the ranges 81-9F and EO-FC, while the second byte is between 40 and FC, as shown in Table 2-4.

Table 2-4: Code Range of JIS X 0208 Characters in Shift JIS

Byte	Range
First byte	81-9F, E0-FC
Second byte	40-FC (except 7F)

Figure 2-6 illustrates the first and second byte code space of Shift JIS.

Figure 2-6: Code Space of Shift JIS

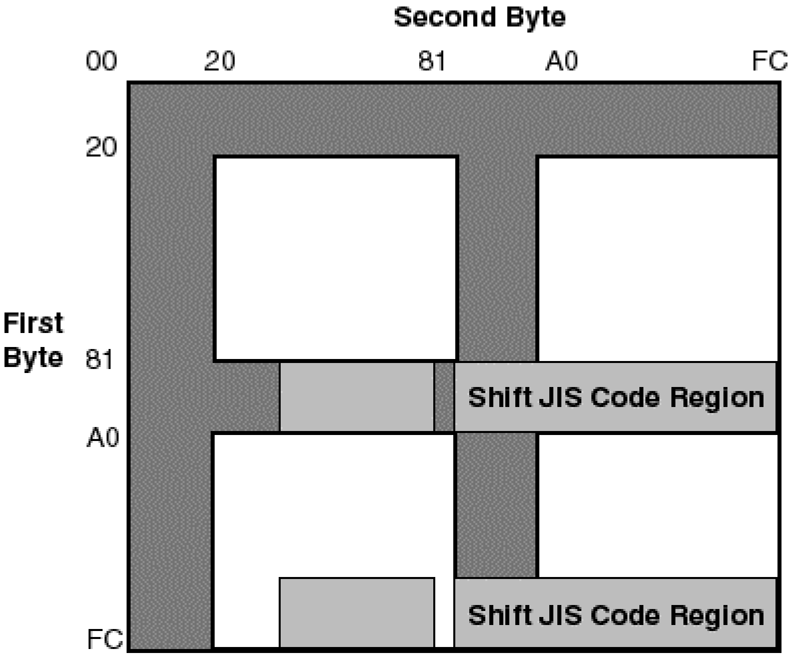


Table 2-5 illustrates the mapping from the encoding of the first byte to the corresponding character sets in the Shift JIS encoding.

Table 2-5: Character Set Mapping in Shift JIS

Code Range of First Byte	Character Set	Bytes per Character
00-7F	JIS Roman (X 0201)	1
81-9F	JIS X0208	2
A1-DF	JIS Katakana (X 0201)	1
E0-FC	JIS X 0208	2

Shift JIS provides an area for defining UDCs as follows:

Number of characters:	2,444
Code range:	F040 - FCFC

2.5 JIS Kanji

The JIS Kanji codesets use the ISO 2022 methodology for encoding the JIS X 0208 and JIS X 0201 character sets. There are two types of JIS Kanji encoding: 7-bit JIS Kanji code and 8-bit JIS Kanji code.

2.5.1 7-Bit JIS Kanji Code

In 7-bit JIS Kanji encoding, all characters are represented as 7 bits. Characters are interpreted according to control sequences as follows:

Kanji in sequence (ESC $ B)

> Code values following the Kanji-in sequence (ESC $ B) are treated as characters in the JIS X 0208 Kanji character set.

Kanji out sequence (ESC (B)

> Code values following the Kanji-out sequence (ESC (B) are treated as ASCII characters.

Supplementary Kanji in sequence (ESC $ (D)

> Code values following the supplementary Kanji in sequence (ESC $ (D) are treated as characters in the JIS X 0212 supplementary Kanji character set.

User-Defined Character (UDC) in sequence (ESC $ (0)

> Code values following the UDC in sequence (ESC $ (0) are treated as characters in the vendor-defined or user-defined character set.

Kana in (SO) and Kana out (SI) sequences

> Code values following the Shift-Out (SO) control character (0x0e) and preceding the Shift-In (SI) control character (0x0f) are treated as characters in the JIS X 0201 Katakana character set.

Katakana in sequence (ESC (I)

> Code values following the Katakana in sequence (ESC (I) are treated as characters in the JIS X 0201 Katakana character set. In this case, the Kanji out sequence is used to switch back to ASCII code.

The Katakana in and Kanji out sequences are an alternative to using the Kana in and out sequences (SO/SI).

2.5.2 8-Bit JIS Kanji Code

In 8-bit JIS Kanji encoding the JIS X 0201 Katakana characters are represented as 8-bits. Using this form of encoding, control sequences have the following effect:

Kanji in sequence (ESC $ B)

Code values following the Kanji in sequence (ESC $ B) are treated as characters in the JIS X 0208 Kanji character set.

Supplementary Kanji in sequence (ESC $ (D)

Code values following the supplementary Kanji in sequence (ESC $ (D) are treated as characters in the JIS X 0212 supplementary Kanji character set.

User-Defined Character (UDC) in sequence (ESC $ (0)

Code values following the UDC in sequence (ESC $ (0) are treated as vendor-defined or user-defined characters.

Kanji out sequence (ESC (B)

Code values following the Kanji out sequence (ESC (B) are treated as ASCII characters.

Kana in and out sequences (SI/SO)

These sequences are ignored.

2.5.3 Restrictions

The JIS Kanji codesets can be used in codeset conversion and terminal display.

For codeset conversion using the `iconv` utility, the string `JIS7` indicates 7-bit JIS Kanji code that follows a Katakana in sequence and the string `jiskanji7` indicates 7-bit JIS Kanji code entered between Kana in and out sequences. The following sequences to the `iconv` utility are valid within the input data that `iconv` converts, but `iconv` does not generate these sequences when converting to JIS Kanji:

- Kanji in (ESC $ @)
- Kanji in (ESC & @ ESC $ B)
- Kanji in (ESC $ (B)
- Kanji in (ESC $ (@)
- Supplementary Kanji in (ESC $ D)
- Kana in (ESC (J)
- Kana in (ESC (H)

For terminal display using `tty`, the string `jis7` indicates 7-bit JIS Kanji code and the string `jis8` indicates 8-bit JIS Kanji code. When the terminal code is set to `jis7`, the Kana in and out sequences (SI/SO) are used for JIS X 0201 Katakana character representation.

2.6 ISO-2022-JP

The ISO-2022-JP codeset consists of the following character sets:

- ASCII
- JIS X 0201-1976
- JIS X 0208-1978
- JIS X 0208-1990

Note

JIS X 0208-1990 is a revised version of JIS X 0208-1978. Some characters of JIS X 0208-1978 were mapped to other positions.

Before a character set is used, it must be identified using an escape sequence as follows:

Escape Sequence	Character Set
ESC (B	ASCII
ESC (J	JIS X 0201-1976 (left-hand part)
ESC $ @	JIS X 0208-1978
ESC $ B	JIS X 0208-1990

It is assumed that the starting code of a line is ASCII (including CR alone and LF alone, but not including the combination CRLF). If there are JIS X 0208 characters on a line, there must be a switch to ASCII or to the left-hand part of JIS X 0201 (Roman letters) before the end of the line (in other words, before the CRLF, or carriage return and line feed).

For example, if a line starts with the ASCII character 9, followed by the JIS X 0208-1978 character at row 16 column 1, the line is encoded as follows:

```
39h ESC $ @ 30h 21h .... ESC ( B .... CRLF
```

If a line starts with the JIS X 0208-1978 character at row 16 column 1, followed by the ASCII character 9, then the line is encoded as follows:

```
ESC $ @ 30h 21 ESC ( B 39h .... CRLF
```

Once a character set is designated, there is no need to redesignate the character set if the adjacent character belongs to the same character set. For example, the following practice is not recommended:

```
ESC $ B .... ESC $ B ....
```

Currently, the ISO-2022-JP codeset can be used in codeset conversion.

The `iconv` utility uses the following escape sequences when code is converted to ISO-2022-JP:

Escape Sequence	Character Set
ESC (B	ASCII
ESC $ B	JIS X 0208

2.7 Extended ISO-2022-JP

The extended ISO-2022-JP codeset, denoted as `ISO-2022-JPext`, is an extended version of the ISO-2022-JP codeset. It is extended to support narrow JIS X 0201 Katakana characters, JIS X 0212 characters, and user-defined characters (UDC).

This codeset can be used in codeset conversion.

The `iconv` utility uses the following escape sequences when code is converted to ISO-2022-JPext:

Escape Sequence	Character Set
ESC (B	ASCII
ESC $ B	JIS X 0208
ESC (I	JIS X 0201 Katakana
ESC $ (D	JIS X 0212
ESC $ (0	UDC

2.8 UCS-2/UCS-4

UCS is a standard character encoding for the universal character set specified in the Unicode and ISO/IEC 10646 standards. UCS has two forms; UCS-2 (16-bit, or 2 octet units) and UCS-4 (32-bit, or 4 octet units). Unicode uses the UCS-2 form, which is commonly used on personal computers. ISO/IEC allows either UCS-2 or UCS-4 encoding. UCS-4 encoding is in use on systems that can support the larger data unit size.

The current version of the DIGITAL UNIX operating system supports both UCS-2 and UCS-4 encoding. UCS-4 is available in some Japanese locales, and can be used in codeset conversion. For information about codeset conversion, see Section 2.10. For information about locales, see Chapter 3, Locales.

2.9 UTF-8

Unicode and ISO/IEC 10646 standards define transformation formats for the universal character set. For the most part, the following UCS transformation formats (UTFs) exist to transform UCS values into sequences of bytes for handling by various byte-oriented protocols:

- UTF-8, the standard method for transforming UCS-4 or UCS-2 data into a sequence of 8-bit bytes and ensuring interchange transparency for characters from the ASCII character set (code positions 0 through 127).
- UTF-7, the standard interchange format for environments that strip the eighth bit from each byte.
- UTF-16, a transformation format that allows systems that can process only 16-bit units (specified by UCS-2 encoding) to support the extended character definition space that is included in UCS-4.

The current version of the DIGITAL UNIX operating system supports UTF-8 and UTF-16 but not UTF-7. UTF-8 can be used in codeset conversion and in the *universal.utf8* locale. For information about codeset conversion, see Section 2.10. For information about locale variants, see Chapter 3, Locales.

2.10 Codeset Conversion

The `iconv` utility provided by DIGITAL UNIX converts the encoding of characters in one codeset to another and writes the results to standard output. The following pairs of Japanese codeset converters are provided:

	DEC Kanji	Japanese EUC	Super DEC Kanji	Shift JIS	JIS 7	ISO-2022-JP	ISO-2022-JPext	UCS-4	UTF-8
DEC Kanji	—	Y	Y	Y	Y	Y	Y	Y	Y
Japanese EUC	Y	—	Y	Y	Y	Y	Y	Y	Y
Super DEC Kanji	Y	Y	—	Y	Y	Y	Y	Y	Y
Shift JIS	Y	Y	Y	—	Y	Y	Y	Y	Y
JIS7	Y	Y	Y	Y	—	N	N	N	N
ISO-2022-JP	Y	Y	Y	Y	N	—	N	N	N
ISO-2022-JPext	Y	Y	Y	Y	N	N	—	N	N
UCS-4	Y	Y	Y	Y	N	N	N	—	Y
UTF-8	Y	Y	Y	Y	N	N	N	Y	—

For example, you can enter the following command to convert a DEC Kanji file to a Shift JIS file:

```
% iconv -f deckanji -t SJIS <file>
```

Use the strings shown in Table 2-6 as the parameters to the `iconv` utility.

Table 2-6: Codeset Names

Codeset	String
DEC Kanji	deckanji
Japanese EUC	eucJP
Super DEC Kanji	sdeckanji
Shift JIS	SJIS
JIS7 (ESC (I for katakana)	JIS7
JIS7 (SO/SI for katakana)	jiskanji7
ISO-2022-JP	ISO-2022-JP
Extended ISO-2022-JP	ISO-2022-Jpext
UCS-2	UCS-2
UCS-4	UCS-4
UTF-8	UTF-8

2.10.1 User–Defined Character Mappings

There are four supported Japanese codesets that are used in the Japanese locales. They are DEC Kanji, super DEC Kanji, Japanese EUC, and SJIS. Each one has its own UDC ranges. There is a predefined mapping for UDC among these four codesets, as shown in the following table:

SJIS	Deckanji	sdeckanji	eucJP
0xf040-0xf4fc	0xa121-0xaa7e	0xa121-0xaa7e	0xf5a1-0xfefe
0xf540-0xf9fc	0xab21-0xb47e	0xab21-0xb47e	0x8ff5a1-0x8ffefe
0xfa40-0xfcfc	0xb521-0xbb7e	0xb521-0xbb7e	0x8feea1-0x8ff3fe

If you try to modify the codeset of a UDC, the UDC manager will ask if you want the other codeset values to be changed accordingly. Always choose the default answer to avoid problems with other software. For instance, if you define a SJIS UDC value of 0xf040, it will be mapped to the deckanji and sdeckanji value of 0xa121 and the eucJP value of 0xf5a1 automatically.

