



JFP Reference Manual 5 : Standards, Environments, and Macros

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

Part No: 817-0648-10
December 2002

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.



021225@5115



Contents

Preface 5

JFP Reference Manual 5 : Standards, Environments, and Macros 11

Intro_jfp(5) 12

eucJP(5) 13

iconv_ja(5) 15

PCK(5) 26

Preface

Both novice users and those familiar with the SunOS operating system can use online man pages to obtain information about the system and its features. A man page is intended to answer concisely the question “What does it do?” The man pages in general comprise a reference manual. They are not intended to be a tutorial.

Overview

The following contains a brief description of each man page section and the information it references:

- Section 1 describes, in alphabetical order, commands available with the operating system.
- Section 1M describes, in alphabetical order, commands that are used chiefly for system maintenance and administration purposes.
- Section 2 describes all of the system calls. Most of these calls have one or more error returns. An error condition is indicated by an otherwise impossible returned value.
- Section 3 describes functions found in various libraries, other than those functions that directly invoke UNIX system primitives, which are described in Section 2.
- Section 4 outlines the formats of various files. The C structure declarations for the file formats are given where applicable.
- Section 5 contains miscellaneous documentation such as character-set tables.
- Section 6 contains available games and demos.
- Section 7 describes various special files that refer to specific hardware peripherals and device drivers. STREAMS software drivers, modules and the STREAMS-generic set of system calls are also described.

- Section 9 provides reference information needed to write device drivers in the kernel environment. It describes two device driver interface specifications: the Device Driver Interface (DDI) and the Driver/Kernel Interface (DKI).
- Section 9E describes the DDI/DKI, DDI-only, and DKI-only entry-point routines a developer can include in a device driver.
- Section 9F describes the kernel functions available for use by device drivers.
- Section 9S describes the data structures used by drivers to share information between the driver and the kernel.

Below is a generic format for man pages. The man pages of each manual section generally follow this order, but include only needed headings. For example, if there are no bugs to report, there is no BUGS section. See the `intro` pages for more information and detail about each section, and `man(1)` for more information about man pages in general.

NAME	This section gives the names of the commands or functions documented, followed by a brief description of what they do.								
SYNOPSIS	<p>This section shows the syntax of commands or functions. When a command or file does not exist in the standard path, its full path name is shown. Options and arguments are alphabetized, with single letter arguments first, and options with arguments next, unless a different argument order is required.</p> <p>The following special characters are used in this section:</p> <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 10px;">[]</td> <td>Brackets. The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.</td> </tr> <tr> <td style="padding-right: 10px;">. . .</td> <td>Ellipses. Several values can be provided for the previous argument, or the previous argument can be specified multiple times, for example, "filename . . .".</td> </tr> <tr> <td style="padding-right: 10px;"> </td> <td>Separator. Only one of the arguments separated by this character can be specified at a time.</td> </tr> <tr> <td style="padding-right: 10px;">{ }</td> <td>Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.</td> </tr> </table>	[]	Brackets. The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.	. . .	Ellipses. Several values can be provided for the previous argument, or the previous argument can be specified multiple times, for example, "filename . . .".		Separator. Only one of the arguments separated by this character can be specified at a time.	{ }	Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.
[]	Brackets. The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.								
. . .	Ellipses. Several values can be provided for the previous argument, or the previous argument can be specified multiple times, for example, "filename . . .".								
	Separator. Only one of the arguments separated by this character can be specified at a time.								
{ }	Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.								

PROTOCOL	This section occurs only in subsection 3R to indicate the protocol description file.
DESCRIPTION	This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. It does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, and functions are described under USAGE.
IOCTL	This section appears on pages in Section 7 only. Only the device class that supplies appropriate parameters to the <code>ioctl(2)</code> system call is called <code>ioctl</code> and generates its own heading. <code>ioctl</code> calls for a specific device are listed alphabetically (on the man page for that specific device). <code>ioctl</code> calls are used for a particular class of devices all of which have an <code>io</code> ending, such as <code>mtio(7I)</code> .
OPTIONS	This section lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.
OPERANDS	This section lists the command operands and describes how they affect the actions of the command.
OUTPUT	This section describes the output – standard output, standard error, or output files – generated by the command.
RETURN VALUES	If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a function can return only constant values, such as 0 or -1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions declared void do not return values, so they are not discussed in RETURN VALUES.
ERRORS	On failure, most functions place an error code in the global variable <code>errno</code> indicating why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than

