



Korean Solaris User's Guide

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

Part No: 817-2522-10
January 2005

Copyright 2005 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, SunSoft, SunDocs, SunExpress, SunOS is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. 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. Netscape Navigator is a trademark or registered trademark of Netscape Communications Corporation in the United States and other countries. PostScript is a trademark or registered trademark of Adobe Systems, Incorporated, which may be registered in certain jurisdictions.

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.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

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.

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, SunSoft, SunDocs, SunExpress, SunOS sont des marques déposées ou enregistrées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. Netscape Navigator est une marque de Netscape Communications Corporation aux Etats-Unis et dans d'autres pays. PostScript est une marque de fabrique d'Adobe Systems, Incorporated, laquelle pourrait être déposée dans certaines juridictions.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



040805@9495



Contents

Preface 11

Part I Features Overview 15

- 1 Overview of Features 17**
 - New Localized Features 17
 - Language Support 18
 - Locale Attributes 18
 - Locale Categories 20
 - Locale Modifier 21

Part II End-User Information 23

- 2 Introduction to the Korean Solaris Software 25**
 - Design of the Korean Solaris Software 25
 - Korean Locales 25
 - ▼ How to Set the Locale at Login 26
- 3 CDE 27**
 - Korean Graphical User Interface 27
 - Korean Input/Output 27
 - Localized CDE Desktop 28

4	Entering Korean Text	31
	Character Input Overview	31
	Supported Character Sets	32
	Input Window Areas	33
	Basic Functions for Korean Input	34
	Starting to Input Hangeul	34
	▼ How to Switch to the Hangeul Input Mode	34
	▼ How to Enter Hangeul Characters	35
	▼ How to Backspace and Delete Korean Characters	36
	▼ How to Commit Hangeul Input	38
	▼ How to Return to the ASCII/English Input Mode	39
	Switching Keyboard Layouts	40
	▼ How to Switch Keyboard Layouts	40
	Using Hangeul/Hanja Conversion to Enter Hanja	41
	▼ How to Use the Hanja Lookup Mode	42
	▼ How to Convert Hanja to Hangeul in Character Step Mode	44
	▼ How to Convert Hanja to Hangeul Using Word Conversion Modes	46
	Entering Special Symbol Characters	47
	▼ How to Enter Special Symbol Characters	47
	Entering Hangeul Characters Using Hex Input Mode	51
	▼ How to Use the Hex Input Mode	52
	Switching Between the Half_Width and the Full_Width Character Modes	55
	▼ How to Switch Between Half_Width and Full_Width Character Modes	55
	Auxiliary Windows for the Korean Input Method	56
	Palette Auxiliary Window	56
	Input Method Options	57
	Virtual Keyboard Auxiliary Window	59
	Lookup Table Auxiliary Window	60
	Input Method Help	61
5	Hanja Tool	63
	Hanja Tool Features	63
	Using Hanja Tool	64
	▼ How to Launch the Hanja Tool	64
	▼ How to Find the Hanja for a Hangeul word	65
	▼ How to Add Hanja to the Hangeul-Hanja Dictionary	67
	▼ How to Modify Hanja in the Hangeul-Hanja Dictionary	70
	▼ How to Delete Hanja from the Hangeul-Hanja Dictionary	71

6	Localized Applications	73
	Using the <code>mailx</code> Utility with Korean Characters	73
	Using <code>talk</code> with Korean Characters	74
	The <code>xtobdf</code> Utility	74
	The <code>Sdtconvtool</code> Utility	74
	▼ How to Use the <code>Sdtconvtool</code> Utility	74
	The <code>iconv</code> Utility	75
7	Korean Solaris Printing	79
	Line Printer Support	79
	Printing an EUC (Wansung) File to a Printer that Does Not Support EUC	79
	Printing a <code>ko_KR.UTF-8</code> File to Printers that Do Not Support UTF-8	81
	Laser Printer Support	81
	Using the <code>mp</code> Utility	81
Part III	System Administrator Information	83
8	Configuring a Korean Solaris System	85
	Setting the Default Locale	85
	▼ How to Set the Default Locale	85
	Using Aliases to Set the Locale	86
	▼ How to Set Locale Aliases for the Korn Shell	86
	How to Set Locale Aliases for the C Shell	86
9	Managing the <code>htt</code> Input Method Server	87
	Input Method Server Basic Properties	87
	Three <code>htt</code> Processes	88
	<code>htt</code> Application Display	88
	How and When <code>htt</code> is Started	89
	Using the <code>iiim</code> Server in the <code>ko_KR.UTF-8</code> Locales	89
10	Fonts	91
	Display PostScript System (DPS)	91
	Using Korean PostScript Fonts and DPS Facilities	91
	Creating Composite Korean Fonts	94
	Using Korean Fonts in DPS Programming	94

	BDF and PCF Formats	95
	▼ How to Convert BDF Fonts to PCF Format	95
	▼ How to Install and Check an Edited Font	95
	Changing Font Directories	96
	Bitmap and TrueType Fonts	96
11	Korean Printing Facilities	99
	Printing Korean Output to a Line Printer	99
	Using the mp Utility	100
	Using mp as an Xprt (X Print Server) Client	101
12	TTY Environment and Support	103
	TTY STREAMS	103
	Traditional STREAMS	104
	Korean Solaris STREAMS	104
	TTY Utilities	105
	EUC ioctl Features	105
	The termcap database	105
	The terminfo database	106
	TTY Commands	106
	setterm Command	106
	/bin/stty Command	107
	TTY Setup Examples	107
	Configuring STREAMS for Korean Solaris Software	107
	Configuring STREAMS for the ko.UTF-8 Locale	109
	Terminal Support	110
	Installing a Terminal	110
	▼ How to Install a Terminal to Support KS C 5601	111
	▼ How to Set Up a ttya Port from the Command Line	112
	▼ How to Verify TTY Set Up	113
	▼ How to Use Packed Code and Johap TTY in the ko Locale	113
	▼ How to Use Combination Code and Johap TTY in the ko Locale	114
	▼ How to Use the EUC TTY in the ko.UTF-8 Locale	114
	▼ How to Use the Johap TTY in the ko.UTF-8 Locale	115
A	Binary Compatibility Package	117
	Korean Test Utilities	118

Asian-Specific Utilities	119
Conversion Utilities	119
Conversion Within a Code Set	120
Conversion for Korean Character Codes	120

B Mapping Korean Keyboard Functions 121

Glossary	123
----------	-----

Index	127
-------	-----

Tables

TABLE 1-1	Korean Locales	20
TABLE 1-2	Korean Code sets	20
TABLE 4-1	Character Code Standards	32
TABLE 4-2	Lookup Choice Navigation Keys	43
TABLE 4-3	Formatting Keys for Committed Strings	44
TABLE 6-1	Korean <code>iconv</code> Code Conversion Modules (<code>ko</code> locale)	75
TABLE 6-2	Common Korean <code>iconv</code> Code Conversion Modules (<code>ko</code> and <code>ko.UTF-8</code> locales)	77
TABLE 10-1	Korean Solaris Operating System DPS Fonts	92
TABLE 10-2	Korean Bitmap Fonts for the <code>ko</code> Locale	96
TABLE 10-3	Korean Bitmap Fonts for the <code>ko.UTF-8</code> Locale	97
TABLE 10-4	Korean TrueType Fonts for the <code>ko/ko.UTF-8</code> Locales	97
TABLE 12-1	<code>ioctl</code> Requests and Descriptions	105
TABLE 12-2	<code>termcap</code> Variables and Descriptions	105
TABLE 12-3	<code>terminfo</code> Variables and Definitions	106
TABLE A-1	Korean Character Classification Functions	118
TABLE A-2	Korean Code Conversion Functions	120

Preface

Korean Solaris User's Guide provides information for users, system administrators, and developers on the Solaris™ Operating System (Solaris OS). The guide also describes Korean localized desktop tools and utilities available with the Korean Common Desktop Environment (CDE).