You should not use UDC outside the ranges defined in the above table; if you do, the automatic mapping will not work properly.

2.11 Codeset for Peripheral Devices

The DIGITAL UNIX operating system provides a mechanism by which you configure your system to run applications with peripherals, such as terminals and printers, supporting different codesets. You can specify the codesets for the applications, terminals, and printers independently as shown in Table 2-7. The DIGITAL UNIX software automatically converts data to the appropriate codeset.

The DEC terminal codeset is similar to DEC Kanji, but has support for Kana characters (in eucJP) as well. It has support for JISX0208 and JISX0208-1978, but not JISX0212 of eucJP. The dec78 codeset supports an older version of JISX0208-1978 which has characters that are slightly different from JISX0208-1983 supported in dec and deckanji.

Table 2-7: Feasible Codeset for Applications, Terminals, and Printers

Application Code	Terminal Code	Printer Code
DEC Kanji	DEC (dec) DEC78 (dec78)	DEC Kanji
Japanese EUC	Japanese EUC	Japanese EUC
Super DEC Kanji		Super DEC Kanji
Shift JIS	Shift JIS (SJIS) JIS7 (jis7) JIS8 (jis8)	Shift JIS (SJIS)

Note

Japanese DECterm software supports the deckanji, sdeckanji, or eucJP codeset (except for the user-defined characters) as its terminal code.

For the details about setting up terminal code and printer code, please see *Writing Software for the International Market* or *Nihongo Kinou Guide Book* (written in Japanese).

3

Locales

The DIGITAL UNIX operating system supports the Japanese locales shown in Table 3-1.

Table 3-1: Japanese Locales

Codeset	Locale
DEC Kanji	ja_JP ja_JP.deckanji ja_JP.deckanji@ucs4
Japanese EUC	ja_JP.eucJP
Super DEC Kanji	ja_JP.sdeckanji
Shift JIS	ja_JP.SJIS ja_JP.SJIS@ucs4

The `ucs4` modifier indicates that UCS-4 is used as the internal processing code. The classification information, however, is not provided for the full set of UCS-4 characters, but only for those specified in the other Japanese locales.

If you are using DECwindows Motif, you can select the locale through the Language Menu of Session Manager. If you are using CDE, you can select the locale using the language menu on CDE login screen. The applicable locales are shown in Table 3-2.

Table 3-2: Japanese Language Names

Locale	Language Name
ja_JP	Japanese
ja_JP.deckanji	Japanese (DEC Kanji)
ja_JP.eucJP	Japanese (EUC)
ja_JP.sdeckanji	Japanese (Super DEC Kanji)
ja_JP.SJIS	Japanese (Shift JIS)

4

Local Language Devices

4.1 Terminals

The DIGITAL UNIX operating system software supports the following Japanese terminal types:

- VT282-J
- VT382-J
- VT383-J
- VT284-J
- VT286-J

Japanese DECterm is the emulation of the VT282-J and VT382-J Japanese terminals which provide compatible functionalities for running Japanese character-cell terminal applications. For details about Japanese DECterm, see Chapter 9, Other Japanese Features.

4.2 Printers

The DIGITAL UNIX operating system software supports the following Japanese printers:

- LA84-J
- LA86-J
- LA88-J
- LA90-J
- LA280-J

- LA380-J
- LN03S-JA
- LN82R
- DEClaser 2300
- DEClaser 2400

The print filters shown in Table 4-1 are provided to support Japanese printers.

Table 4-1: Japanese Print Filters

Filter Name	Printer Name
la84of	LA84-J printer
la86of	LA86-J printer
la90of	LA90-J printer
la280of	LA280-J printer
la380of	LA380-J printer
ln03jaof	LN03S-JA printer
ln05jaof	DEClaser 2300 printer
ln10jaof	DEClaser 2400 printer
ln82rof	LN82R printer
la88of	LA88-J printer
ln82rof_sdeckanji	LN82R printer
ln82rof_SJIS	LN82R printer

For the details about setting up Japanese printer queues, see *Writing Software for the International Market* or *Nihongo Kinou Guide Book* (written in Japanese).

5

Fonts

5.1 DECwindows Fonts

The DIGITAL UNIX operating system software provides the Japanese DECwindows fonts described in Table 5-1 in various sizes and typefaces for 75 dpi (dot-per-inch) and 100 dpi display devices.

Table 5-1: Japanese Screen Fonts

Typefaces	Character Set	Resolution	Size
Kmenu	ISO Latin-1 JIS X 0201	75, 100	12
Screen	DEC Suppl DEC Tech ISO Latin-1 JIS X 0201 JIS X 0208 GL JIS X 0208 GR	75	10, 14, 18, 24
Mincho	ISO Latin-1 JIS X 0201 JIS X 0208 GR	75, 100	8, 10, 12, 14, 18, 24
Gothic	ISO Latin-1 JIS X 0201 JIS X 0208 GR	75, 100	8, 10, 12, 14, 18, 24

No 100 dpi Screen family fonts are provided in the kit. To allow you to use the Screen family fonts on 100 dpi display devices, a font alias file is provided to map the 75 dpi font names to the respective 100 dpi font names.

5.1.1 XLFD Font Names

You must specify the DECwindows font names in XLFD (X Logical Font Description) format in your application programs or in the application resource files. You can specify wildcards (*) for any fields in the font names.

You can use the following font names for either 75 dpi or 100 dpi display devices. To explicitly state the display resolution, you can specify 75 or 100 in the X- and Y-resolution fields, that is, the second and third asterisks in the following XLFD names:

- Kmenu family:

  ISO Latin-1 Font

  ```
  -JDECW-Kmenu-Medium-R-Normal--*-120-*-*-P-70-iso8859-1
  ```

  JIS X 0201 Font

  ```
  -JDECW-Kmenu-Medium-R-Normal--*-120-*-*-P-70-jisx0201.1976-0
  ```

- Screen family:

  JIS X0201 Fonts

  ```
  -JDECW-Screen-Medium-R-Normal--*-100-*-*-M-50-jisx0201-RomanKana
  -JDECW-Screen-Medium-R-Normal--*-140-*-*-M-70-jisx0201-RomanKana
  -JDECW-Screen-Medium-R-Normal--*-180-*-*-M-80-jisx0201-RomanKana
  -JDECW-Screen-Medium-R-Normal--*-240-*-*-M-120-jisx0201-RomanKana
  ```

  JIS X 0208 GL (Kanji00) Fonts

  ```
  -JDECW-Screen-Medium-R-Normal--*-100-*-*-M-100-jisx0208-Kanji00
  -JDECW-Screen-Medium-R-Normal--*-140-*-*-M-140-jisx0208-Kanji00
  -JDECW-Screen-Medium-R-Normal--*-180-*-*-M-160-jisx0208-Kanji00
  -JDECW-Screen-Medium-R-Normal--*-240-*-*-M-240-jisx0208-Kanji00
  ```

  JIS X 2080 GR (Kanji11) Fonts

  ```
  -JDECW-Screen-Medium-R-Normal--*-100-*-*-M-100-jisx0208-Kanji11
  -JDECW-Screen-Medium-R-Normal--*-140-*-*-M-140-jisx0208-Kanji11
  -JDECW-Screen-Medium-R-Normal--*-180-*-*-M-160-jisx0208-Kanji11
  -JDECW-Screen-Medium-R-Normal--*-240-*-*-M-240-jisx0208-Kanji11
  ```

- Mincho family:

  ISO Latin-1 Fonts

  ```
  -JDECW-Mincho-Medium-R-Normal--*80-*-*-M-40-iso8859-1
  -JDECW-Mincho-Medium-R-Normal--*100-*-*-M-50- iso8850-1
  -JDECW-Mincho-Medium-R-Normal--*120-*-*-M-60- iso8859-1
  -JDECW-Mincho-Medium-R-Normal--*140-*-*-M-70- iso8859-1
  ```

-JDECW-Mincho-Medium-R-Normal--*180-*-*-M-90- iso8859-1
-JDECW-Mincho-Medium-R-Normal--*240-*-*-M-120-iso8859-1

JIS X 0201 Fonts

-JDECW-Mincho-Medium-R-Normal--*80-*-*-M-40-jisx0201.1976-0
-JDECW-Mincho-Medium-R-Normal--*100-*-*-M-50-jisx0201.1976-0
-JDECW-Mincho-Medium-R-Normal--*120-*-*-M-60-jisx0201.1976-0
-JDECW-Mincho-Medium-R-Normal--*140-*-*-M-70-jisx0201.1976-0
-JDECW-Mincho-Medium-R-Normal--*180-*-*-M-90-jisx0201.1976-0
-JDECW-Mincho-Medium-R-Normal--*240-*-*-M-120-jisx0201.1976-0

JIS X 0208 GR Fonts

-JDECW-Mincho-Medium-R-Normal--*80-*-*-M-80-jisx0208.1983-1
-JDECW-Mincho-Medium-R-Normal--*100-*-*-M-100-jisx0208.1983-1
-JDECW-Mincho-Medium-R-Normal--*120-*-*-M-120-jisx0208.1983-1
-JDECW-Mincho-Medium-R-Normal--*140-*-*-M-140-jisx0208.1983-1
-JDECW-Mincho-Medium-R-Normal--*180-*-*-M-180-jisx0208.1983-1
-JDECW-Mincho-Medium-R-Normal--*240-*-*-M-240-jisx0208.1983-1

- Gothic family:

  ISO Latin-1 Fonts

  -JDECW-Gothic-Medium-R-Normal--*-80-*-*-M-40-iso8859-1
  -JDECW-Gothic-Medium-R-Normal--*-100-*-*-M-50- iso8859-1
  -JDECW-Gothic-Medium-R-Normal--*-120-*-*-M-60- iso8859-1
  -JDECW-Gothic-Medium-R-Normal--*-140-*-*-M-70- iso8859-1
  -JDECW-Gothic-Medium-R-Normal--*-180-*-*-M-90- iso8859-1
  -JDECW-Gothic-Medium-R-Normal--*-240-*-*-M-120- iso8859-1

  JIS X 0201 Fonts

  -JDECW-Gothic-Medium-R-Normal--*-80-*-*-M-40-jisx0201.1976-0
  -JDECW-Gothic-Medium-R-Normal--*-100-*-*-M-50-jisx0201.1976-0
  -JDECW-Gothic-Medium-R-Normal--*-120-*-*-M-60-jisx0201.1976-0
  -JDECW-Gothic-Medium-R-Normal--*-140-*-*-M-70-jisx0201.1976-0
  -JDECW-Gothic-Medium-R-Normal--*-180-*-*-M-90-jisx0201.1976-0
  -JDECW-Gothic-Medium-R-Normal--*-240-*-*-M-120-jisx0201.1976-0

  JIS X 0208 GR Fonts

  -JDECW-Gothic-Medium-R-Normal--*-80-*-*-M-80-jisx0208.1983-1
  -JDECW-Gothic-Medium-R-Normal--*-100-*-*-M-100-jisx0208.1983-1
  -JDECW-Gothic-Medium-R-Normal--*-120-*-*-M-120-jisx0208.1983-1
  -JDECW-Gothic-Medium-R-Normal--*-140-*-*-M-140-jisx0208.1983-1

-JDECW-Gothic-Medium-R-Normal--*-180-*-*-M-180-jisx0208.1983-1
-JDECW-Gothic-Medium-R-Normal--*-240-*-*-M-240-jisx0208.1983-1

- XLFD of Miscellaneous Japanese Screen Fonts:

XLFD Font Name	Character Set
-JDECW-Screen-Medium-R-Normal--*-140-*-*-M-70-iso8859-1	ISO Latin-1
-JDECW-Screen-Medium-R-Normal--*-140-*-*-M-70-DEC-DECsuppl	DEC Supplemental
-JDECW-Screen-Medium-R-Normal--*-140-*-*-M-70-DEC-DECtech	DEC Technical
-JDECW-Screen-Medium-R-Normal--*-140-*-*-M-70-DEC-DRCS	DEC DRCS
-JDECW-Screen-Medium-R-Normal--*-180-*-*-M-80-iso8859-1	ISO Latin-1
-JDECW-Screen-Medium-R-Normal--*-180-*-*-M-80-DEC-DECsuppl	DEC Supplemental
-JDECW-Screen-Medium-R-Normal--*-180-*-*-M-80-DEC-DECtech	DEC Technical
-JDECW-Screen-Medium-R-Normal--*-180-*-*-M-80-DEC-DRCS	DEC DRCS
-JDECW-Screen-Medium-R-Normal--*-240-*-*-M-120-iso8859-1	ISO Latin-1
-JDECW-Screen-Medium-R-Normal--*-240-*-*-M-120-DEC-DECsuppl	DEC Supplemental
-JDECW-Screen-Medium-R-Normal--*-240-*-*-M-120-DEC-DECtech	DEC Technical
-JDECW-Screen-Medium-R-Normal--*-240-*-*-M-120-DEC-DRCS	DEC DRCS

5.1.2 Bitmap Font Samples

Figure 5-1 to Figure 5-4 show samples of the various families of Japanese fonts.