	one condition can cause the same error, each condition is described in a separate paragraph under the error code.
USAGE	This section lists special rules, features, and commands that require in-depth explanations. The subsections listed here are used to explain built-in functionality: Commands Modifiers Variables Expressions Input Grammar
EXAMPLES	This section provides examples of usage or of how to use a command or function. Wherever possible a complete example including command-line entry and machine response is shown. Whenever an example is given, the prompt is shown as <code>example%</code> , or if the user must be superuser, <code>example#</code> . Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS, and USAGE sections.
ENVIRONMENT VARIABLES	This section lists any environment variables that the command or function affects, followed by a brief description of the effect.
EXIT STATUS	This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion, and values other than zero for various error conditions.
FILES	This section lists all file names referred to by the man page, files of interest, and files created or required by commands. Each is followed by a descriptive summary or explanation.
ATTRIBUTES	This section lists characteristics of commands, utilities, and device drivers by defining the attribute type and its corresponding value. See <code>attributes(5)</code> for more information.
SEE ALSO	This section lists references to other man pages, in-house documentation, and outside publications.

DIAGNOSTICS	This section lists diagnostic messages with a brief explanation of the condition causing the error.
WARNINGS	This section lists warnings about special conditions which could seriously affect your working conditions. This is not a list of diagnostics.
NOTES	This section lists additional information that does not belong anywhere else on the page. It takes the form of an aside to the user, covering points of special interest. Critical information is never covered here.
BUGS	This section describes known bugs and, wherever possible, suggests workarounds.

JFP Reference Manual 5 : Standards, Environments, and Macros

Intro_jfp(5)

NAME	Intro_jfp, intro_jfp – introduction to miscellany	
DESCRIPTION	Among the topics presented in this section are:	
	Headers	The header (.h) files <code>fcntl</code> , <code>floatingpoint</code> , <code>math</code> , <code>langinfo</code> , <code>nl_types</code> , <code>siginfo</code> , <code>signal</code> , <code>stat</code> , <code>stdarg</code> , <code>types</code> , <code>ucontext</code> , <code>values</code> , <code>varargs</code> , and <code>wait</code> (on the <code>wstat</code> page) are described.
	Environments	The user environment (<code>environ</code>), the subset of the user environment that depends on language and cultural conventions (<code>locale</code>), the large file compilation environment (<code>lfcompile</code>), and the transitional compilation environment (<code>lfcompile64</code>) are described.
	Macros	The macros to format Reference Manual pages (<code>man</code> and <code>mansun</code>) as well as other text format macros (<code>me</code> , <code>mm</code> , and <code>ms</code>) are described.
	Characters	Tables of character sets (<code>ascii</code> , <code>charmap</code> , <code>eqnchar</code> , and <code>iconv</code>), file format notation (<code>formats</code>), file name pattern matching (<code>fnmatch</code>), and regular expressions (<code>regex</code> and <code>regexp</code>) are presented.
	FNS	Topics concerning the Federated Naming Service (<code>fns</code> , <code>fns_initial_context</code> , <code>fns_policies</code> , and <code>fns_references</code>) are discussed.
	Standards	The POSIX (IEEE) Standards and the X/Open Specifications are described on the <code>standards</code> page.
LIST	Name	Description
	Intro_jfp(5)	introduction to miscellany
	eucJP(5)	map between Japanese EUC and character set
	eucjp(5)	See <code>eucJP(5)</code>
	iconv_ja(5)	code set conversions in ja locale
	PCK(5)	map between PCK and character set
	pck(5)	See <code>PCK(5)</code>

NAME eucJP, eucjp – map between Japanese EUC and character set

DESCRIPTION In SunOS and JFP, Japanese EUC (Extended UNIX code) is used as character code system expressing multi-byte languages including Japanese character in the "ja" locale. This manual page shows map between Japanese EUC and character set.