About This Book

This guide offers information on using, customizing, and administering the Korean Solaris Operating System and CDE.

How This Book Is Organized

Each chapter of this guide covers a different aspect of the Korean Solaris software.

Part I introduces the features of the Korean Solaris Operating System.

- [Chapter 1](#) describes the new features and the language support that are available in this release.

Part II includes end-user information.

- [Chapter 2](#) discusses the design of the Korean Solaris software and provides information on the supported locales.
- [Chapter 3](#) describes the features of the Korean Common Desktop Environment (CDE) for windowed applications.
- [Chapter 4](#) describes the different Korean character input modes and provides procedures that you can use to enter and convert Korean characters.

- [Chapter 5](#) describes the Hanja tool features and provides procedures that you can use to create and edit local Hangeul-Hanja conversion dictionaries.
- [Chapter 6](#) describes how to use the `mailx`, `talk`, `xtobif`, `stdconvtool`, and `iconv` utilities. The chapter also provides tables of code conversion modules.
- [Chapter 7](#) describes line printer, PostScript™ printer, and laser printer support in this release.

Part III includes information for system administrators.

- [Chapter 8](#) provides procedures on setting a default locale, adding a new locale, and changing font directories.
- [Chapter 9](#) discusses input method server basics and application display.
- [Chapter 10](#) provides information on PostScript, bitmap, and TrueType fonts.
- [Chapter 11](#) discusses printing to line printers and to and PostScript printer.
- [Chapter 12](#) provides information on TTY STREAMS, command, setup, and terminal support.
- [Appendix A](#) discusses running compiled binary code of the SunOS™ 4.x, Solaris 1.x, and Asian OpenWindows 2.x applications recompilation.
- [Appendix B](#) provides information on Korean keyboard mapping.
- [Glossary](#) defines key words and phrases found in the Korean Solaris documentation.

Related Books

The following books include further details on the topics covered in this guide.

- *Solaris 10 Release Notes* offers the most up-to-date information about the features offered in this release.
- *Solaris 10 Installation Guide* describes the process of installing the Solaris Operating System.
- *Solaris Common Desktop Environment: Advanced User's and System Administrator's Guide* provides information on CDE for users and system administrators.
- *International Language Environments Guide* describes the internationalization and the localization features that are available for application developers in this release.

Accessing Sun Documentation Online

The docs.sun.comSM Web site enables you to access Sun technical documentation online. You can browse the archive or search for a specific book title or subject at <http://docs.sun.com>.

Typographic Conventions

The following table describes the typographic changes used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with on-screen computer output	<code>machine_name% su</code> Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You must be <i>root</i> to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	machine_name%
C shell superuser prompt	machine_name#
Bourne shell and Korn shell prompt	\$
Bourne shell and Korn shell superuser prompt	#

Features Overview

This part introduces the features of the Korean Solaris Operating System (Solaris OS).

Overview of Features

The Korean Solaris Operating System is the internationalization and the Korean localization of the Solaris Operating System and the Common Desktop Environment (CDE) window system.

This chapter describes the new features and the language support that are available in the Korean Solaris Operating System.

New Localized Features

- The Korean input methods, based on SunIM I/F (Sun Internet Intranet Input Method Interface), are available in the current Solaris release with new features and many enhancements.
 - Two new keyboard layouts are supported with the current Solaris Korean language engine.
 - 3 Beol Sik 390
 - 3 Beol Sik final
 - An auxiliary window is included that enables you to do the following tasks:
 - Adjust the behavior of input methods by changing the input method options.
 - Input characters from virtual keyboards.
 - Access all the symbols that are contained in a current character set with a single mouse click.
 - Use the online help menu to find answers to questions about the auxiliary window.
 - Support is available for the features that enables you to do the following tasks:
 - Enable or disable the option to automatically commit individual character.

- Use different options to convert characters and words from Hangul to Hanja.
- Choose from several keyboard layouts including 2 Beol Sik, 3 Beol Sik 390, and 3 Beol Sik final.
- Switch between half-width and full-width modes.
- Unicode 3.2 support. The `ko_KR.UTF-8` (`ko.UTF-8`) locale has been updated to support the new 3.2 version of the Unicode Standard. The new version introduces an additional 1,016 new characters and contains various normative and informative changes.

Unicode 3.2 also defines the following UTF-8 byte sequences as invalid.

- 0xED as the first byte.
 - 0xA0 to 0xBF as the second byte.
- These sequences exclude the surrogate code points between U+D800 and U+DFFF. To comply with the new definition, all Korean UTF-8 iconv modules have been enhanced to detect the newly defined invalid UTF-8 byte sequences.
- In the current Solaris release, the `mp` printing utility replaces the `xetops` and the `xutops` utilities.

Note – The `xetops` and `xutops` printing utilities are no longer supported in the Korean Solaris Operating System. The utilities were formerly used to convert Korean text files to PostScript. The conversion enabled the printing of Korean characters to PostScript printers with no resident Asian fonts. The `xetops` utility was used in the `ko_KR.EUC` locale. The `xutops` utility was used in the `ko_KR.UTF-8` locale.

The `mp` printing utility was first released with the Solaris 9 Operating System.

Language Support

The Solaris Operating System builds inherent internationalization features into every localized product. Localization facilities support the ANSI C recommendations for internationalization and localization that define the locale and related categories.

Locale Attributes

A *locale* contains a language with culturally specific information and conventions for a particular global region. Each process in the Solaris Operating System has the following set of locale attributes:

- Locale settings, which provide the `locale` and `setlocale` commands you use to list and set attributes before you start a process from the command line.
For example, the Korean locales and the English/ASCII locale both have a category that defines the display of time and date according to the cultural format, as well as the actual Korean or English/ASCII characters for time and date.
- Code sets, which support coding conventions for the KS X 1001 and KS X 1005-1 character sets. These code sets enable you to input, display, and print Korean text in file names, system messages, and terminal (TTY), email, and data file content.
- `htt` input method server, which handles Korean input for the Korean Solaris Operating System. The `htt` server receives your keyboard input and converts it to Korean characters that are used in Korean Solaris applications.

Korean Locales

In December 1995, the Korean government announced a standard Korean code set, KS X 1005-1, which is based on ISO 10646-1/Unicode 2.0.

The ISO-10646 character set uses two universal character sets:

- UCS-2. Universal Character Set (two-byte form)
- UCS-4. Universal Character Set (four-byte form)

The ISO-10646 character set cannot be used directly on IBM PC-based operating systems. For example, the kernel and many other modules of the Korean Solaris Operating System interpret certain byte values as control instructions, such as a null character (0x00) in any string. The ISO-10646 character set can be encoded with any bit combinations in the first or subsequent bytes. The ISO-10646 characters cannot be freely transmitted through the Solaris system with these limitations.

In order to establish a migration path, the ISO-10646 character set defines the UCS Transformation Format (UTF), which encodes the ISO-10646 characters without using C0 controls (0x00..0x1F), C1 controls (0x80..0x9F), space (0x20), and DEL (0x7F).

The `ko_KR.UTF-8` locale supports KS X 1005-1, the Korean standard code set. The locale supports the characters of the previous KS X 1005 code set, all of the 11,172 Korean characters, and the extended ASCII code set. Until the Universal UTF/UCS becomes available, the `ko_KR.UTF-8` locale supports ISO-10646 code subset that is related to the Korean characters and fonts. The ISO-10646 standard covers all characters in the world. With the input methods and fonts provided in this release, you can enter, display, and print characters of any language.

In the `ko_KR.EUC` locale, the EUC scheme is used to encode KS X 1001. The `ko_KR.UTF-8` locale supports the KS X 1005-1/Unicode 3.2 code set, which is a superset of KS X 1001. These two locales look the same to the end user, but the internal character encoding is different.

The Korean Solaris Operating System provides simultaneous support for the locales in the following table. The locales look the same to the end user, but the internal character encoding is different.