Figure 5-1: Kmenu Font Samples

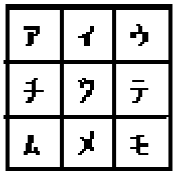


Figure 5-2: Screen Font Samples

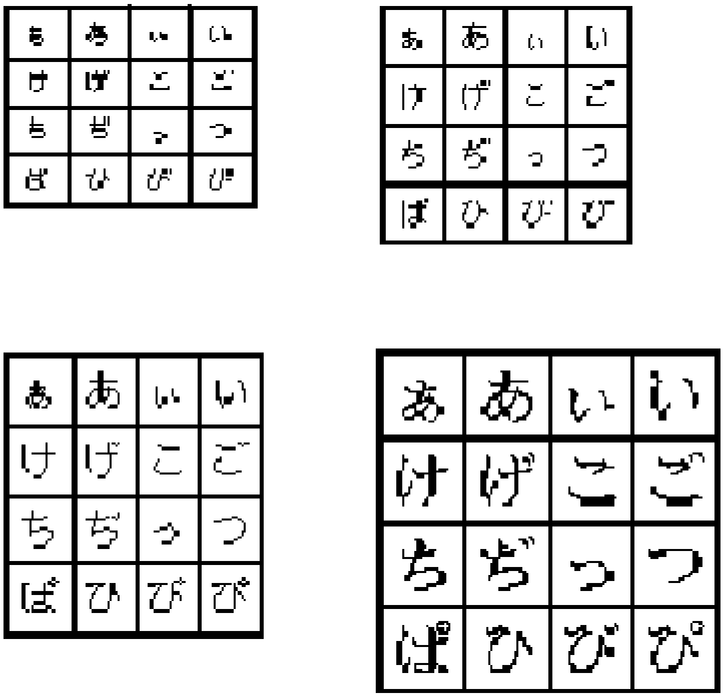


Figure 5-3: Mincho Font Sample

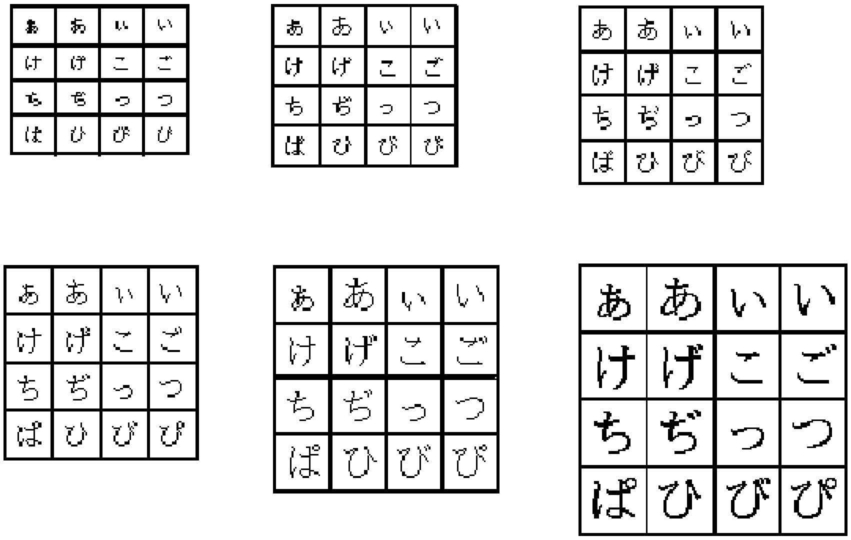
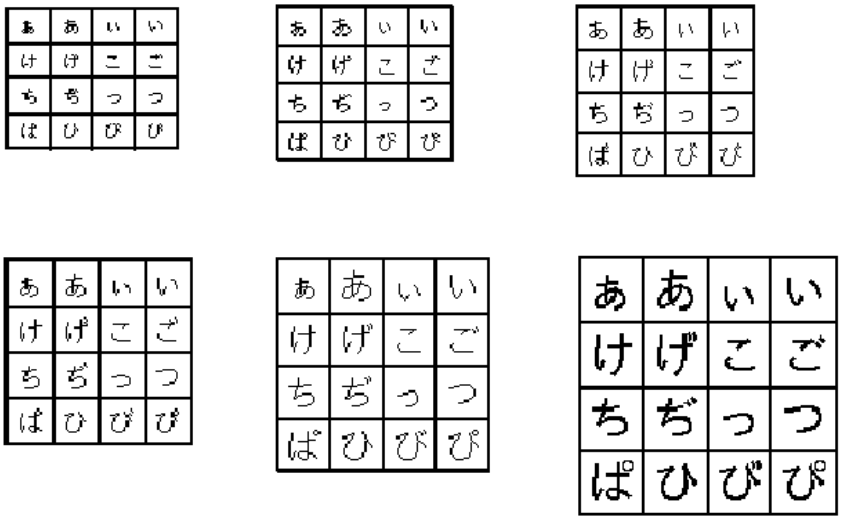


Figure 5-4: Gothic Font Sample



5.1.3 DECwindows Font Encoding

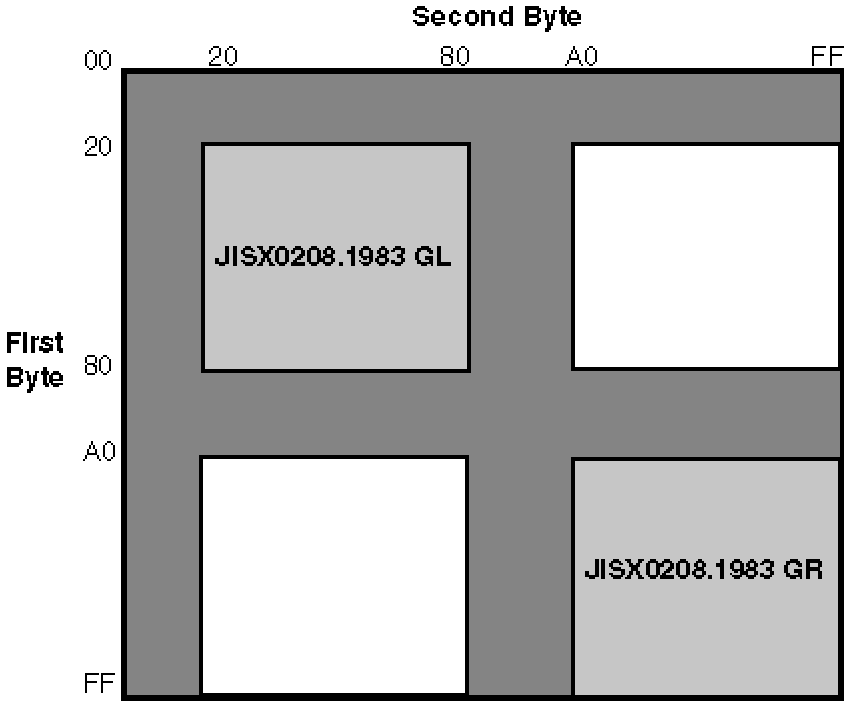
The X Consortium's Window System defines two encodings for the JIS X 0208-1983 character set, as shown in Table 5-2.

Table 5-2: Japanese DECwindows Font Encodings

Encoding	Character Set Registry
GL	jisx0208.1983-0
GR	jisx0208.1983-1

Figure 5-5 shows the difference between these two encoding schemes.

Figure 5-5: Japanese Font Encoding Schemes



Different vendors may adopt different encoding schemes to produce their fonts. With the exception of the Kanji00 Screen family fonts, the Japanese fonts supplied by Compaq Computer Corporation are all in GR encoding.

To allow you to run applications on third-party workstations on which only GL encoded fonts are installed, the DIGITAL UNIX implementation of the X Window System, Release 6 (X11R6), supports the conversion of GR encoding to GL encoding for text drawing and measurement, as shown in Table 5-3.

Table 5-3: Font Encoding Conversion

From	To
Jisx0208.1983-1	jisx0208.1983-0

For details, see *Writing Software for the International Market*.

For historical reasons, the Kanji Screen fonts are available in the kanji00 and kanji11 encodings. These encodings are equivalent to the X Consortium's font encodings, as shown in Table 5-4.

Table 5-4: Equivalent Japanese Font Encodings

Screen Font Encoding	X Consortium's Font Encoding
Kanji00	jisx0208.1983-0
Kanji11	Jisx0208.1983-1

5.1.4 Specifying Fonts in DECwindows Applications

The default font list used in the Motif Toolkit is shown in Table 5-5.

Table 5-5: Japanese Default Font List

XLFD Font Name	Character Set
-JDECW-Kmenu-Medium-R-Normal--*-120-*-*-P-*-iso8859-1	ISO8859-1
-JDECW-Gothic-Medium-R-Normal--*-120-*-*-M-*-jisx0201.1976-0	JISX0201.1976-0
-JDECW-Gothic-Medium-R-Normal--*-120-*-*-M-*-jisx0208.1983-1	JISX0208.1983-1
-JDECW-Screen-Medium-R-Normal--*-100-*-*-M-*-jisx.UDC-1	JISX.UDC-1
-JDECW-Gothic-Medium-R-Normal--*-120-*-*-M-*	Fontset

To override the default font list of a Japanese DECwindows application, you should specify the ISO Latin-1, JIS X 0201, JIS X 0208 and JIS (UDC) fonts as well as the Japanese fontset when creating widget instances. For details, please see *Writing Software for the International Market*.

5.2 Printer Fonts

The Japanese printer fonts for DIGITAL UNIX are resident in the printers, and not bundled with the DIGITAL UNIX operating system software. Please check the printer documentation for information about printer fonts.

6

Keyboards

The DIGITAL UNIX operating system software supports the Japanese keyboard types shown in Table 6-1.

Table 6-1: Japanese Keyboard Types

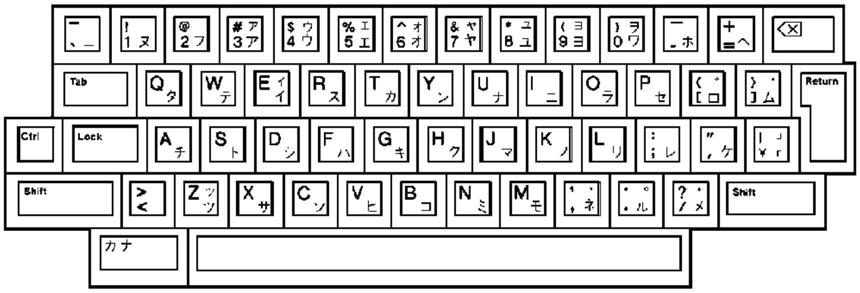
Keyboard	Layout	ConvKey	KanaLED	Description
LK201-AJ	ANSI	X	0	Japanese LK201 keyboard
LK401-AJ /LK40W-AJ	ANSI	X	X	Japanese KL401 keyboard
LK401-BJ	ANSI	0	0	Japanese LK401 keyboard
LK401-JJ	JIS	0	0	Japanese LK411 keyboard
LK411-AJ /LK41W-AJ	ANSI	0	0	Japanese LK411 keyboard
LK411-JJ /LK41W-JJ	JIS	0	0	Japanese LK421 keyboard
LK421-AJ	JIS	X	X	Japanese LK421 keyboard
LK421-JJ	JIS	0	0	Japanese LK431 keyboard
PCXA-AA	JIS	0	X	Japanese PC 106 keyboard
LK97W-AJ	JIS	0	X	Japanese PC 109 keyboard

Several Japanese keyboards have special keys for Japanese input methods. (See the ConvKey column in Table 6-1). LK401W-AJ, LK41W-AJ, and LK41W-JJ are equivalent to KJ401-AJ, LK411-AJ, and LK41W-JJ respectively, except color.