The following is a map table for Japanese EUC and character set (SS2 stands for 0x8e, and SS3 stands for 0x8f).

TABLE 1

Japanese EUC	character set
0x00 – 0x19	JIS X 0201–1976 function character set
0x20	JIS X 0201–1976 space character
0x21 – 0x7e	JIS X 0201–1976 figure character set for roman character
0x7f	JIS X 0201–1976 erase character
0x80 – 0x9f	ISO 6429 C1 control character (except for 0x8e, 0x8f)
SS2 0xa1 – 0xdf	JIS X 0201–1976 figure character set for katakana (except for an area of undefined character E/0 – F/14)
SS3 0xa1a1 – 0xa1fe	JIS X 0212–1990 (1 ku 1 ten – 1 ku 94 ten)
SS3 0xa2a1 – 0xa2fe	JIS X 0212–1990 (2 ku 1 ten – 2 ku 94 ten)
:	:
:	:
SS3 0xf3a1 – 0xf3fe	JIS X 0212–1990 (83 ku 1 ten – 83 ku 94 ten)
SS3 0xf4a1 – 0xf4fe	JIS X 0212–1990 (84 ku 1 ten – 84 ku 94 ten)
0xa1a1 – 0xa1fe	JIS X 0208–1990 (1 ku 1 ten – 1 ku 94 ten)
0xa2a1 – 0xa2fe	JIS X 0208–1990 (2 ku 1 ten – 2 ku 94 ten)
:	:
:	:
0xf3a1 – 0xf3fe	JIS X 0208–1990 (83 ku 1 ten – 83 ku 94 ten)
0xf4a1 – 0xf4fe	JIS X 0208–1990 (84 ku 1 ten – 84 ku 94 ten)

Each character set corresponds with Japanese EUC code set numbers as follows.

eucJP(5)

TABLE 2

character set (coding character set)	Japanese EUC extended code set no.
JIS X 0201 figure character set for roman character	0
JIS X 0208-1990	1
JIS X 0201 figure character set for katakana	2
JIS X 0212-1990	3

NOTES In Japanese EUC area below has special meaning.

TABLE 3

Japanese EUC	area of character set	meaning
SS3 0xf3a1 - 0xf4fe	JIS X 0212-1990 83 ku - 84 ku	Vender Defined Character (IBM extension character not included in JIS X 0212-1990)
SS3 0xf5a1 - 0xfefe		User Defined Character 11 ku - 20 ku
0xada1 - 0xadfe	JIS X 0208-1990 13 ku	Vender Defined Character (special symbols)
0xf5a1 - 0xfefe		User Defined Character 1 ku - 10 ku

SEE ALSO PCK(5)

NAME iconv_ja – code set conversions in ja locale

DESCRIPTION The following code set conversions are supported:

Code Set Conversions Supported	
Source Code	Target Code
euJP	PCK
euJP	ISO-2022-JP
euJP	ISO-2022-JP.RFC1468
euJP	JIS7
euJP	SJIS
euJP	UTF-8
euJP	UTF-8-Java
euJP	jis
euJP	ibmj
euJP	ibmj-EBCDIK
SJIS	euJP
SJIS	ISO-2022-JP
SJIS	UTF-8
SJIS	jis
SJIS	ibmj
PCK	euJP
PCK	UTF-8
PCK	UTF-8-Java
PCK	ISO-2022-JP
PCK	ISO-2022-JP.RFC1468
PCK	jis
PCK	ibmj
PCK	ibmj-EBCDIK
ISO-2022-JP	euJP
ISO-2022-JP	PCK

iconv_ja(5)

ISO-2022-JP	SJIS
ISO-2022-JP	UTF-8
UTF-8	euJP
UTF-8	SJIS
UTF-8	PCK
UTF-8	ISO-2022-JP
UTF-8	ISO-2022-JP.RFC1468
UTF-8-Java	euJP
UTF-8-Java	PCK
JIS7	euJP
jis	euJP
jis	PCK
jis	SJIS
ibmj	euJP
ibmj	PCK
ibmj	SJIS
ibmj-EBCDIK	euJP
ibmj-EBCDIK	PCK