TABLE 1-1 Korean Locales

Locale	Description
ko_KR.EUC (ko)	Korean EUC (KS X)
ko_KR.UTF-8 (ko.UTF-8)	Korean UTF-8 (Unicode 3.2)

Korean Code sets

The following table lists the supported code sets for each Korean locale.

TABLE 1-2 Korean Code sets

Locale	code set
ko_KR.EUC (ko)	KS X 1001
ko_KR.UTF-8 (ko.UTF-8)	KS X 1005-1/Unicode 3.2

Korean Input Methods and Fonts

The Korean Solaris Operating System provides input methods and fonts for all characters covered the ISO-10646 standard. These methods and fonts enable you to enter, display, and print any character in any language.

The following features are supported by the Korean input methods that are available for the ko_KR.EUC (ko) and the ko_KR.UTF-8 (ko.UTF-8) locales:

- Hangul 2 Beol Sik keyboard support
- Hangul 3 Beol Sik 390 keyboard support
- Hangul 3 Beol Sik final keyboard support
- Hangul-Hanja conversion
- Special character input
- Character input using hexadecimal code

For a complete list of scalable and bitmap fonts supported for the ko_KR.EUC (ko) and the ko_KR.UTF-8 (ko.UTF-8) locales, see [Chapter 10](#).

Note – You can use Hangul or standard Sun keyboards to enter Korean text.

Locale Categories

In the Korean Solaris Operating System, you can use the following general and specific categories as defined by ANSI C for the Korean and English locales:

- General LC_ALL setting that invokes all of the categories for locale-related aspects of the environment.
- Specific settings for particular aspects of the environment, which include:
 - LC_CTYPE
 - LC_TIME
 - LC_NUMERIC
 - LC_MONETARY
 - LC_COLLATE
 - LC_MESSAGES

For example, the Korean and the English/ASCII locales have the LC_TIME category that defines the display of the time and date according to the cultural format, as well as the actual Korean or English/ASCII characters used in the display.

Locale Modifier

Aspects of a locale-sensitive operation can be modified by using a locale modifier. The output of the locale command is:

```
system % locale
LANG=ko_KR.EUC
LC_CTYPE="ko_KR.EUC"
LC_NUMERIC="ko_KR.EUC"
LC_TIME="ko_KR.EUC"
LC_COLLATE="ko_KR.EUC"
LC_MONETARY="ko_KR.EUC"
LC_MESSAGES="ko_KR.EUC"
LC_ALL=ko_KR.EUC

system % cat data_file
```

```
1
  あ
  寒
  한
```

If you sort the data_file, the sort result is:

```
system % sort data_file
```

```
1
  あ
  한
  寒
```

In this case, the text is sorted on the code point value of each character defined in the current locale, `ko` (`ko_KR.EUC`). This might not be the desired result.

The Solaris operating environment provides a locale modifier. You can modify the behavior of `sort` by changing the current locale from `ko` to `ko_KR.EUC@dict` as shown:

```
system % env LANG=ko_KR.EUC@dict sort data_file
```

```
1
  한
  寒
  あ
```

PART II

End-User Information

This part includes end-user information.

Introduction to the Korean Solaris Software

This chapter describes the design of the Korean Solaris software. The chapter also provides information about the locales supported in the Korean Solaris environment. The chapter discusses the following topics:

- “Design of the Korean Solaris Software” on page 25
- “Korean Locales” on page 25
- “How to Set the Locale at Login” on page 26

Design of the Korean Solaris Software

The Korean localization of the internationalized release of CDE provides two work environments: a user environment and a development environment. Each environment is localized to handle the linguistic and cultural conventions that are unique to the Korean language.

- The user environment has desktop tools and a window manager (`dttwm`) that are customized to communicate in the language of a particular locale.
- The development environment provides internationalized versions of `Xlib` and `Motif` that programmers use to develop localized applications. For further information, see the *International Language Environments Guide*.

Korean Locales

You can set any of the following locales when you login to your Korean desktop:

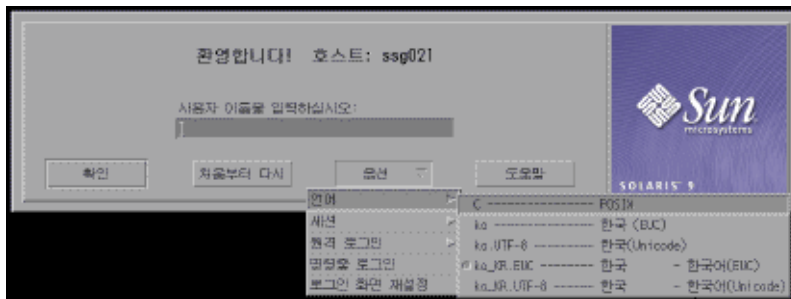
- `C` – ASCII English environment.

- ko – Korean extended UNIX code (EUC). This locale supports the KS C 5601-1992 standard.
- ko.UTF-8 – Korean Universal Multiple Octet Coded Character Set (UCS) Transmission Format. This locale supports the Unicode 3.2 standard.
- ko_KR.EUC – Symbolic link to the ko locale.
- ko_KR.UTF-8 – Symbolic link to the ko.UTF-8 locale.

▼ How to Set the Locale at Login

You can set your default locale or change it using the following procedure.

1. Choose Language from the Options menu on the login screen.



2. Select the desired locale for your CDE session.

Your choices are the C, ko, ko.UTF-8, ko_KR.UTF-8, and the ko_KR.EUC locales.

CDE

The following sections provide information on the Korean localization of the Korean Common Desktop Environment (CDE) for windowed applications.

- “Korean Graphical User Interface” on page 27
- “Localized CDE Desktop” on page 28

Korean Graphical User Interface

The Korean CDE Motif graphical user interface is similar in layout and design to the U.S. release of CDE. Korean CDE supports multibyte characters and Korean messages with Motif objects. Differences in character width and proportional spacing cause minor differences in the exact layout of some Motif objects.

All application windows that can take Korean input include a *status area* associated with their input window to show the current conversion mode. With an input conversion mode on, as Korean is being typed its entry point becomes a highlighted (reverse video) *preedit area* until the input is converted to Korean or special characters and committed. Some input modes also provide conversion choices among several Korean characters on menus.

Korean Input/Output

To accommodate the diversity of Korean, the Solaris software provides several different input methods for entering Korean characters. With these methods you can enter ASCII/English characters and Korean Hangul/Hanja characters using an ASCII keyboard or a Korean keyboard.

Your Korean input is stored temporarily as an intermediate representation. The Korean language engine then transforms the representation into the displayed character string.

Localized CDE Desktop

The following desktop tools are available in this release. All of the tools can handle Korean input and output.

- Address Manager – Carries out remote operations and finds information about the systems and users on your network. This application can speed up such tasks as sending email, logging in remotely, and setting appointments on someone else’s calendar.
- Application Manager – Manages the tools and other software applications available on your system.
- Audio Tool – Records, plays, and saves audio files in AU, AIFF, and WAV format.
- Calculator – Mimics the function of a hand-held calculator.
- Calendar Manager – Manages appointments and To Do lists. You can use this application to set and distribute appointment reminders.
- Clock – Displays the current time in analog or digital format. You can control the display of local time with this application.
- Console – Starts a `dtterm` terminal emulator as your workspace console window.
- File Manager – Displays the files and folders on your system. You can move, copy, open, and delete files and folders with this application. You can also use the application to view the contents of your floppy diskettes and CD-ROM discs.
- Find Files – Enables you to search your system for files or folders that match specific search criteria. Your criteria might include, for example, the name, location, or the size of a file or folder you want to find.
- Front Panel – Contains controls for access to applications and utilities on the system. The Front Panel is a centrally-located window that occupies all workspaces.
- Help – Displays searchable help information for CDE.
- Icon Editor – Enables you to create new icons or modify existing icons.
- Image Viewer – Enables you to view, print, and save the contents of file types such as GIF, TIFF, JPEG, and PostScript. You can use the Snapshot function of this application to capture a picture of a window or another part of your screen. The picture is saved as a raster file in bitmap format.
- Mailer – Handles the distribution and receipt of your electronic mail messages.
- Performance Meter – Monitors various aspects of system performance.