6.1 Keyboard Layout

The illustrations in this chapter show the keyboard layouts for the Japanese input methods. Online copies of these pictures are available at the specified locations. These figures are in Graphic Interchange Format (GIF) and you can display them on screen using Netscape, or include them in a Microsoft Word document.

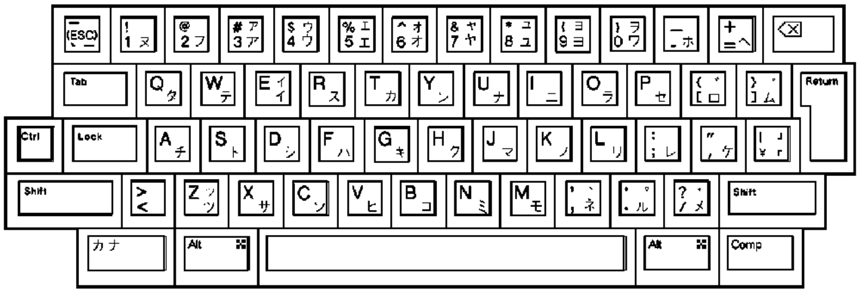
Figure 6-1: LK201-AJ Keyboard Layout



Required Keymap: japanese_lk201aj

Location of file: /japandoc/japan14.gif

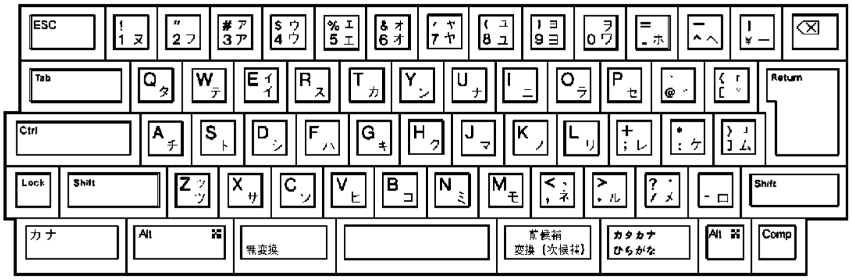
Figure 6-2: LK401-AJ Keyboard Layout



Required Keymap: japanese_lk401aj

Location of file: /japandoc/japan15.gif

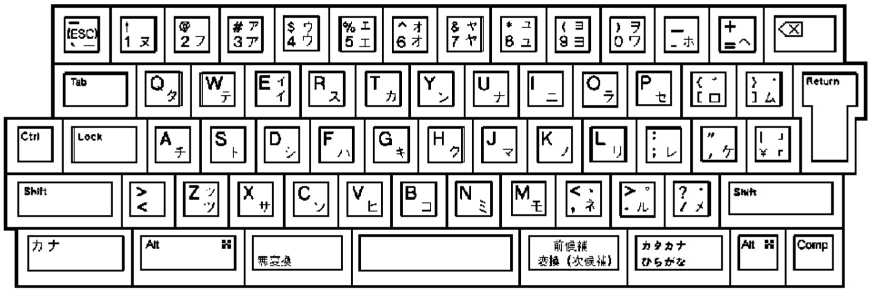
Figure 6-3: LK401-JJ Keyboard Layout



Required Keymap: japanese_lk401jj

Location of file: /japandoc/japan16.gif

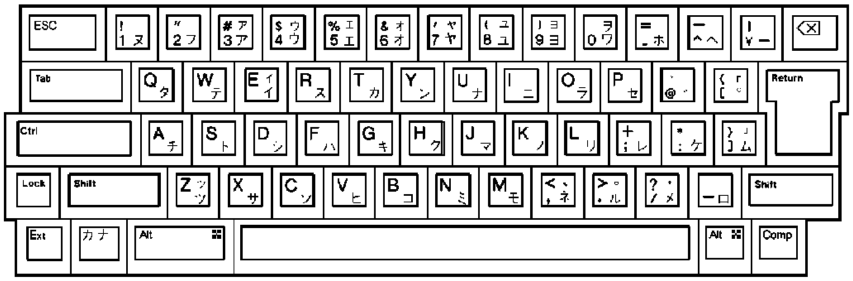
Figure 6-4: LK401-BJ Keyboard Layout



Required Keymap: japanese_lk401bj

Location of file: /japandoc/japan17.gif

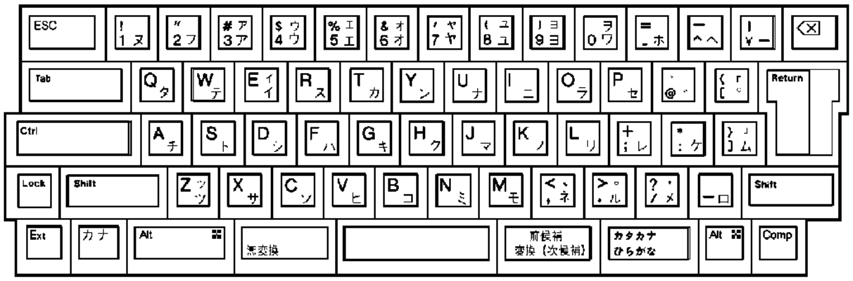
Figure 6-5: LK421-AJ Keyboard Layout



Required Keymap: japanese_lk421aj

Location of file: /japandoc/japan18.gif

Figure 6-6: LK421-JJ Keyboard Layout



Required Keymap: japanese_lk421jj

Location of file: /japandoc/japan19.gif

6.2 Keyboard Indicator

If you are using XDM, the keyboard indicator (kb_indicator) is automatically activated by the session manager and controls the compose lamp of the Japanese keyboards. For the keyboards which do not have a kana lamp (see KanaLED column in Table 6-1), add the following definition of the keyboard indicator to the Automatic Startup menu of the session manager and restart the session:

```
“/usr/bin/X11/kb_indicator -map”
```

Kana status will be displayed in the keyboard indicator window. A button in the keyboard indicator window controls the compose mode like the kana key on the keyboard.

If you are using CDE, you can manually execute the above command or add it to your personal profile.

6.3 JIS Keyboard

You may encounter difficulties with JIS layout keyboards (see Layout column in Table 6-1) before running a session manager with the appropriate keymap. These problems occur on systems running DIGITAL UNIX V4.0C or earlier, which assume ANSI layout keyboards by default.

On systems (excluding DEC3000, DEC4000, DEC7000, DEC10000 series) running DIGITAL UNIX 4.0D or later, you can set the keyboard type at console firmware. The steps are as follows:

1. Update the console firmware to the latest version.
2. Set the `kbd_hardware_type` language variable to appropriate values, as shown in Table 6-2.

   ```
   set kbd_hardware_type 411
   set language 50
   ```

3. Initialize console firmware:
   init
4. Confirm that the vlaues are set correctly:

   ```
   show
             .
             .
        kbd_hardware_type    LK411
        language             50
        language_name        Japanese (JIS)
             .
             .
   ```

Table 6-2: Japanese Keyboard Types

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

Note: LK20*, LK40* and LK42* are not listed because they are used for DEC3000, DEC4000, DEC7000, DEC10000 series.

7

Japanese Input Methods

There are two methods for inputting Japanese characters:

- Kana input for inputting Kana characters
- Conversion for inputting two-byte Kanji characters, Kana characters, letters and symbols defined in JIS X 0208. The DIGITAL UNIX software supports four input methods under this category:
  - Romaji-to-Kanji
  - Kana-to-Kanji
  - Internal code
  - JIS Ku-ten code

In this document, these four conversion methods are collectively referred to as Kana-Kanji Conversion.

7.1 Kana Input Method

7.1.1 Character Cell Terminals

For the Japanese versions of VT terminals (VT282-J, VT382-J, and VT383-J) Kana characters are entered in the Kana input mode. The Kana input mode is provided by the firmware of the terminals and is activated by pressing the [カナ] key, which is located on the lower-left side of the main keyboard. For details, see Chapter 6, Keyboards. In the Kana mode, you can enter Kana characters by pressing the keys on the main keyboard, as if entering English characters.

7.1.2 CDE and DECwindows Motif

The X Window System provides a mechanism for associating encodings of symbols which are usually printed on keycaps (called KeySyms) with physical or logical keys. Each key can be associated with a list of KeySyms. Depending on the Shift/Lock state, different KeySyms are obtained by pressing the same key. For instance, in the case of the American keyboard, the input of 3 or # is switched by the Shift state. The mechanism for supporting two KeySyms per key was insufficient for some languages, such as Hebrew and Japanese. Therefore, the Mode Switch mechanism was added to the X Window System to allow switching of Group 1 and Group 2 characters, where Group 1 is for ASCII KeySyms and Group 2 is for native language characters. That is, the Mode Switch mechanism works as a Group Shift key for these languages.

In addition, you can input different sets of characters through the same keyboard by changing the associated KeySyms.

In the X Consortium's release of the X Window System, the Mode Switch On state is effective only for entering one single character. After a character is entered, the Mode Switch state is reset. In the DIGITAL UNIX implementation, the Mode Switch mechanism is enhanced to support both single-shot switching and lock-shift switching, depending on the keymap definition. In the latter case, once the Mode Switch state is set, it remains unchanged until the Mode Switch key is pressed again to modify the state.

DIGITAL UNIX windows environment (CDE and DECwindows Motif) provides the keymaps shown in Table 7-1 for setting Japanese KeySyms.

Table 7-1: Japanese Keymaps

Keymap	Keyboard
JAPANESE_LK201AJ	LK201-AJ
JAPANESE_LK401AJ JAPANESE_LK401AJ_ESC	LK401-AJ/LK40W-AJ
JAPANESE_LK401JJ	LK401-JJ
JAPANESE_LK401BJ JAPANESE_LK401BJ_ESC	LK401-BJ
JAPANESE_LK421AJ	LK421-AJ
JAPANESE_LK421JJ	LK421-JJ
JAPANESE_LK411AJ JAPANESE_LK411AJ_ESC	LK411-AJ/LK41W-AJ
JAPANESE_LK411JJ	LK411-JJ/LK41W-JJ
JAPANESE_PCXAJAA	PCXAJAA/LK97W-AJ

All the keymaps in Table 7-1 support locking-shift mode switching. With these keymaps, you can enter English characters in the Mode Switch Off state and Kana characters in the Mode Switch On state.

The effect of setting the Mode Switch on is similar to that of entering the Kana input mode on Japanese VT terminals.

7.1.2.1 Loading a Keymap

The procedures for activating the Kana input mode on VT terminals and CDE/DECwindows Motif are similar. You must load a Japanese keymap in order to input Kana characters on DECwindows Motif; for VT terminals, you can skip this step.

To load a Japanese keymap, choose the Keyboard... menu item from the Options menu of Session Manager. The keyboard dialog box displays and you can choose one of the Japanese keymaps according to the type of the keyboard connected to your workstation. For instance, if you are using a Japanese LK401-AJ keyboard, choose the japanese_lk401aj item, and so on.

The keymap is a global attribute to a display device. After you select a keymap, the keymap is effective to all client applications running on your display device. If you plan to run applications in different languages, check for any conflicts between the keymap you are using and the keymap required by those languages.

7.1.3 Activating Kana Input Mode

The key sequences for activating the Kana input mode are shown in Table 7-2.

Table 7-2: Key Sequence for Invoking Kana

Keyboard Type	Kana Key Sequence
VT Terminals	カナ
CDE/DECwindows Motif	
• LK201-J	[カナ/Space]
• Others	カナ

Note

The [カナ] key is equivalent to the [Compose] key on the standard LK201 keyboard or the [Left-Compose] key on the standard LK401 keyboard.