Code Set Conversions Supported	
Source Code	Target Code
euJP	ibm930
euJP	ibm931
euJP	ibm939
euJP	ibm5026
euJP	ibm5035
euJP	FujitsuJEF-ascii-code
euJP	FujitsuJEF-kana-code
euJP	FujitsuJEF-ascii-face
euJP	FujitsuJEF-kana-face

eucJP	HitachiKEIS83
eucJP	HitachiKEIS90
eucJP	NECJIPS
PCK	ibm930
PCK	ibm931
PCK	ibm939
PCK	ibm5026
PCK	ibm5035
PCK	FujitsuJEF-ascii-code
PCK	FujitsuJEF-kana-code
PCK	FujitsuJEF-ascii-face
PCK	FujitsuJEF-kana-face
PCK	HitachiKEIS83
PCK	HitachiKEIS90
PCK	NECJIPS
UTF-8	ibm930
UTF-8	ibm931
UTF-8	ibm939
UTF-8	ibm5026
UTF-8	ibm5035
UTF-8	ms932
UTF-8	UTF-8-ms932
UTF-8	FujitsuJEF-ascii-code
UTF-8	FujitsuJEF-kana-code
UTF-8	FujitsuJEF-ascii-face
UTF-8	FujitsuJEF-kana-face
UTF-8	HitachiKEIS83
UTF-8	HitachiKEIS90
UTF-8	NECJIPS
UTF-8-ms932	UTF-8

iconv_ja(5)

ibm930	eucJP
ibm930	PCK
ibm930	UTF-8
ibm931	eucJP
ibm931	PCK
ibm931	UTF-8
ibm939	eucJP
ibm939	PCK
ibm939	UTF-8
ibm5026	eucJP
ibm5026	PCK
ibm5026	UTF-8
ibm5035	eucJP
ibm5035	PCK
ibm5035	UTF-8
FujitsuJEF-ascii-code	eucJP
FujitsuJEF-ascii-code	PCK
FujitsuJEF-ascii-code	UTF-8
FujitsuJEF-kana-code	eucJP
FujitsuJEF-kana-code	PCK
FujitsuJEF-kana-code	UTF-8
FujitsuJEF-ascii-face	eucJP
FujitsuJEF-ascii-face	PCK
FujitsuJEF-ascii-face	UTF-8

Code Set Conversions Supported	
Source Code	Target Code
FujitsuJEF-kana-face	eucJP
FujitsuJEF-kana-face	PCK
FujitsuJEF-kana-face	UTF-8

HitachiKEIS83	eucJP
HitachiKEIS83	PCK
HitachiKEIS83	UTF-8
HitachiKEIS90	eucJP
HitachiKEIS90	PCK
HitachiKEIS90	UTF-8
NECJIPS	eucJP
NECJIPS	PCK
NECJIPS	UTF-8
ms932	UTF-8

The descriptions of each code sets in the above table are as follows:

Description of Supported Code Sets	
Codeset	Description
eucJP	Japanese EUC
PCK	PC kanji
SJIS	the same as PC kanji (eol in future)
ISO-2022-JP	Coded representation of the character sets ISO 646 IRV or JIS X 0201, JIS X 0208, and JIS X 0212 according to <i>UI/OSF Application Platform Profile for Japanese Environment Version 1.1 item 7.1</i> using the designation sequence to G0 specified by ISO 2022

Description of Supported Code Sets	
Codeset	Description
JIS7	same as ISO-2022-JP
ISO-2022-JP.RFC1468	Coded representation of the character sets ISO 646 IRV or JIS X 0201-1976 (except for figure character set for katakana), and JIS X 0208-1983 according to RFC1468 (Request for Comments: 1468 Japanese Character Encoding for Internet Messages) using the designation sequence to G0 specified by ISO 2022
jis	JIS 7bit code used in JLE, JFP 2.4 and the preceding releases

iconv_ja(5)

ibmj	IBM Kanji code
ibmj-EBCDIK	Maps single-byte code set (SBCS) of IBM host code to the character set that is called the EBCDIK code set in general. The character code set includes the IBM code page 290 and three more characters "' (0x79), '{' (0xc0), and '}' (0xd0). Japanese katakana characters are included, but lowercase alphabet letters are not. In case of double-byte code set (DBCS), the description is the same as the code set "ibmj."
UTF-8	UNI CODE
UTF-8-Java	UNI CODE implemented in Java