- Print Manager – Enables you to submit, view, and cancel print jobs. This application is the graphical front-end to the print command. It supports drag-and-drop file transfer operations.
- Process Manager – Displays the processes that are currently running on your workstation. This information enables you to perform actions on the active processes.
- Style Manager – Enables you to customize some of the visual elements and system device behaviors of your workspace environment, such as colors and fonts, keyboard, mouse, window, and session start-up behaviors.
- Terminal – Acts as an ASCII character terminal that you can use to enter UNIX[®] commands at a system shell prompt.
- Text Editor – Enables you to create and edit text files. This application is used in CDE tools such as the Mailer composition window.

Entering Korean Text

This chapter provides procedures and other information that you can use to enter Korean text.

This chapter includes the following sections:

- “Supported Character Sets” on page 32
- “Input Window Areas” on page 33
- “Basic Functions for Korean Input” on page 34
- “Switching Keyboard Layouts” on page 40
- “Using Hangul/Hanja Conversion to Enter Hanja” on page 41
- “Entering Special Symbol Characters” on page 47
- “Entering Hangul Characters Using Hex Input Mode” on page 51
- “Switching Between the Half_Width and the Full_Width Character Modes” on page 55
- “Auxiliary Windows for the Korean Input Method” on page 56

Character Input Overview

This chapter describes the Korean Solaris input modes that you can use to enter the following kinds of characters.

- ASCII/English
- Hangul
- Hanja
- Special symbols

You can type all of these characters in the input areas of the following application windows:

- In terminal emulation (TTY) windows, such as Terminal windows
- In text entry areas, such as those found in the Text Editor and Mailer applications

- In dialog boxes, such as the new folder name box in the File Manager application
- In other special use subwindows, such as pop-ups

Supported Character Sets

The locale that you choose determines the characters that are available for input. If you select the `ko_KR.EUC` locale, for example, you can enter the characters in the KS X 1001 code set. In the `ko_KR.UTF-8` locale, you can input all of the 11,172 Korean characters that are composed according to the Johap principle. The following descriptions summarize the encoding standards that define the characters for the `ko_KR.EUC` locale and for the `ko_KR.UTF-8` locale.

TABLE 4-1 Character Code Standards

ko_KR.EUC (ko) locale	
Wansung code	This two-byte code is specified in Korean Industry Standard KS X 1001, known as KS C 5601-1987, for Hangul, Hanja, and other characters. In the Korean Solaris software, the KS X 1001 characters are in the EUC code set.
ko_KR.UTF-8 (ko.UTF-8) locale	
Johap or Packed code	This two-byte code consists of a leading bit followed by three 5-bit fields. The three fields each contain the codes or a leading consonant that is followed by a vowel and a final consonant, if there is one, for a Hangul character. This two-byte code is specified in Korean Industry Standard KS C 5601-1992-3.

TABLE 4-1 Character Code Standards (Continued)

ko.UTF-8	<p>Korean Universal Multiple Octet Coded Character Set (UCS) Transmission Format. ko.UTF-8 supports all the characters of KS C 5601 and the 11,172 characters from Johap, as well as all Korean-related Unicode 3.2 characters and fonts.</p> <p>ko.UTF-8 supports the following subset of Unicode:</p> <ul style="list-style-type: none">■ Basic Latin and Latin-i (190 characters) – Row 00 of BMP (Basic Multilingual Plane)■ Symbolic characters – Row 20 to Row 27, and Row 32 of BMP including box (line) drawing characters that are defined in KS C 5601■ Numerals that are defined in KS C 5601 (20 characters) – Row 21 and Row FF of BMP■ Roman, Greek, Japanese, and Cyrillic alphabet characters that are defined in KS C 5601 (362 characters) – Row 02, Row 04, Row 30 and Row FF of BMP■ Jamo (Hangul alphabet) characters (94 characters) – Row 31 of BMP■ Pre-composed Hangul syllables (11,172 characters) – From Row AC to Row D7 of BMP■ Hanja characters defined in KS C 5601 (4,888 characters) – From Row 4E to Row 9F and from Row F9 to Row FA of BMP
----------	---

Input Window Areas

In the Korean Operating System, application subwindows contain two areas that are used to enter Korean characters. A lookup choice window and an auxiliary window are also available in Korean input mode.

- **Preedit area** – The text entry area that holds your character formations before you *commit* them.

Note – When you commit characters, the characters are put in the text block that is assembled for the application.

- **Status area** – The area at the lower left of the application subwindow that displays the current conversion mode and the active keyboard. Later sections in this chapter discuss keyboard switching and using the available conversion modes.
- **Lookup choice window** – A popup window that displays Hanja or special character lookup choices which are available for the conversion of characters or radicals in the preedit area.

- **Auxiliary window** – This window contains a *palette* of icons that you can click to make the following choices:
 - Select your input mode
 - Toggle between the full_width and the half_width character modes
 - Convert Hangul characters in the preedit area to Hanja
 - Select your virtual keyboard

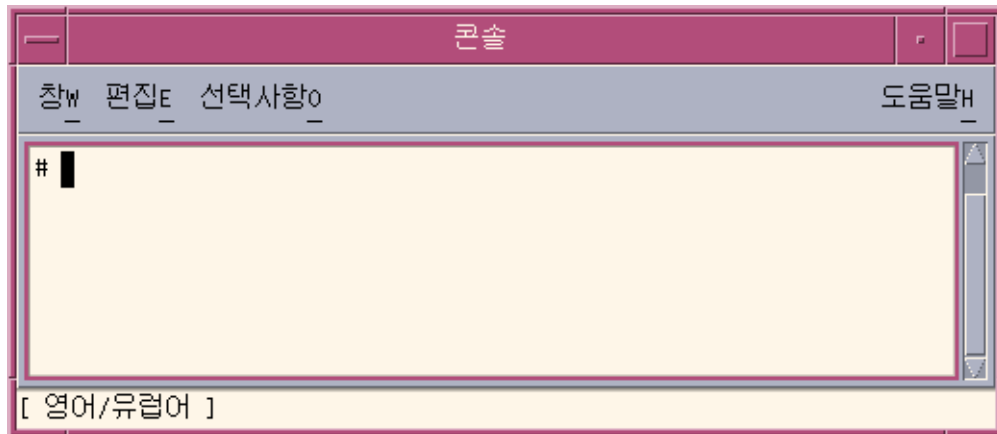
Basic Functions for Korean Input

This section describes the process you use to enter, edit, and commit Hangul characters. The section also provides information on switching keyboard layout.

Starting to Input Hangul

ASCII/English is the initial default input mode. When the status area at the lower left corner of the application subwindow shows the input mode as ASCII/English, whatever you type from the keyboard will appear as ASCII text.

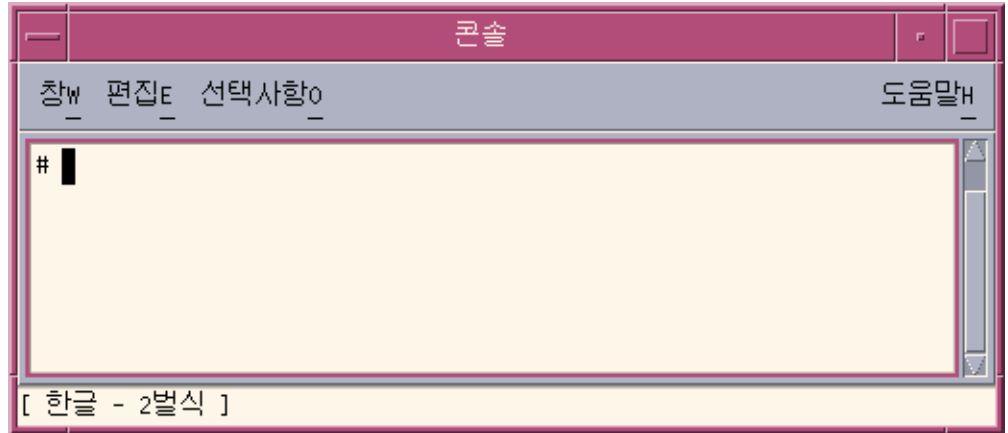
The status region in the lower left corner shows ASCII/English input mode.