You can press the appropriate Kana key sequence to toggle between English input mode and Kana input mode. If a keyboard has a mode-switch LED for Kana mode, it is switched on or off according to the input mode. (See Table 6-1 for keyboard LEDs.) The DIGITAL UNIX Keyboard Indicator utility provides a visual feedback of the input mode for keyboards without the LED. For details, see *Writing Software for the International Market.*

7.1.4 English Character Input

To input English characters, make sure the Kana mode is set off. In this state, you can input English letters, numeric characters, and punctuation marks by pressing keys on the main keyboard. On a Japanese keyboard, the English characters or symbols that are entered are printed on the left-hand side of the keycaps.

7.1.5 Kana Character Input

To input Kana characters, make sure the input mode is set to Kana. In this mode, you can input Kana characters by pressing keys on the main keyboard.

For example, if you press the [A] key when the Kana mode is set on, the Hiragana character [ち] or the Katakana character [チ] are entered, depending on the default application setting or the input method parameters. For instance, you can control whether Hiragana characters or Katakana characters are entered by including the following resource in the IMlib profile data files `im_profile*.dat` file.

```
DEC-JAPANESE.ECHO.kana : hiragana
```

The default profile data files provided by the DIGITAL UNIX operating system are found in `$I18NPATH/jsy/imlib`. You can change the default value of hiragana in the resource file to katakana or jiskana.

On a Japanese keyboard, the Katakana characters that are entered are printed on the right side of the keycaps.

For example, to enter Kana mode, press [kana/Space] on the LK201-AJ keyboard, Right-[Control] on the PCXAJ-AA keyboard, or [kana] on the other Japanese keyboards. Press the same key again to go back to ASCII mode.

For details about Japanese keymaps, see Chapter 6.

7.2 Kana–Kanji Conversion

7.2.1 Character–Cell Terminals

For character-cell terminal applications, Japanese input methods are provided either by the applications themselves or through the internationalized `tty` terminal driver.

If your application handles the input of Japanese data, refer to the application’s documentation for the details of Japanese input methods.

For details about customizing your terminal driver settings using the `stty` command to enable Kana-Kanji conversion, see *Writing Software for the International Market.*

7.2.2 CDE/DECwindows Motif

DECwindows Motif implements Japanese input methods in the form of an independent process, called the input server. The Japanese input server is an X client process that works on a standard X-server if the required Japanese fonts are installed. This means that the process can run on any system that can access your X display device, including the device itself.

You can write internationalized applications using standard X11R6 or OSF/Motif R1.2 application programming interfaces to talk to this input server. For details on developing internationalized software with X11R6 and OSF/Motif R1.2, see *Writing Software for the International Market.*

Before you can input Japanese data, you must start the Japanese input server on your workstation or any system that can be accessed by your workstation through the network.

To start the input server on the workstation that you are using, enter the following command:

```
% /usr/bin/x11/dxjim &
```

To start the input server on a remote system, enter the following command on that system:

```
% setenv DISPLAY <display-name>
% /usr/bin/x11/dxjim &
```

where <display-name> is the display name for your workstation.

Once the Japanese input server is invoked, the applications that were internationalized with X11R6 Xlib or OSF/Motif R1.2 to support Japanese can communicate with the server to provide input method services.

7.2.3 CDE/DECwindows Motif Interface

You can interact with the Japanese input server through a Motif-style user interface that allows an input method to provide feedback about the data being edited. This feedback helps you compose a character, list choices for selection, provide options for customizing the input server, and so on.

The X Input Method specification defines the three input areas described in Table 7-3.

Table 7-3: Window Input Areas

Area	Description
Auxiliary area	An option menu helps you customize the Japanese input methods and the input method window.
Status area	This area displays the internal state of the Japanese input method that is critical to you.
Pre-edit area	This area displays the intermediate text that is being composed. This area also provides a lookup region which displays a list of valid candidates for the input key sequences.

7.2.4 Interaction Styles

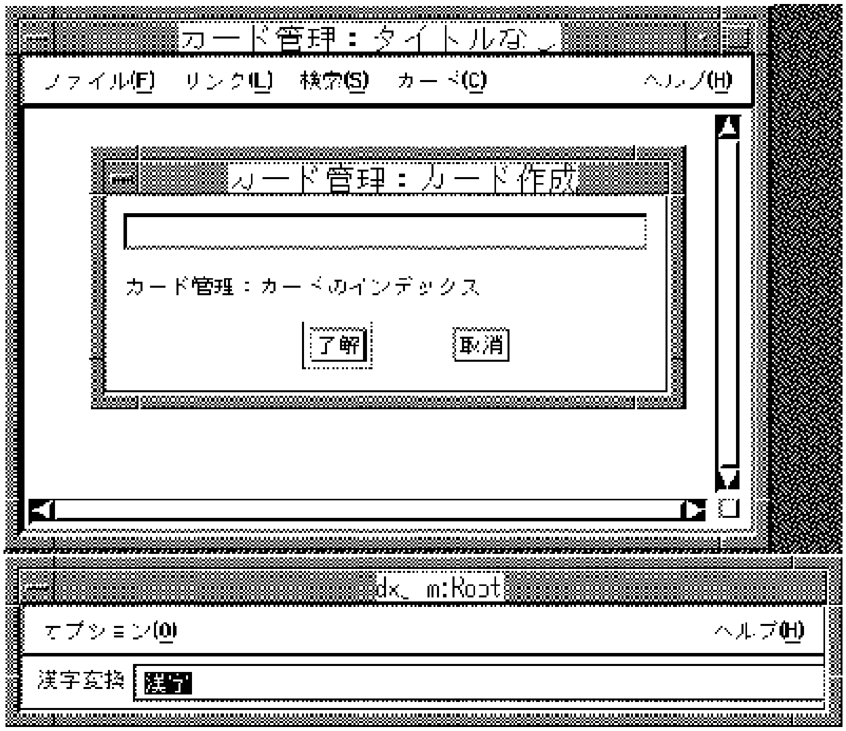
The use of the input areas depends on the interaction style (or pre-edit style) selected for the application. The Japanese input server supports three interaction styles:

- Root window
- On-the-spot
- Over-the-spot

7.2.4.1 Root Window

If you want to display the pre-edit data in a separate input window which lies outside the application window, choose the root window interaction style. The input window is scaleable and movable; you can modify its size and position to meet your preferences.

Figure 7-1: Root Window Interaction Style



You can press [Shift/Space][1] to toggle between the Japanese input mode and the English input mode. When you are in the Japanese mode, the keys that you press are displayed inside this root window. Conversion operations can be performed on this pre-edit data to compose Japanese characters. When the pre-edit data is converted to the Japanese characters that you want, press [Return] to accept the Japanese characters and send them to the application.

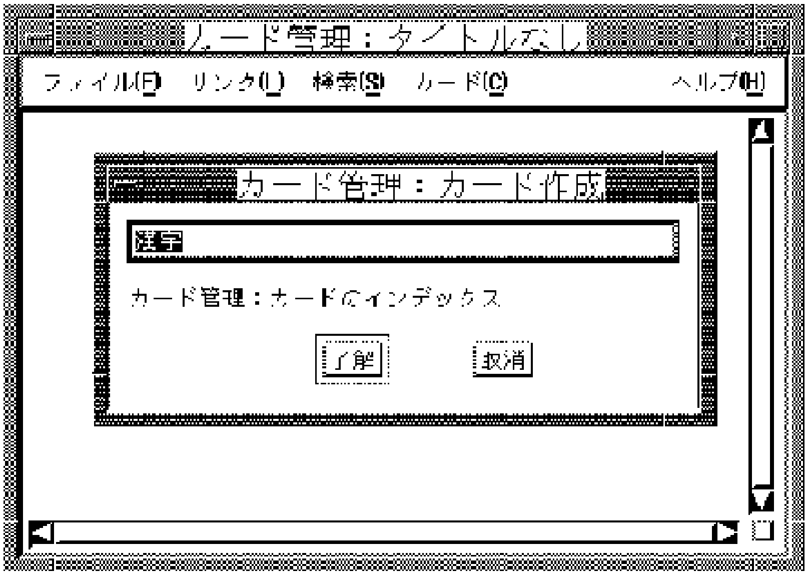
7.2.4.2 On-the-Spot

If you want to focus your attention at the point of insertion during Japanese data input, you can choose the on-the-spot interaction style. In this style, the Japanese input server directs your applications to display the pre-edit data in a way which is consistent with the appearance of the existing data in the input field. For instance, an application can select the font size and typeface to display the pre-edit data and control how the data is

[1] This toggle key sequence is customizable through the Japanese input server user interface.

highlighted. Formatting of existing data can also be performed to allow the pre-edit data to be displayed at the insertion point.

Figure 7-2: On-the-Spot Interaction Style



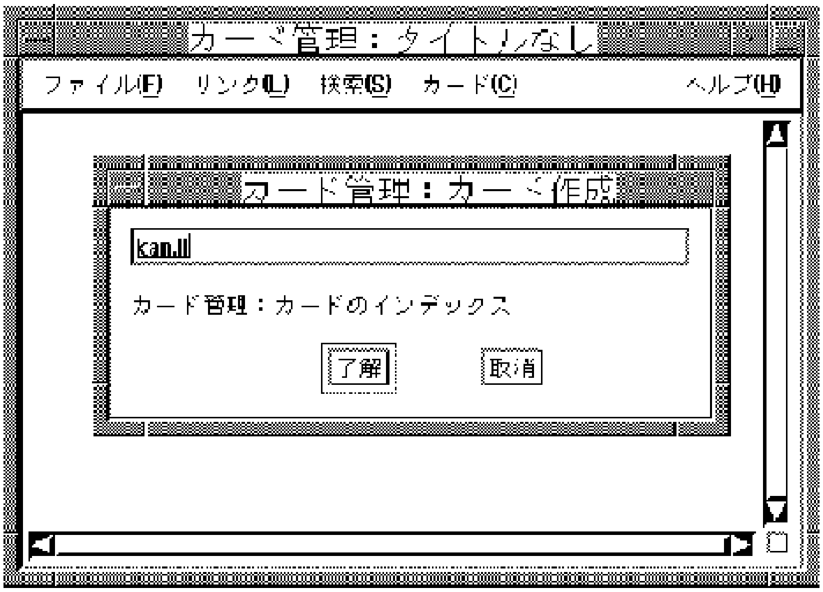
漢字

カード管理：カードのインデックス

7.2.4.3 Over-the-Spot

Another alternative which allows you to perform Japanese data input at the point of insertion is the over-the-spot interaction style. In this style, an input window is created when you invoke the Japanese input mode; the window is placed over the current point of insertion. The pre-edit data entered is displayed in this window. The look-and-feel is similar to that of the on-the-spot style, except that applications do not control the display attributes of the pre-edit data and will not perform any formatting to accommodate the pre-edit. Therefore, there is less communication overhead between the Japanese input server and applications. The performance of data entry through the over-the-spot interaction style is better.

Figure 7-3: Over-the-Spot Interaction Style



You can press [Shift/Space][2] to toggle between the Japanese input mode and the English input mode. When you are in the Japanese input mode, the keys that you press are underlined and displayed in a pre-edit window right over the point of insertion. Conversion operations can be performed on this pre-edit data to compose Japanese characters. When the pre-edit data is converted to the Japanese characters that you want, press [Return] to accept the Japanese characters and send them to the application.

7.2.5 Customization of Key Sequences

By default, the key sequence used to invoke the Japanese/English input mode and the one to confirm the acceptance of converted Japanese data are predefined as <Shift/Space> and [Return] respectively. You can change these default key sequences to meet your needs. You can customize the following key sequences from the Options menu of the Japanese input server:

- Start Japanese Input Mode
- End Japanese Input Mode (Start English Input Mode)
- Confirm Input

[2] This toggle key sequence is customizable through the Japanese input server user interface.

The dialog box provides an interface that you can use to customize a key sequence. It allows you to select a trigger key and to toggle the on/off state of the Ctrl, Alt and Shift modifiers. The trigger keys that you can choose include NoSymbol, [F1] **-** [F20], [Space], [Return], [Compose], and [A**]** through [Z]. If you choose NoSymbol, no invocation sequence is provided for the selected action.

For each modifier key, you can select the on/off state with the toggle buttons identified in Table 7-4.