Description of Supported Code Sets	
Codeset	Description
ibm930	IBM CCSID 930: SBSC code page 290 (extended), character set 1172, DBCS code page 300, character set 1001 4370 user defined characters
ibm931	IBM CCSID 931: SBSC code page 37, character set 101, DBCS code page 300, character set 1001 4370 user defined characters
ibm939	IBM CCSID 930: SBSC code page 1027, character set 1172, DBCS code page 300, character set 1001 4370 user defined characters
ibm5026	IBM CCSID 5026: same as ibm930, except this code set supports 1880 user defined characters
ibm5035	IBM CCSID 5035: same as ibm939, except this code set supports 1880 user defined characters
FujitsuJEF-ascii-code	Fujitsu JEF code which uses EBCDIC(ASCII) as single byte character set. Charatcters differently handled between JIS C 6226:1978 and JIS X 0208:1983 are mapped based on code value.
FujitsuJEF-ascii-face	Fujitsu JEF code which uses EBCDIC(ASCII) as single byte character set. Charatcters differently handled between JIS C 6226:1978 and JIS X 0208:1983 are mapped based on character face.
FujitsuJEF-kana-code	Fujitsu JEF code which uses EBCDIC(Kana) as single byte character set. Caratcters differently handled between JIS C 6226:1978 and JIS X 0208:1983 are mapped based on code value.

FujitsuJEF-kana-face	Fujitsu JEF code which uses EBCDIC(Kana) as single byte character set. Characters differently handled between JIS C 6226:1978 and JIS X 0208:1983 are mapped based on character face.
----------------------	---

Description of Supported Code Sets	
Codeset	Description
HitachiKEIS8	Hitachi KEIS83
HitachiKEIS90	Hitachi KEIS83
NECJIPS	NEC JIPS code
ms932	Shift JIS codeset which is supported by Windows NT 3.51. Conversion between this codeset and UTF-8 is done in the same way Windows NT 3.51 does.
UTF-8-ms932	UTF-8 encoded Unicode which was converted from ms932

Conversions are performed as described below. For all conversions, if the source code set includes characters not included in the target code set, conversion and output for all such characters will be done using a substitute character.

eucJP to PCK (SJIS) and PCK (SJIS) to eucJP

Conversion between eucJP and PCK (SJIS) can be used to convert JIS X 0201, JIS X 0208, JIS X 0212, and user-defined and vendor-defined characters based on *TOG Japanese Vendors Council (TOG/JVC) Recommended Code Set Conversion Specification* between Japanese EUC and Shift-JIS. If input data which does not belong to the source code set is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

eucJP to ISO-2022-JP(JIS7) and ISO-2022-JP(JIS7) to eucJP

Conversion between eucJP and ISO-2022-JP(JIS7) can be used to convert JIS X 0201, JIS X 0208 and JIS X 0212. If input data which does not belong to the source code set is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

eucJP to ISO-2022-JP.RFC1468

Conversion from eucJP to ISO-2022-JP.RFC1468 can be used to convert JIS X 0201 (except for figure character set for katakana) and JIS X 0208. If JIS X 0201 (figure character set for katakana), JIS X 0212, a user-defined, or a vendor-defined character is encountered among input data, it will be replaced with the substitute character ' ? ' (0x3f). If input data which does not belong to these code sets is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

eucJP to jis and jis to eucJP

Conversion between eucJP and jis is provided for the compatibility with `ujtojis7()` and `jis7touj()` libraries, and `euctojis` and `jistoeuc` utilities. It is extended to handle JIS X 0212. See `jistoeuc(1)`.