▼ How to Switch to the Hangul Input Mode

- To enter Korean characters, type Control-spacebar to turn on Hangul input conversion. Alternatively, if you have a Hangul keyboard, you can press the Hangul/English key to turn on Hangul input conversion.

The status region shows Hangul as the conversion mode. The area also shows the active keyboard layout.



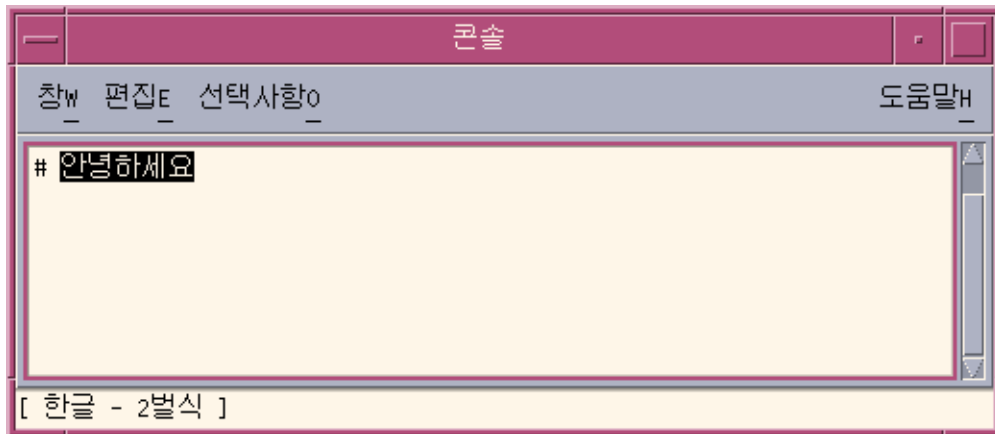
Note – You can also use Control-spacebar or the Hangul/English key to toggle off Hangul input conversion.

When Hangul input conversion is on, an auxiliary window appears with a palette of icons that you can use to make mode and keyboard selections.



▼ How to Enter Hangul Characters

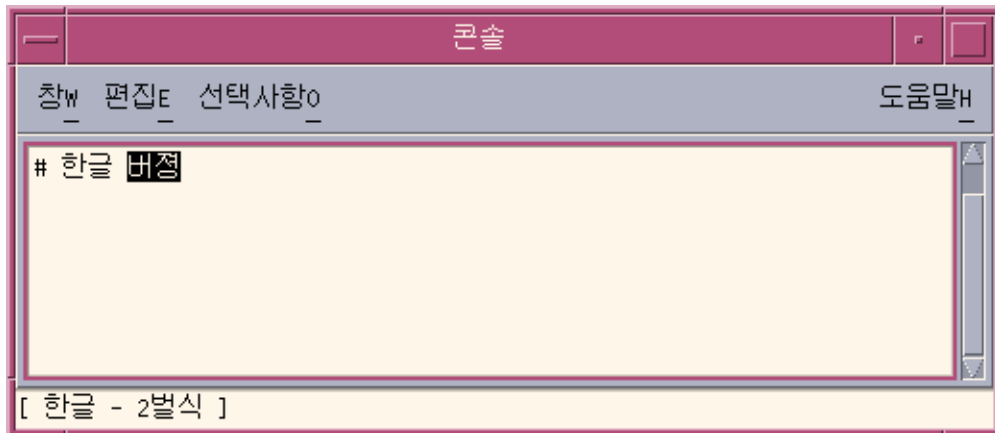
1. Type Control-spacebar press the Hangul/English key on a Hangul keyboard to turn on Hangul input conversion.
2. Type the appropriate sequence of keystrokes to compose Hangul characters in the preedit area.



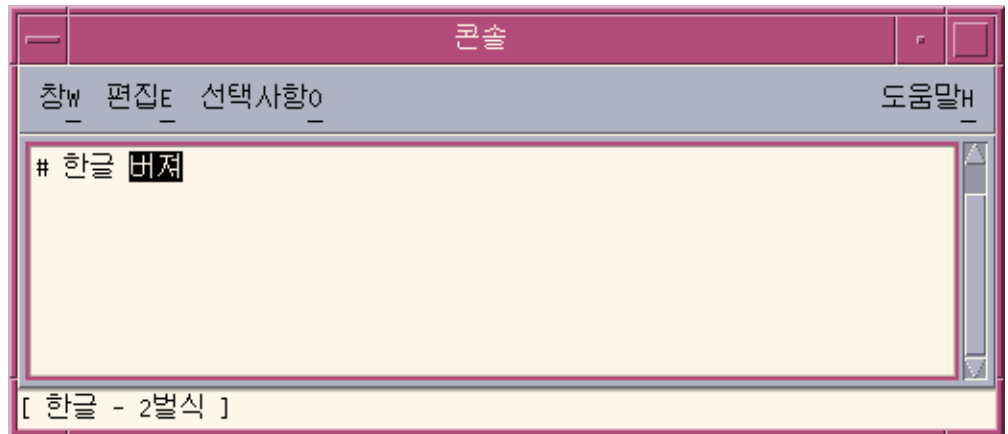
▼ How to Backspace and Delete Korean Characters

The Backspace and Delete keys remove the last character (JaMo) of the current syllable. The following procedure demonstrates how to delete characters of a syllable with the Backspace key.

1. Notice that the text you are composing appears highlighted in the preedit area.

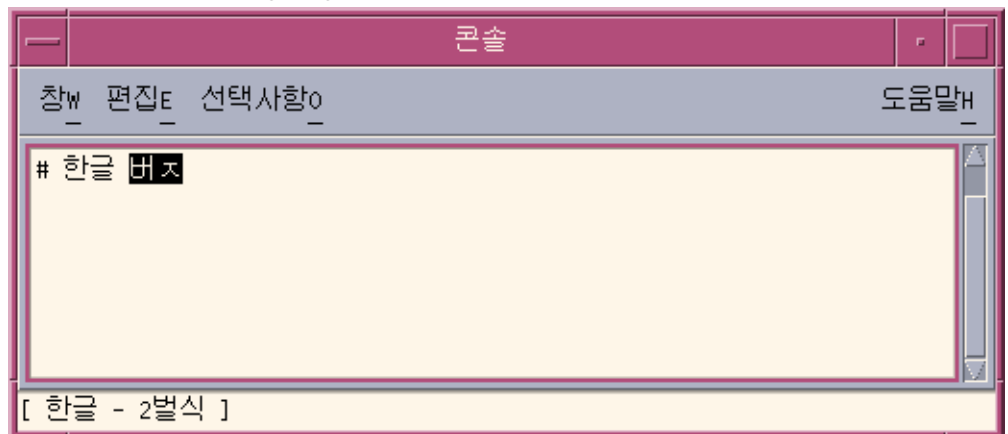


2. To remove the last character of the last syllable, press the Backspace key once. Notice that the last character of the syllable changes.



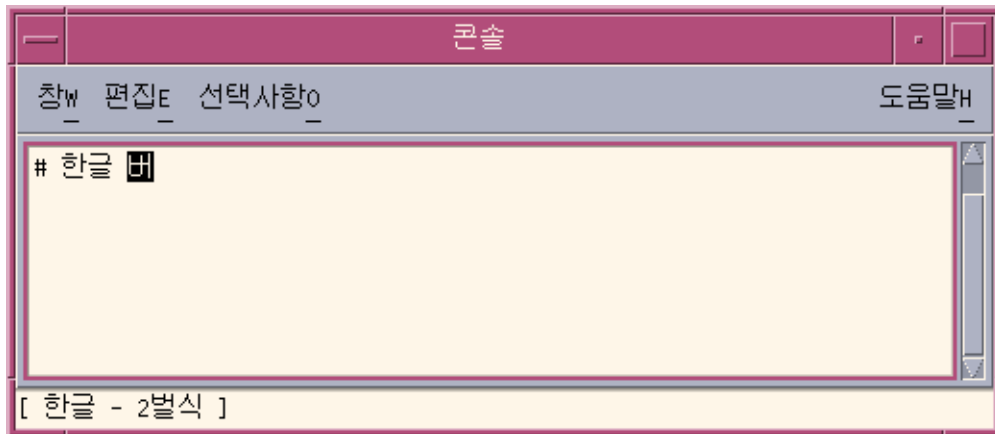
3. Press the Backspace key a second time.