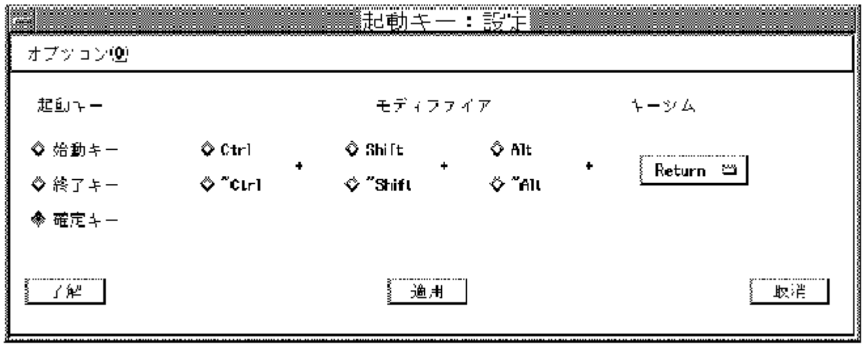
Table 7-4: Modifier State Customization

Modifier	On State	Off State
Ctrl	“Ctrl”	“~Ctrl”
Shift	“Shift”	“~Shift”
Alt	“Alt”	“~Alt”

The tilde (~) sign means that you should not press that modifier key when invoking the action. In addition to the on/off state, you can deselect both of the states for a modifier key, that is, neither state is selected. To do this, click the toggle button that is currently set on. If you deselected a modifier, the invocation key with or without holding the modifier key can be accepted by the input server.

When one of the invocation key sequences is selected, for example, the Confirm Input key sequence, the state of the toggle switches and the trigger key displayed on the dialog box is updated to reflect the current value.

Figure 7-4: Customization of Invocation Key Sequences



If you want to change the Confirm Input key sequence to [Ctrl/Return], select the Ctrl, ~Alt, and ~Shift buttons.

7.2.6 Selecting an Interaction Style

You can specify the priority of the interaction styles of DECwindows Motif applications by specifying the VendorShell resource, XmNpreeditType. By default, the resource value is “onthespot,overthespot,offthespot,root”. This list is in priority order. The first style will be used if it is available in an input method, else the second, and so on.

You can use one of the following methods to choose your preferred interaction style:

- Use the command line option `-xrm` to specify the resource value when you start an application. For example, the following command will start Cardfiler with the root window interaction style:

  ```
  % dxcardfiler -xrm '*preeditType: root' &
  ```

  To start Cardfiler with the over-the-spot interactive style, you can enter:

  ```
  % dxcardfiler -xrm '*preeditType: overthespot' &
  ```

  To start Cardfiler with the on-the-spot interactive style, you can enter:

  ```
  % dxcardfiler -xrm '*preeditType: onthespot' &
  ```

- Use the Session Manager’s Options menu in XDM:

  — From the Session Manager’s Options menu, select Input Style.…

  — In the popup Input Style Options window, click on one of the pre-edit styles.

  — Navigate the selected pre-edit style through the Style Preference list by clicking on the up-arrow or down-arrow button.

  — Repeat these steps until the priority list is the one you desire.

After you select your preferred interaction style, the applications you invoke will start up with the new setting.

7.2.7 Types of Conversion

The input of Japanese characters is implemented through the software conversion of phonetic codes representing their pronunciation. There are four types of input conversion methods:

- Romaji-to-Kanji
- Kana-to-Kanji
- Internal Code
- JIS Ku-ten Code (Row-Column Code)

7.2.8 Conversion Operations

The Romaji-to-Kanji and Kana-to-Kanji methods are performed by the conversion functions provided by the operating system. The conversion process involves the operations described in Table 7-5.

Table 7-5: Kana-to-Kanji Conversion Operations

Operation	Description
HENKAN	Converts the phonetic code input
JI-KOUHO	Selects another candidate for conversion
MAE-KOUHO	Selects the previous conversion candidate
ICHIRAN	Lists all conversion candidates
KAKUDAI	Lengthens the current phrase
SHUKUSHOU	Shortens the current phrase
JI-BUNSETSU	Moves to the next phrase
MAE-BUNSETSU	Moves to the previous phrase
HIRAGANA	Converts current input to Hiragana
KATAKANA	Converts current input to Katakana
ZENKAKU	Converts current input to full form alphabets
HANKAKU	Converts current input to half form alphabets
KIGOU	Converts current input to special graphic character
MU-HENKAN	No conversion is applied

These operations are invoked by using the key sequences assigned to them, which vary according to different facilities, keyboard types, or user’s customization. Several Japanese keyboards, such as LK401-JJ, LK401-BJ, and LK421-JJ provide special keys for the conversion input methods. These are shown in Table 7-6.

Table 7-6: Japanese Conversion Keys

Key	Key Label/Unshift	Function/Unshift	Key Label/Shift	Functions/Shift
1	変換（次候補）	HENKAN or JI-KOUHO	前候補	MAE-KOUHO
2	無変換	MU-HENKAN	無変換	MU-HENKAN
3	ひらがな	HIRAGANA	カタカナ	KATAKANA

For details about Japanese conversion keys, see Chapter 6.

7.2.9 Conversion Key Bindings

Table 7-7 through Table 7-11 summarize the conversion operations and their associated key bindings for the following facilities:

- jvms
- jeve
- taro
- keypad
- control

All Japanese keyboards support these input key sequences. In addition to these sequences, several keyboards also provide additional keys for issuing Japanese input conversion commands.

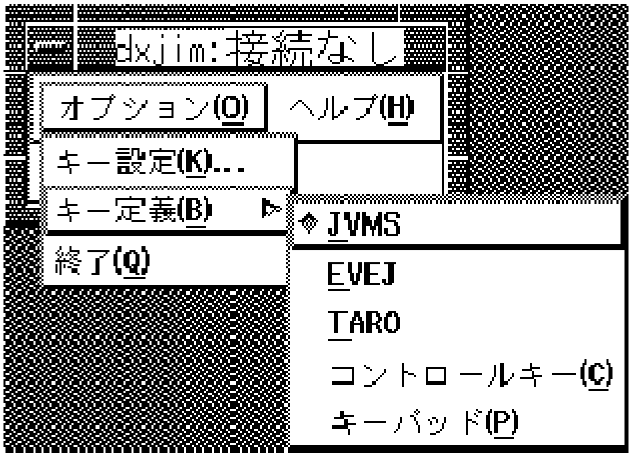
For the operating system commands and utilities, you can select your preferred key binding by defining the JSYKKSEQ environment variable or by using your private settings, which are stored in the `~/.jsykkseq`. file The system default conversion key sequences are stored in the `$I18NPATH/usr/skel/ja_JP` file.

Note

The taro key binding is not supported by the DIGITAL UNIX operating system.

For CDE and DECwindows Motif, the key binding can be selected through the Options menu of the Japanese input server.

Figure 7-5: Customizing Japanese Input Key Binding



You can select your favorite key binding from the Key Binding cascade menu.

The set of conversion key sequences are all defined in IMlib profile data files located in the `$I18NPATH/usr/jsy/imlib` directory.

Note that in Tables 7-7 through 7-11, the ICHIRAN operation is supported only in CDE and DECwindows Motif.

Table 7-7: jvms Conversion Key Sequences

Operation	All Keyboards	LK401-BJ, LK401-JJ, LK411-AJ(LK41W-AJ), LK411-JJ(LK41W-JJ), LK421-JJ, PCXAJ-AA, LK97W-AJ
HENKAN	[Ctrl/Space] or [Ctrl/@] or [Ctrl/G][Ctrl/K]	変換(次候補)
JI-KOUHO	[Ctrl/Space] or [Ctrl/@] or [Ctrl/G][Ctrl/K]	変換(次候補)
MAE-KOUHO	[Ctrl/G][Ctrl/Space] or [Ctrl/G][Ctrl/@] or [Ctrl/G][Ctrl/L]	前候補
ICHIRAN	[Ctrl/‘]	
SHUKUSHOU	[Ctrl//] or [Ctrl/] or [Shift/LeftArrow]	
KAKUDAI	[Ctrl/G][Ctrl//] or [Ctrl/G][Ctrl/] or [Shift/RightArrow]	
JI-BUNSETSU	[Ctrl/P] or [Shift/DownArrow]	
MAE-BUNSETSU	[Ctrl/G][Ctrl/P] or [Shift/UpArrow]	
HIRAGANA	[Ctrl/L]	ひらがな
KATAKANA	[Ctrl/K]	カタカナ
ZENKAKU	[Ctrl/F]	[Alt/ひらがな]
HANKAKU	[Ctrl/G][Ctrl/F]	[Ctrl/Alt/ひらがな]
KIGOU	[Ctrl/]]	[Alt/無変換]
MU-HENKAN	[Ctrl/N]	無変換

Table 7-8: jeve Conversion Key Sequences

Operation	All Keyboards	LK401-BJ, LK401-JJ, LK411-AJ(LK41W-AJ), LK411-JJ(LK41W-JJ), LK421-JJ, PCXAJ-AA, LK97W-AJ
HENKAN	[Ctrl/Space] or [Ctrl/@] or [PF1][Space]	変換(次候補)
JI-KOUHO	[PF1][Ctrl/Space] or [Ctrl/@] or [PF1][Space]	変換(次候補)
MAE-KOUHO	[PF1][Ctrl/Space] [PF1]	前候補
ICHIRAN	[Ctrl/@]	
SHUKUSHOU	[Ctrl/‘]	
KAKUDAI	[Ctrl/A]	
JI-BUNSETSU	[PF1][Ctrl/A]	
MAE-BUNSETSU	[Ctrl/P]	
HIRAGANA	[PF1][Ctrl/P]	ひらがな
KATAKANA	[Ctrl/H]	カタカナ
ZENKAKU	[Ctrl/K]	[Alt/ひらがな]
HANKAKU	[Ctrl/F]	[Ctrl/Alt/ひらがな]
KIGOU	[Ctrl/E]	[Alt/無変換]
MU-HENKAN	[PF1][Z] [Ctrl/N]	無変換

Table 7-9: taro Conversion Key Sequences

Operation	All Keyboards	LK401-BJ, LK401-JJ, LK411-AJ(LK41W-AJ), LK411-JJ(LK41W-JJ), LK421-JJ, PCXAJ-AA, LK97W-AJ
HENKAN	[Space]	変換(次候補)
JI-KOUHO	[Space]	変換(次候補)
MAE-KOUHO	[UpArrow]	[前候補
ICHIRAN	[Ctrl/‘]	
SHUKUSHOU	[RightArrow]	
KAKUDAI	[LeftArrow]	
JI-BUNSETSU	[DownArrow]	
MAE-BUNSETSU	N/A	
HIRAGANA	[F11]	ひらがな
KATAKANA	[F12]	カタカナ
ZENKAKU	[F14]	[Alt/ひらがな]
HANKAKU	[F13]	[Ctrl/Alt/ひらがな]
KIGOU	[Ctrl/]]	[Alt/無変換]
MU-HENKAN	N/A	無変換

Table 7-10: Keypad Conversion Key Sequences

Operation	All Keyboards	LK401-BJ, LK401-JJ, LK411-AJ(LK41W-AJ), LK411-JJ(LK41W-JJ), LK421-JJ, PCXAJ-AA, LK97W-AJ
HENKAN	[KP7]	変換(次候補)
JI-KOUHO	[KP7]	変換(次候補)
MAE-KOUHO	[PF1][KP7]	前候補
ICHIRAN	[Ctrl/‘]	
SHUKUSHOU	[KP4]	
KAKUDAI	[PF1][KP4]	
JI-BUNSETSU	[KP5]	
MAI-BUNSETSU	[PF1][KP5]	
HIRAGANA	[KP1]	ひらがな
KATAKANA	[KP2]	カタカナ
ZENKAKU	[KP3]	[Alt/ひらがな]
HANKAKU	[KP9]	[Ctrl/Alt/ひらがな]
KIGOU	[KP8]	[Alt/無変換]
MU-HENKAN	[KP0]	無変換

Table 7-11: Control Conversion Key Sequences

Operation	**All Keyboards**	**LK401-BJ, LK401-JJ, LK411-AJ(LK41W-AJ), LK411-JJ(LK41W-JJ), LK421-JJ, PCXAJ-AA, LK97W-AJ**
HENKAN	[Ctrl/K]	変換(次候補)
JI-KOUHO	[Ctrl/K]	変換(次候補)
MAI-KOUHO	[Ctrl/L]	前候補
ICHIRAN	[Ctrl/‘]	
SHUKUSHOU	[Ctrl/A]	
KAKUDAI	[Ctrl/X]	
JI-BUNSETSU	[Ctrl/P][R]	
MAE-BUNSETSU	[Ctrl/P][L]	
HIRAGANA	[Ctrl/G]	ひらがな
KATAKANA	[Ctrl/N]	カタカナ
ZENKAKU	[Ctrl/E]	[Alt/ひらがな]
HANKAKU	[Ctrl/T]	[Ctrl/Alt/ひらがな]
KIGOU	[Ctrl/F]	[Alt/無変換]
MU-HENKAN	[Ctrl/B]	無変換

7.2.10 Romaji-to-Kanji Input Method

The Romaji-to-Kanji input method converts Romaji to Kanji characters. You input English characters to represent the phonetic pronunciation and then convert them to the correct ideographic Kanji.