eucJP to UTF-8 and UTF-8 to eucJP

Conversion between eucJP and UTF-8 can be used to convert JIS X 0201, JIS X 0208, JIS X 0212, a user-defined, and a vendor-defined character. If input data which does not have the corresponding character in the target code set is encountered, it will be replaced with the substitute character (eucJP: ' ? ' (0x3f), UTF-8: U+FFFD (0xefbfbf)). If input data which does not belong to these code sets is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

eucJP to UTF-8-Java and UTF-8-Java to eucJP

Conversion between eucJP and UTF-8-Java can be used to convert JIS X 0201, JIS X 0208, and JIS X 0212. If a user-defined or vendor-defined character is encountered among input data, it will be replaced with the substitute character (eucJP: ' ? ' (0x3f), UTF-8: U+FFFD (0xefbfbf)). If input data which does not belong to these code sets is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

eucJP to ibmj and ibmj to eucJP

Conversion between eucJP and ibmj is based on the IBM Kanji codebook (4th edition - September 1987), JIS X 0201, and JIS X 0208-1983. If you convert eucJP to ibmj, JISX 0201 and JIS X 0201 are all converted to substitute character.

eucJP to ibmj-EBCDIK and ibmj-EBCDIK to eucJP

Conversion between eucJP and ibmj-EBCDIK is based on the IBM Kanji codebook (4th edition - September 1987), JIS X 0201, and JIS X 0208-1983. If you convert eucJP to ibmj-EBCDIK, JISX 0201 and JIS X 0201 that have not correspondence characters with ibmj-EBCDIK are all converted to substitute character.

PCK (SJIS) to ISO-2022-JP and ISO-2022-JP to PCK (SJIS)

Conversion between PCK (SJIS) and ISO-2022-JP can be used to convert JIS X 0201, JIS X 0208, JIS X 0212, and user-defined and vendor-defined characters based on *TOG Japanese Vendors Council (TOG/JVC) Recommended Code Set Conversion Specification between Japanese EUC and Shift-JIS*. If input data which does not belong to the source code set is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

PCK (SJIS) to ISO-2022-JP.RFC1468

Conversion from PCK (SJIS) to ISO-2022-JP.RFC1468 can be used to convert JIS X 0201 (except for figure character set for katakana) and JIS X 0208. If JIS X 0201 (figure character set for katakana), a user-defined, or a vendor-defined character is encountered among input data, it will be replaced with the substitute character ' ? ' (0x3f). If input data which does not belong to these code sets is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

PCK (SJIS) to UTF-8 and UTF-8 to PCK (SJIS)

Conversion between PCK (SJIS) and UTF-8 can be used to convert JIS X 0201, JIS X 0208, a user-defined, and a vendor-defined character. If input data which does not have the corresponding character in the target code set is encountered, it will be replaced with the substitute character (PCK: ' ? ' (0x3f), UTF-8: U+FFFD (0xefbfbf)). If input data which does not belong to these code sets is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

PCK (SJIS) to UTF-8-Java and UTF-8-Java to PCK (SJIS)

Conversion between PCK (SJIS) and UTF-8-Java can be used to convert JIS X 0201 and JIS X 0208. If a user-defined or vendor-defined character is encountered among input data, it will be replaced with the substitute character (PCK: ' ? ' (0x3f), UTF-8: U+FFFD (0xefbfbf)). If input data which does not belong to these code sets is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

PCK (SJIS) to jis and jis to PCK (SJIS)

Conversion between PCK (SJIS) and jis is provided for the compatibility with `sjtojis7()` and `jis7tosj()` libraries, and `sjtojis` `jistosj` utilities. It is extended based on *TOG Japanese Vendors Council (TOG/JVC) Recommended Code Set Conversion Specification between Japanese EUC and Shift-JIS*. See `jistosj(1)`.

PCK (SJIS) to ibmj and ibmj to PCK (SJIS)

Conversion between PCK (SJIS) and ibmj is based on the IBM Kanji codebook (4th edition - September 1987), JIS X 0201, and JIS X 0208-1983. If you convert PCK (SJIS) to ibmj, all characters converted to JIS X 0212 by kana characters (0xa1 to 0xdf) and *TOG Japanese Vendors Council (TOG/JVC) Recommended Code Set Conversion Specification between Japanese EUC and Shift-JIS* are all converted to substitute character.