The last character changes again.

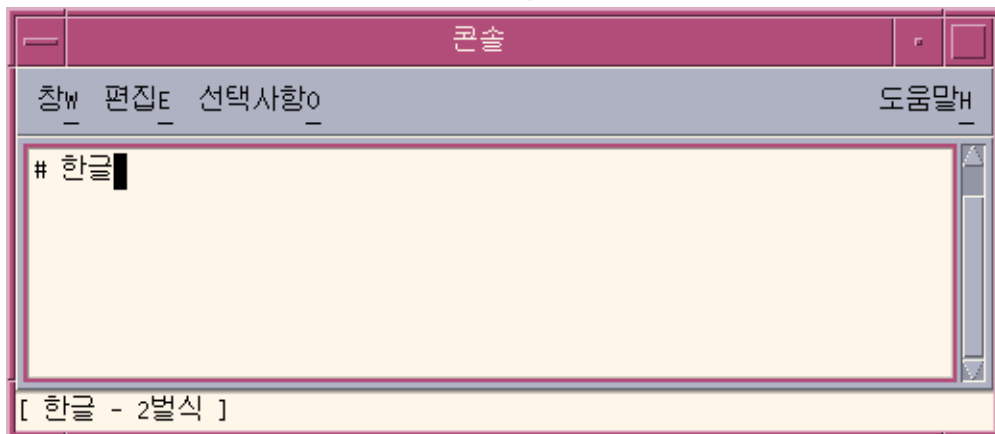


4. Press the Backspace key a third time.

Notice that only the first character of the syllable remains.

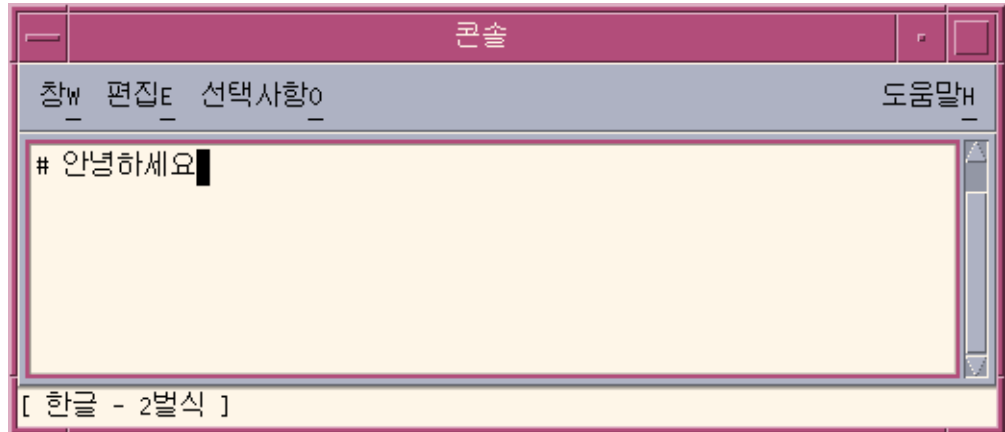


5. Press the Backspace key twice more.
You have now removed each character in the syllable.



▼ How to Commit Hangul Input

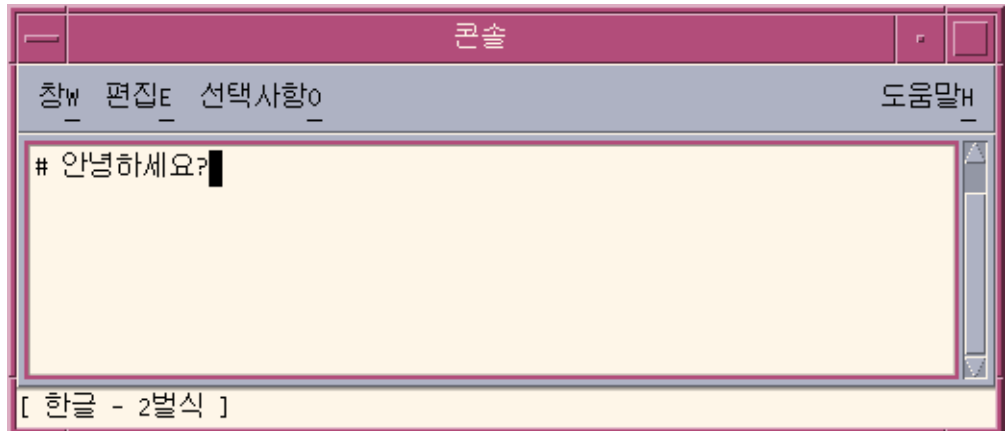
The following figure shows five Hangul syllables that have not been *committed*. Until the syllables are added to a text block that is being assembled for the application, the syllables are not considered to be committed.



- **Press Control-K to commit the syllables.**

Control-K is treated as a nonprinting character that is not committed with the Hangul syllables in the input string.

A character such as a space, a number, a punctuation mark, or another printable character can be committed to appear with Hangul syllables.



▼ How to Return to the ASCII/English Input Mode

- When you finish using the Hangul input mode, press Control-spacebar to return to the ASCII/English input mode.

Switching Keyboard Layouts

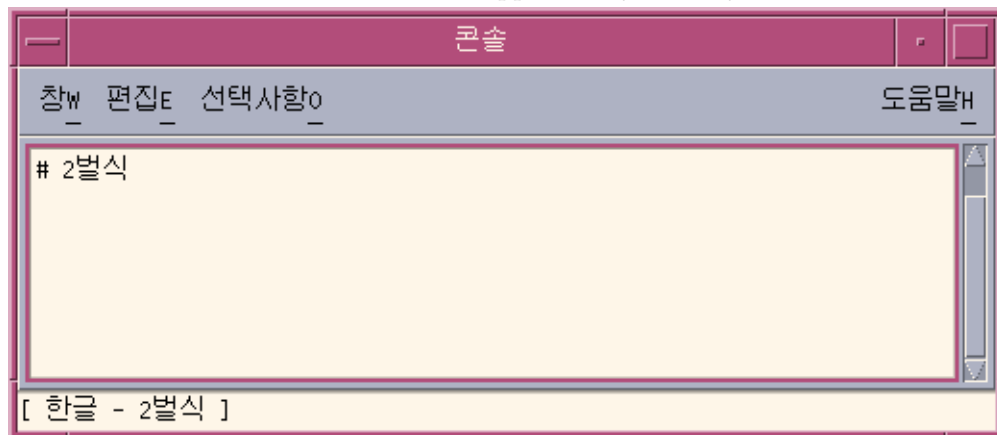
You can switch to one of the three most commonly used keyboard layouts, each of which is supported for use on your Korean desktop:

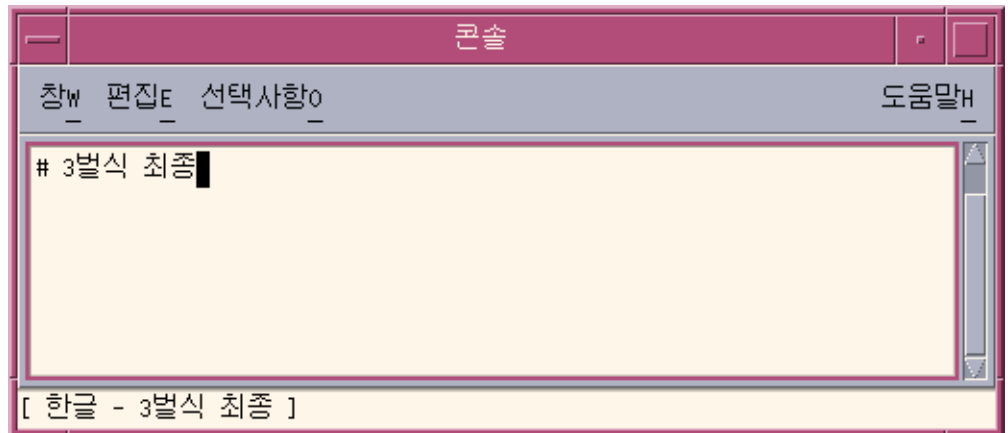
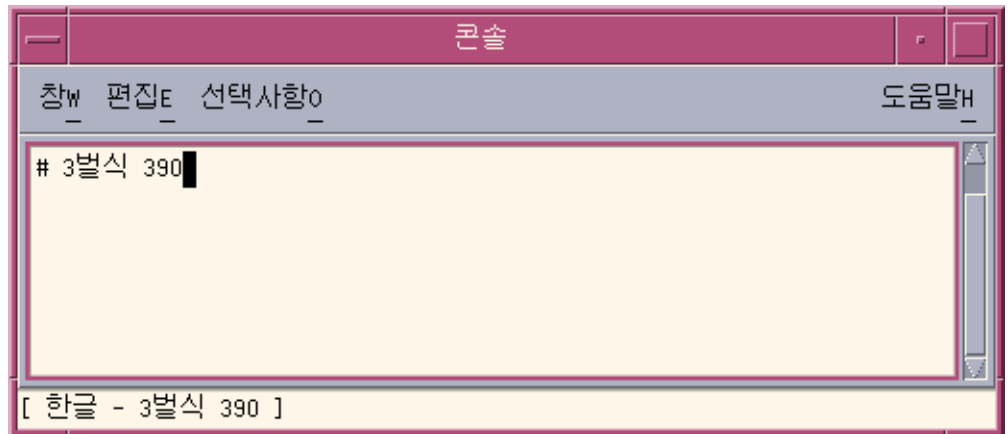
- 2 Beol Sik
- 3 Beol Sik 390
- 3 Beol Sik final

▼ How to Switch Keyboard Layouts

- **Press Shift-F2 to switch from one keyboard layout to another.**

The following figures show three application subwindows. In each subwindow, the status area indicates that one of the three supported keyboard layouts is active.





Using Hangeul/Hanja Conversion to Enter Hanja

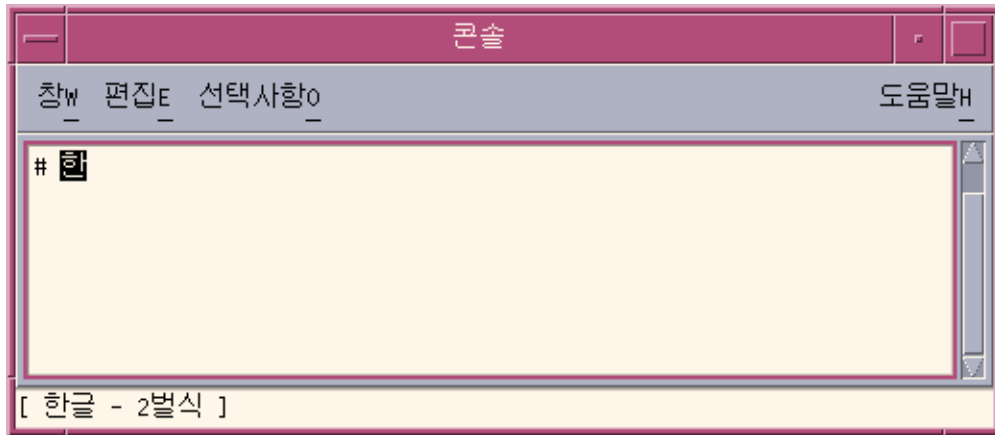
You can enter Hanja by converting Hangeul input to Hanja character by character or word by word. You convert the Hangeul characters to Hanja in the preedit area in one of two ways.

- Select the Hanja for your pronunciation from the candidates that appear in a lookup choice window.
- Select the Hanja for your pronunciation from the candidates that you cycle through in the preedit area.

The procedures in this section explain how to convert Hangeul to Hanja character by character. You can use the same methods to convert Hangeul to Hanja word by word.

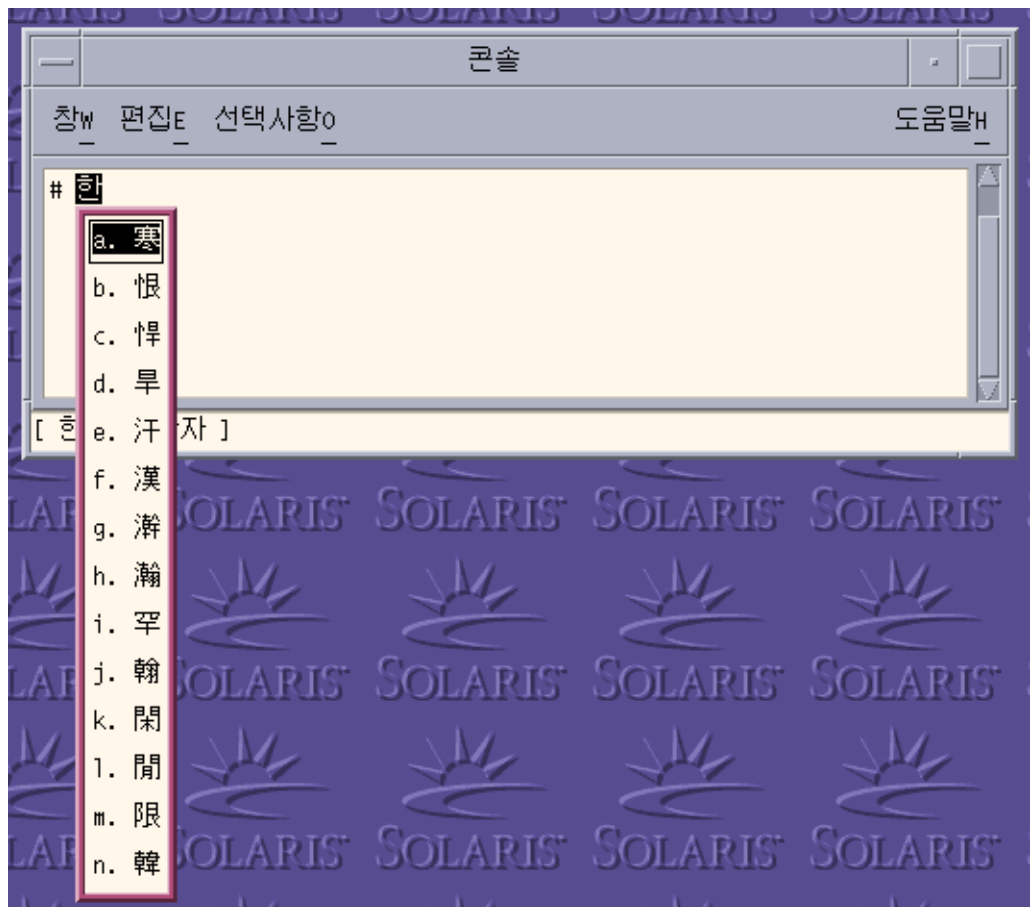
▼ How to Use the Hanja Lookup Mode

1. Press **Control-spacebar** to turn on Hangeul input conversion.
2. Type the appropriate sequence of keystrokes to compose a Hangeul character in the preedit area for your Hanja pronunciation.



3. Type **Control-W** to display the lookup window.

Note – If the Hanja candidate is the only one in the dictionary, Control-W could automatically commit the candidate depending on option set in the Input Method Options Auxiliary window. For information on the Options Auxiliary window, see “Input Method Options” on page 57



4. Press the following keys to navigate page by page through the lookup choices that are available for the Hanja pronunciation you enter.