You use the HENKAN command to start Romaji-to-Kanji conversion. To start the HENKAN command, enter the appropriate key sequence as defined in Table 7-7 through Table 7-11.

For example, the Romaji representation of “日本”, which means “Japan” in English, is “nihon”. To enter “日本”, type “nihon” through the main keyboard and then press the HENKAN key sequence to convert “nihon” to “日本”.

A Romaji string can represent more than one unique Japanese phrase. For example, the Romaji string “jin” can represent “人”, “陣”, “ジン”, and so on. If the converted string is not the one you want, press the JI-KOUHO or MAE-KOUHO key sequence to select the next string or the previous string. To display the complete list of characters corresponding to the Romaji code entered, press the ICHIRAN key.

The Romaji entered is divided into phrases (called “BUNSETSU” in Japanese) and each phrase is converted to Kanji independently. The same Romaji string can be divided differently to form different groupings of phrases. For example, the Romaji string “kanji” can be divided into two phrases as “kan-ji” or three phrases as “ka-n-ji”. If the grouping of phrases is not the one desired, press the SHUKUSHOU or KAKUDAI key sequence to modify the division and then revise the conversion of phrases.

Each of the divided phrases can be converted independently. When you press the HENKAN key sequence to convert Romaji containing more than one phrase, the first phrase is highlighted to show that you can further convert this phrase with the JI-KOUHO or MAE-KOUHO command. To convert the next phrase or the previous phrase, use the JI-BUNSETSU or MAE-BUNSETSU command to move to the next phrase or the previous phrase and apply the JI-KOUHO or MAE-KOUHO conversion command.

If you want to convert the Romaji to Hiragana or Katakana, press the HIRAGANA or KATAKANA key sequence. For example, the Romaji “kanji” can be converted to a Hiragana string “かんじ” or a Katakana “カンジ”.

7.2.10.1 Romaji Input

Using the Romaji-to-Kanji input method, you can input the Romaji in Alphabet input mode.

For example, to input “日本” using the Romaji-to-Kanji method, input the Romaji “nihon” by pressing the [N][I][H][O][N] keys. Usually, the input data is highlighted to indicate the Romaji for conversion.

Enter [N][I][H][O][N]

```
NIHON
```

7.2.10.2 Conversion (HENKAN)

After entering the Romaji, press the key sequence designated for the HENKAN operation (see Table 7-7 through Table 7-11). The Romaji “nihon” is converted to “日本”.

```
日本_
```

7.2.10.3 Candidate Selection (JI-KOUHO and MAE-KOUHO)

The string “日本” is not the only string represented by the Romaji “nihon”. “二本” is another valid string. If “日本” is not the string you want, press the JI-KOUHO key sequence to display another string. For some applications, a list of valid strings is displayed to facilitate selection.

Note

The display of multiple strings is not supported in the CSText widget.

Press the JI-KOUHO key sequence:

```
二本_
1. 日本　2. 二本
```

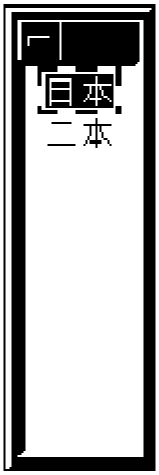
If you want to select the previous string, press JI-KOUHO. The previous string is displayed at the cursor position:

```
日本_
1. 日本　2. 二本
```

7.2.10.4 Display All Candidates (ICHIRAN)

If you are using CDE or DECwindows Motif, you can issue the ICHIRAN command to list all valid conversion strings. When you press the ICHIRAN key sequence, a selection box pops up and displays all candidates for your selection:

Figure 7-6: Multiple Candidates



7.2.10.5 Grouping Of Phrases (KAKUDAI and SHUKUSHOU)

The Romaji “nihon” can be divided in various ways into different groupings of phrases, such as “nihon”, “niho-n”, and “ni-hon”. You can use the KAKUDAI and SHUKUSHOU commands to modify the division of Romaji into phrases. For example, if you press the

SHUKUSHOU key sequence after “nihon” is converted to “日本”, the Romaji “nihon” changes to “niho-n” and the Japanese phrases for “niho” and “n” are displayed in place of “日本”:

にほん_

Here “にほ” is the Japanese string for “niho” and “ん” is the Japanese string for “n”. To facilitate conversion the phrase being converted is highlighted.

If you press the SHUKUSHOU key sequence again, the grouping changes from “niho-n” to “ni-hon” and the Japanese phrases for “ni” and “hon” are displayed.

荷本_

Here “荷” is the Japanese string for “ni” and “本” is the Japanese string for “hon”. The first phrase, “荷”, is now highlighted and you can convert this string further to other candidates, if applicable. For example, if you press the JI-KOUHO key sequence, the next candidate for “ni” is displayed:

```
似本_
 1.荷 2.似 3.煮 4.二 5.弐 6.尼 7.丹 8.仁
```

To reverse the SHUKUSHOU command, press the KAKUDAI key sequence.

7.2.10.6 Navigation Between Phrases (JI-BUNSETSU and MAE-BUNSETSU)

To move to the next phrase or the previous phrase for conversion, issue the JI-BUNSETSU or MAE-BUNSETSU command. In this example, you move to the “本” phrase by pressing the JI-BUNSETSU key sequence:

荷本_

The phrase “本” is highlighted.

7.2.10.7 Hiragana and Katakana Input

If you want to convert Romaji or a divided phrase (that is, the highlighted string on the screen) to a Hiragana string, press the HIRAGANA key sequence. In the previous example, the highlighted character “荷” is converted to the Hiragana character “に”:

に本_

Similarly, you can convert the string to a Katakana string by pressing the KATAKANA key sequence. In this example, the character will be converted to “ニ”:

ニ本_

7.2.10.8 Full-Form and Half-Form Letters (ZENKAKU and HANKAKU)

If you want to convert Romaji (even after conversion is taken place) to a full-form or a half-form letter string, press the ZENKAKU or HANKAKU key sequence. Full-form letters are two-byte letter characters defined in JIS X 0208 while half-form letters are single-byte letter characters defined in ASCII.

Press ZENKAKU:

ｎｉｈｏｎ＿

Press HANKAKU:

ｎｉｈｏｎ＿

Note

The ZENKAKU and HANKAKU commands affect the whole Romaji code even if you have converted individual phrases to Japanese characters.

7.2.10.9 No Conversion (MU-HENKAN)

If you do not want to apply any conversion or restore the original Romaji code after applying conversion, press the MU-HENKAN key.

7.2.11 Kana–to–Kanji Input Method

The Kana-to-Kanji input method converts Kana into Kanji characters. Each Kana is associated with a key on the keyboard. To invoke the Kana-to-Kanji input method, invoke the “kana” input mode (see Section 7.1, Kana Input Method) and input the Kana characters. Use the HENKAN command to start Kana-to-Kanji conversion by pressing the appropriate key sequence. The Kana input is divided into phrases (“BUNSETSU” in Japanese) and converted to Kanji. If the grouping of phrases is not the one desired, press the SHUKUSHOU or KAKUDAI key sequence to revise the grouping of Kana codes and revise the conversion. If the converted phrase displayed is not the desired phrase, press the JI-KOUHO or MAE-KOUHO key sequence to select the next or previous phrase. To display the complete list of characters corresponding to the Kana code input, press the ICHIRAN key sequence.

The behavior of the Kana-to-Kanji input method, in many aspects, is very similar to that of the Romaji-to-Kanji input method. Please refer to Section 7.2.10, Romaji-to-Kanji Input Method for further information.

7.2.11.1 Enable Kana Mode

To enter the phrase “変換” using the Kana-to-Kanji input method, you should first invoke the Kana input mode by pressing the appropriate key sequence, as described in Section 7.1, Kana Input Method. On a Japanese keyboard which has a Kana LED the LED with a label “カナ” will be switched on. If the Keyboard Manager is running on your display, the icon will be changed to indicate the Kana input mode.

7.2.11.2 Kana Input

The Hiragana or Katakana code for “変換” is “へんかん” or “ヘンカン” respectively. Therefore, you must press the following keys to enter the Kana code: ヘンカン. These keys are labeled as [=] [Y] [T] [Y] on a standard LK201 or LK401 keyboard. Whether Hiragana characters or Katakana characters are echoed depends on the customization, as described in Section 7.1, Kana Input Method. By default, Hiragana characters are echoed.

Press ヘンカン or [=] [Y] [T] [Y]:

へんかん_

7.2.11.3 Conversion (HENKAN)

After entering the Kana code, press the HENKAN key sequence (see Table 7-7 through Table 7-11). The Kana character “へんかん” is converted to “変換”:

変換

7.2.12 Internal Code Input Method

Each Kanji character is assigned a unique identification code in the system. These identification codes can be used to enter individual Kanji characters. *The DEC Kanji Code Book* (Part # - AA-A056C-TE-JO) lists the characters in the DEC Kanji character set, along with their respective internal codes.

7.2.12.1 Internal Code Input

To enter a Japanese character using the Internal Code input method, press the [X] key to indicate the input of hexadecimal code, followed by the four hexadecimal numbers which represent the internal code of the desired character. For example, the internal code for “院” is B1A1. To input the internal code enter [X] [B] [1] [A] [1]:

```
xb1a1
```

7.2.12.2 Conversion (KIGOU)

Press the KIGOU key sequence to convert the internal code to the desired character.

院_

7.2.13 JIS Ku–ten Code Input Method

Each character in the JIS X 0208 character set is assigned a particular position in the character plane to uniquely identify that character. To enter Japanese characters using the Ku-ten input method, specify the row and column of the desired character in the character plane. The Ku-Ten code for characters can be found in the *DEC Kanji Code Book* (Part # - AA-A056C-TE-JO).

7.2.13.1 JIS Ku-ten Code Input

Press the [J] key to indicate the input of row-column code, followed by the 2-digit row-column numbers that represent the desired character. For example, the character “院” is placed in the first column of the 17th row. The JIS Ku-ten code is 1701. Therefore, to input the JIS Ku-ten, press [J] [1] [7] [0] [1]:

```
j1701
```

7.2.13.2 Conversion (KIGOU)

Press the KIGOU key sequence to convert the internal code to the desired character:

院

8

Japanese Printing Support

This chapter introduces the Japanese printing support provided by the DIGITAL UNIX operating system software. It describes the supported printers, the print file formats, features, and the maintenance procedures for supporting Japanese printing.

8.1 Supported Printers

8.1.1 Text Printers

The DIGITAL UNIX software supports text printers with built-in Japanese fonts.

8.1.2 PostScript Printers

The DIGITAL UNIX software supports Japanese printing on PostScript printers with resident fonts.

For the details about the supported printer types and print filters, see Chapter 4, Local Language Devices.

8.2 Print File Formats

The DIGITAL UNIX software supports the printing of mixed English and Japanese characters in the following print file formats:

- Plain text files on text and PostScript printers
- Files with the nroff control sequences (for printing with underline, superscript, subscript and bold attributes) on text and PostScript printers
- PostScript files on PostScript printers

The print filters for PostScript printers automatically detect the format of a print file and convert it to the proper format for printing.

8.3 Printing Features

8.3.1 Software On-Demand Font Loading (SoftODL)

SoftODL is a mechanism through which a terminal or a bitmap printer downloads the relevant bitmap font information for a user-defined character (UDC) at the time the character is displayed or printed. The Japanese bitmap printers which can support this feature include:

- LA84-J
- LA86-J
- LA88-J
- LA90-J
- LA280-J
- LA380-J
- LN03S-J
- DEClaser 2300
- DEClaser 2400

8.3.2 Codeset Conversion

The DIGITAL UNIX software includes a codeset conversion mechanism to print text files that have a codeset different from the one used by the printer. For printers with built-in or downloaded Japanese fonts, the codeset of the printer should be defined to match the codeset of the built-in fonts.