PCK to ibmj-EBCDIK and ibmj-EBCDIK to PCK

Conversion between PCK and ibmj-EBCDIK is based on the IBM Kanji codebook (4th edition - September 1987), JIS X 0201, and JIS X 0208-1983. If you convert PCK to ibmj-EBCDIK, all characters converted to JIS X 0212 by JIS X 0212 and *TOG Japanese Vendors Council (TOG/JVC) Recommended Code Set Conversion Specification between Japanese EUC and Shift-JIS* are all converted to substitute character.

ISO-2022-JP to UTF-8 and UTF-8 to ISO-2022-JP

Conversion between ISO-2022-JP and UTF-8 can be used to convert JIS X 0201, JIS X 0208, JIS X 0212, and user-defined and vendor-defined characters. If input data which does not have the corresponding character in the target code set is encountered, it will be replaced with the substitute character (ISO-2022-JP: ' ? ' (0x3f), UTF-8: U+FFFD (0xefbfbf)). If input data which does not belong to these code sets is encountered, `iconv(3C)` will return EILSEQ for `errno`. `iconv(1)` stops at the last point of successful conversion.

UTF-8 to ISO-2022-JP.RFC1468

Conversion from UTF-8 to ISO-2022-JP.RFC1468 can be used to convert JIS X 0201 (except for figure character set for katakana) and JIS X 0208. If JIS X 0201 (figure character set for katakana), JIS X 0212, a user-defined, or a vendor-defined

iconv_ja(5)

character is encountered among input data, it will be replaced with the substitute character '?' (0x3f). If input data which does not belong to these code sets is encountered, `iconv(3C)` will return `EILSEQ` for `errno`. `iconv(1)` stops at the last point of successful conversion.

eucJP, PCK, UTF-8 to ibm930, ibm931, ibm939, ibm5026, ibm5035

Conversion from eucJP, PCK, or UTF-8 to ibm930, ibm931, ibm939, ibm5026, ibm5035 can be used to convert JIS X 0201, JIS X 0208, JIS X 0212, IBM extension characters, and user defined character. Input data which does not have corresponding character in the target code set is replaced with the substitute character. Since ibm931 does not support Kana characters in its single byte code set (SBCS), JIS X 0201 Kana characters are replaced with substitute characters in conversion to ibm931.

ibm930, ibm931, ibm939, ibm5026, or ibm5035 to eucJP, PCK, or UTF-8

Conversion from ibm930, ibm931, ibm939, ibm5026, or ibm5035 to eucJP, PCK, or UTF-8 can be used to convert SBCS/DBCS characters defined in input code set. Input data which does not have corresponding character in the target code set is replaced with the substitute character.

FujitsuJEF-ascii-code or FujitsuJEF-kana-code to eucJP, PCK, or UTF-8: eucJP, PCK, or UTF-8 to FujitsuJEF-ascii-code or FujitsuJEF-kana-code

Conversion from FujitsuJEF-ascii-code or FujitsuJEF-kana-code to eucJP, PCK, or UTF-8 and eucJP, PCK, or UTF-8 to FujitsuJEF-ascii-code or FujitsuJEF-kana-code can be used to convert JIS X 0201, JIS X 0208, and JIS X 0212 characters defined in input code set. Input data which does not have corresponding character in the target code set is replaced with the substitute character. Characters handled differently between JIS C 6226:1978 and JIS X 0208:1983 are converted based on code value. In conversion to FujitsuJEF-kana-code, JIS X 0201 Katakana characters are replaced with the substitute character.

FujitsuJEF-ascii-face or FujitsuJEF-kana-face to eucJP, PCK, or UTF-8: eucJP, PCK, or UTF-8 to FujitsuJEF-ascii-face or FujitsuJEF-kana-face

Characters handled differently between JIS C 6226:1978 and JIS X 0208:1983 are converted based on character face. Other behavior is same as that of FujitsuJEF-ascii-code or FujitsuJEF-kana-code.

HitachiKEIS83 to eucJP, PCK, or UTF-8: eucJP, PCK, or UTF-8 to HitachiKEIS83

Conversion from HitachiKEIS83 to eucJP, PCK, or UTF-8 and from eucJP, PCK, or UTF-8 to HitachiKEIS83 can be used to convert JIS X 0201, JIS X 0208, and JIS X 0212 characters. Input data which does not have corresponding character in the target code set is replaced with the substitute character.