8.4 Commands and Daemons

Before users can configure printers to utilize the printing features supported by the DIGITAL UNIX software, certain commands should be understood. This section discusses these commands in detail; the next section shows how they are used for configuration.

8.4.1 Country Specific Options of the lpr Command

The `-A` option to the `lpr` command is used to pass the country-specific parameters. The parameters found in the `-A` option can also be set as defaults in the `/etc/printcap` file by the `ya` capability. For example, the parameters can be specified in the `-A` option as:

```
% lpr -A "flocale=ja_JP.deckanji
plocale=ja_JP.sdeckanji" <file>
```

The same set of parameters are defined in the `/etc/printcap` file as:

```
:ya="flocale=ja_JP.deckanji
plocale=ja_JP.sdeckanji":\
```

The parameters supplied by the `-A` option to the `lpr` command override the corresponding default values in the `/etc/printcap` file.

The following parameters are applicable to Japanese printing:

- `flocale=<any valid locale>`

  Specifies the locale for the source text file. The printer filters use this locale to validate the characters inside the source text file. If this value is not set properly, the text is interpreted using the current locale. In Japanese printing environment, this value is particularly important for the `lpr` command to correctly interpret the characters. Moreover, if the `plocale` option is also set, the `lpr` command performs codeset conversion for the source text file.

- `plocale=<any valid locale>`

  Specifies the locale for the printer. If the printer has built-in fonts, the `plocale` value should match the codeset of the built-in fonts. If the printer employs the font faulting mechanism, the `plocale` value should match the font used for printing the text file.

- `odldb=<odl database path>`

  Specifies the path of the software on demand loading (SoftODL) database files. This is used to override the system default SoftODL database path, allowing users to use their own SoftODL database.

- `odlstyle=<odl style and size>`

  Specifies what SoftODL font style and size to use. The value is of the form <style>-<NxN> (for example, normal-32x32). If not specified, the system default SoftODL style and size is used.

- `line=<number of lines>`

  Specifies the number of lines per page. When used together with the `-w` command, this option can be used to control the font size and orientation of the output.

8.5 Japanese Printing Setup

When configuring Japanese printers, the only special consideration is whether ODL printing should, or should not be enabled. If enabled, the default ODL database path and style must be specified.

To configure any of the Japanese printers:

1. Use `lprsetup` to add one of the Japanese printers.

2. In response to the following question, answer "y" if you want ODL printing, and "n" if you do not.

   ```
   Do you want to enable ODL? [n]
   ```

3. The default ODL database and style will be used. To change the defaults, modify the `ya` capability after the following prompt:

   ```
   Enter symbol name:
   ```

   The prompt is displayed:

   ```
   Enter a new value for symbol 'ya'? ["plocale=ja_JP.sdeckanji"]
   ```

   Enter a command similar to the following:

   ```
   "plocale=jaJP.sdeckanji odldb=<path to the default ODL db>
   odlstyle=<default ODL style>"
   ```

   Do not forget the double quotation marks around the parameters, and replace the plocale with the value found in the previous prompt, and the default ODL path and style with the suitable one.

To print Japanese text files to print queues connected to dot matrix printers, use the `lpr` command. You can either set the LANG environment variable or use the `-A` option to denote the codeset of the text files. For example, enter the following command to print the file prepared in EUC codeset:

```
% lpr -A "flocale=ja_JP.eucJP" my.file1
```

You can override the default ODL path by including the `odldb` and `odlstyle` options in the `-A` option of their `lpr` command. For example, enter the following command to use the ODL database in `/usr/priv/odl`:

```
% lpr -A "odldb=/usr/priv odlstyle=normal-24x24" my.file2
```

9

Other Japanese Features

9.1 DOS Floppy Disk Access Utility – fdio

Floppy disk I/O (fdio) is a utility you use to read data from or write data to DOS-formatted floppy disks. The read and write functions are supported in both interactive and command modes. With this utility, you can exchange data on floppy disks between PCs and machines running the DIGITAL UNIX software.

This utility supports the following disk formats:

- 3.5 inches 2HD 1.44 Mbyte
- 3.5 inches 2DD 720 Kbyte
- 3.5 inches 2DD 640 Kbyte
- 3.5 inches MO

9.2 Japanese Kana–Kanji Conversion Dictionary Utilities

The Kana-Kanji conversion input methods presented in Chapter 7 are based on a conversion dictionary provided by the DIGITAL UNIX software. The conversion methods search through the dictionary for appropriate candidates for a Kana code or Romaji code and make them available for your selection. In addition to this system dictionary, you can create your own personal dictionary to define additional Kana-Kanji conversions to meet your preferences and requirements.

Two dictionary utilities, jdict2txt and txt2jdic, allow you to maintain and update your personal Kana-Kanji conversion dictionary. You can define Kana-Kanji conversions in text format and convert them into binary format, and vice versa. See Table 9-1.

Table 9-1: Dictionary Utilities

Utility	Conversion
jdic2txt	Dictionary to text conversion
txt2jdic	Text to dictionary conversion

9.3 Wnn with DIGITAL Enhancements

The Wnn utility is a front-end processor for inputting Japanese characters. This utility is public domain software that is very popular in Japan.

The Wnn utility consists of two components, uum and jserver, which convert an input string to a Kanji character string and then pass the string to an application process. This front-end processor design saves the programming effort in incorporating Japanese input methods into source codes.

The Wnn front-end processor is enhanced to access the Kana-Kanji conversion dictionary supplied with the DIGITAL UNIX software. Also, the Wnn library and the dictionary converter (dicconv) will be provided to convert your personal dictionary to the Wnn accessible format.

With the exception of the dictionary converter, the sources of Wnn are available in the kit.

9.4 Nemacs with DIGITAL Enhancements

Nemacs is the Japanese version of the GNU emacs public domain software.

Nemacs supports several major Japanese codesets, the egg input method, and the Kana-Kanji conversion input method.

This component handles the Japanese codesets by itself, and the processing is independent of the locale support provided by the DIGITAL UNIX software.

The Nemacs sources are available in the DIGITAL UNIX kit.

9.5 RWZ01/21 MO Disk Support

DIGITAL UNIX supports the RWZ01/21 Magneto Optical SCSI Disk. All required information is added to the `/etc/disktab` database when you install the DIGITAL UNIX software.

9.5.1 Japanese DECterm

Japanese DECterm is a terminal emulator that provides the Japanese processing functionality. This section describes the Japanese features that are specific to Japanese

DECterm. For details about the common internationalization features provided by DECterm, see *Writing Software for the International Market.*

The following sections describe:

- How to create a Japanese DECterm
- Japanese DECterm emulator features
- Japanese character input/output
- Other VT functionalities

9.5.2 Creating a Japanese DECterm

The terminal type that DECterm will emulate is sensitive to the session language.

To create a Japanese DECterm using the Session Manager, set the session language to one of the Japanese locales and then select DECterm from the Session Manager Applications menu.

Alternatively, you can use the `-xnllanguage` qualifier to specify the terminal type of the DECterm to create. For example, you can use `ja_JP` as the value for `-xnllanguage` to create a Japanese DECterm:

```
% /usr/bin/X11/dxterm -xnllanguage ja_JP
```

If you specify an unknown value for `-xnllanguage`, then ISO Latin-1 DECterm is assumed. If no Japanese font exists, the value defaults back to ISO Latin-1 DECterm.

The user interface language of Japanese DECterm always follows the terminal type. The language is independent of the language selection.

9.5.3 Customizing DECterm

You can apply to your Japanese DECterm windows any of the customization features that are currently applicable to the ISO Latin-1 DECterm window except for the customization of the NRCS character sets. In addition, some options are provided for customizing Japanese features.

Customized features can be saved and restored in the same way as in ISO Latin-1 DECterm.

9.5.4 Font Sizes

You can choose the 24x24, 16x18, or 14x14 Japanese fonts by selecting Big Font, Little Font, or Fine Font from the Window...selection box of the Options menu.

9.5.5 General Settings

Choosing General... from the Options menu allows you to change the general features, such as the terminal type, for the Japanese DECterm from the dialog box displayed. You can also select the following information from the dialog box:

- Japanese terminal ID
  - JDECterm ID (Default)
  - VT382 ID
  - VT286 ID
  - VT284 ID
  - VT282 ID
  - VT220J ID
  - VT102J ID
  - VT100J ID
  - VT80 ID
- 7-bit data display
  - JIS Roman Mode — Displays all 7-bit data as defined in JIS X 0201, that is, JIS Roman. For details about the JIS Roman character set, see Chapter 1.
  - ASCII mode — Displays all 7-bit data as defined in ASCII.
- 8-bit data display
  - Kanji — Displays all 8-bit data as defined in JIS X 0208. For details about the JIS X 0208 character set, see Chapter 1.
  - Katakana — Displays all 8-bit data as defined in JIS X 0201.
- DEC Kanji Variants
  - DEC Kanji 1978
  - DEC Kanji 1983

9.5.6 Copying Information

You can use the Edit menu options to copy information within or between DECterm windows. The Cut-and-Paste operation is enhanced to handle mixed ASCII and Japanese characters.

9.5.7 Default Character Sets

Japanese DECterm supports JIS X 0208 and all character sets supported by the ISO Latin-1 DECterm.

ISO Latin-1 DECterm uses ISO 8859-1 (Latin1) as the default character set. You can change the default by choosing other options from the General... pull down menu on the Options menu. For Japanese DECterm, the default character sets for 7-bit and 8-bit data are the JIS X 0201 and JIS X 0208 Japanese character sets respectively. You can override these default values and choose to use ASCII and JIS Katakana.

In general, Japanese DECterm cannot display mixed Latin-1 and Japanese characters. To achieve this function, you can output the data together with the designated character set escape sequences. For instance, the Japanese terminal driver is enhanced to display Japanese EUC data, which can embed both 8-bit Katakana characters and 8-bit Kanji characters, on Japanese DECterm. It can automatically insert appropriate escape sequences so that Japanese DECterm can correctly display mixed Katakana characters and Kanji characters.

9.5.8 Japanese Character Input

Japanese DECterm will emulate the Japanese VT terminal to support the Kana input mode. When Japanese DECterm is initialized, it is in the English input mode.

You can enter single-byte Kana characters in Japanese DECterm by invoking the Kana input mode as described in Chapter 7. Mixed ASCII/JIS Roman and Kana characters can be displayed properly in DECterm if you correctly select the display character sets.

In addition, an enhancement has been made in Japanese DECterm to send Kana characters encoded in Japanese EUC encoding to the host. You can enable the EUC encoding mode by selecting the General... menu from the Options menu and switch on the EUC Encoding toggle button. With this option on, the Kana characters that you enter are sent to the host together with the eight-bit control character SS2. For details about the Japanese EUC encoding, see Chapter 2.

Note

The Kana-Kanji conversion input methods for JIS X 0208 characters are not incorporated in Japanese DECterm. You must set up the terminal driver or customize applications according to their directions in order to use those input methods.

9.5.9 Special Characters in nroff

The nroff utility has been internationalized to format text of various languages. When a Japanese document is formatted using nroff, its contents are handled according to the following typical Japanese formatting rules:

- Text line wrapping

  Japanese text, unlike English text, does not use space characters to separate words or as break points for text line wrapping. A Japanese text line can be broken between any two consecutive Japanese characters, with the following exceptions:

  -- Some Japanese characters cannot be placed at the beginning of a text line. They are called no-first characters:

  、。，．・：；？！゛゜ヽゝゞヾ〃々ー’”
  ）〕］｝〉》」』】＞ぁぃぅぇぉっゃゅょゎ
  ァィゥェォッャュョヮヵヶ

  -- Some Japanese characters cannot be placed at the end of a text line. They are called no-last characters:

  ‘“（〔［｛〈《「『【＜・

  -- Some English characters are handled similarly according to this rule: No-first English characters:

  !) , . : ; > ?] }

  No-last English characters:

  (< [{

- Right justification

  To justify the right margin of a paragraph, nroff inserts space characters at proper break points. For English, space characters are usually inserted at word breaks. For Japanese, space characters are only inserted at the following places:

  • Before a can-space-before character if it is not placed at the beginning of a text line:

  ｛「『（

  • After a can-space-after character if it is not placed at the end of a text line:

  、。，．？！）｝」』