HitachiKEIS90 to eucJP, PCK, or UTF-8: eucJP, PCK, or UTF-8 to HitachiKEIS90

Behavior of conversion from HitachiKEIS90 to eucJP, PCK, or UTF-8 and from eucJP, PCK, or UTF-8 to HitachiKEIS90 is same as that of KEIS83 except that this conversion uses KEIS90 instead of KEIS83.

NECJIPS to eucJP, PCK, or UTF-8: eucJP, PCK, or UTF-8 to NECJIPS

Conversion from NECJIPS to eucJP, PCK, or UTF-8 and eucJP, PCK, or UTF-8 to NECJIPS can be used to convert JIS X 0201, JIS X 0208, and JIS X 0212 characters.

iconv_ja(5)

Input data which does not have corresponding character in the target code set is replaced with the substitute character. EBCDIC Katakana (EK) code is used for single byte character set.

ms932 to UTF-8 and UTF-8 to ms932

Conversion between ms932 and UTF-8 is done using same way of mapping characters between the two codesets as Windows NT 3.51 does.

UTF-8 to UTF-8-ms932 and UTF-8-ms932 to UTF-8

This converts between "UTF-8" and "UTF-8-ms932", which are UTF-8 encoded Unicode converted from PCK, and that converted from ms932.

SEE ALSO iconv(1), jistoeuc(1), jistosj(1), iconv(3C), iconv(5), iconv_unicode(5)

PCK(5)

NAME PCK, pck – map between PCK and character set

DESCRIPTION PCK stands for PC Kanji code. It is also known as Shift-JIS (MS kanji) code. This manual page shows map between PCK and character set.

PCK allocates the first byte of each kanji code which corresponds to JIS X0208–1990 and so on to the code other than code range of 0x00–0x7f and 0xa1–0xdf. The following is a map table for PCK and character set.

PCK	character set
0x00 – 0x19	JIS X 0201–1976 function character set
0x20	JIS X 0201–1976 space character
0x21 – 0x7e	JIS X 0201–1976 figure character set for roman character
0x7f	JIS X 0201–1976 erase character
0xa1 – 0xdf	JIS X 0201–1976 figure character set for katakana (except for an area of undefined character E/0 – F/14)
0x8140 – 0x817e	JIS X 0208–1990 (1 ku 1 ten – 1 ku 63 ten)
0x8180 – 0x819e	JIS X 0208–1990 (1 ku 64 ten – 1 ku 94 ten)
0x819f – 0x81fc	JIS X 0208–1990 (2 ku 1 ten – 2 ku 94 ten)
:	:
:	:
0x9f9f – 0x9ffc	JIS X 0208–1990 (62 ku 1 ten – 62 ku 94 ten)
0xe040 – 0xe07e	JIS X 0208–1990 (63 ku 1 ten – 63 ku 63 ten)
:	:
:	:
0xea40 – 0xea7e	JIS X 0208–1990 (83 ku 1 ten – 83 ku 63 ten)
0xea80 – 0xea9e	JIS X 0208–1990 (83 ku 64 ten – 83 ku 94 ten)
0xea9f – 0xeafc	JIS X 0208–1990 (84 ku 1 ten – 84 ku 94 ten)

The differences from character set of Japanese EUC are,

- Character set of Japanese EUC codeset 3 (JIS X 0212–1990) is not assigned except a part of Vender Defined Character (IBM extension character) (see NOTES).
- Character set which corresponds to Japanese EUC C1 control code is not assigned.

NOTES In PCK area below has special meaning.

PCK	area of character set	meaning
0x8740 – 0x879e	JIS X 0208–1990 13 ku	Vender Defined Character (special symbols)
0xeb40 – 0xecfc		Invalid
0xed40 – 0xeffc		Vender Defined Character (IBM extension character NEC selected)
0xf040 – 0xf9fc		User Defined Character 1 ku – 20 ku
0xfa40 – 0xfcfc		Vender Defined Character (IBM extension character)

SEE ALSO eucJP(5)

PCK(5)