TABLE 4-2 Lookup Choice Navigation Keys

Keys	Operation
Control-N	Next Page
Page Up	Next Page
Control-P	Previous Page
Page Down	Previous Page

5. When you find a page with the character you want, click the character or press the Enter key to select it. To move from one character to another on the same page, press the Spacebar.

6. You change the format of a committed string using the keys in the following table.

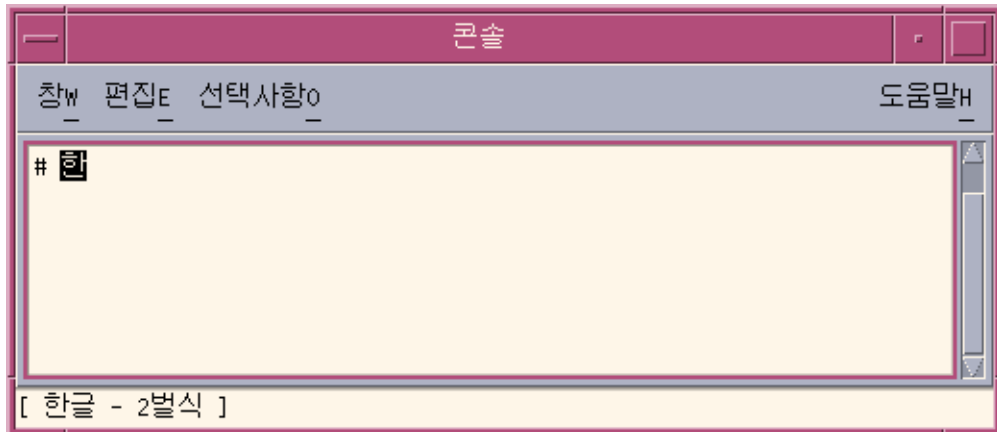
TABLE 4-3 Formatting Keys for Committed Strings

Keys	Committed Format
Control-Q	Hanja characters
Control-E	Hangul characters (converted Hanja characters)
Control-R	Hanja characters (converted Hangul characters)
Control-W	Format set in the Input Method Option Auxiliary window

▼ How to Convert Hanja to Hangul in Character Step Mode

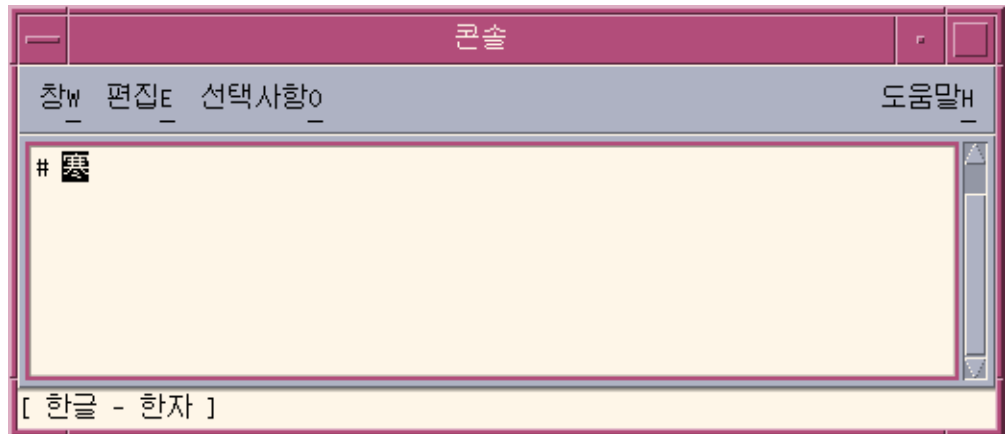
1. Press **Control-spacebar** to turn on Hangul input conversion.
2. Type the appropriate sequence of keystrokes to compose a Hangul character in the preedit area for your Hanja pronunciation.

The status area indicates that you are in Hanja conversion mode.

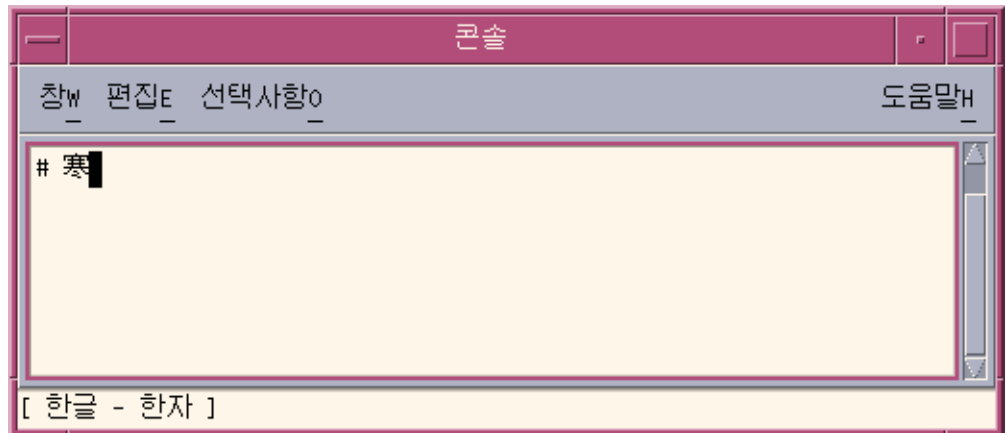


3. Press **Control-N** to convert the Hangul to Hanja.

You can press Control-N again to see another Hanja choice if one exists. Once you cycle through the available choices, the original Hanja reappears. Control-P converts the Hanja to a previous choice.



4. Press Control-K or any other nonprinting character except Control-N or Control-P to commit the highlighted Hanja. The preedit area next to the committed Hanja is then open for new Hangul input.



You can type a space, a number or another printable character to commit the nonprinting character along with the Hanja. When the highlighted Hanja and the printable character are committed, the preedit area next to the character is open for new Hangul input.

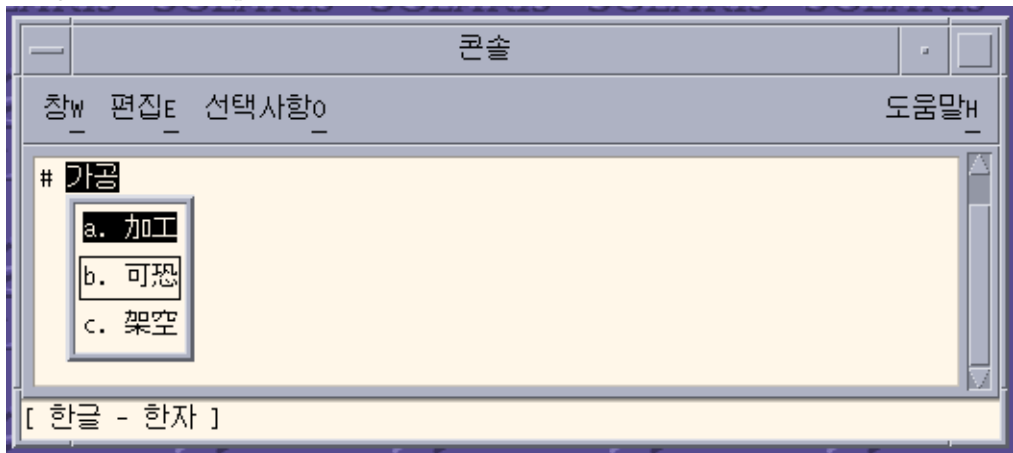
▼ How to Convert Hanja to Hangeul Using Word Conversion Modes

You can convert Hangeul to Hanja, word-by-word, in a similar manner converting Hangeul to the character by character as described in the previous section. The steps that you use for word by word conversion are the same as those for character conversion, with one exception. You turn on Hangeul input conversion after you enter a Hangeul word in the preedit area.

1. **Type the appropriate sequence of keystrokes to compose a Hangeul character in the preedit area for your Hanja pronunciation.**
2. **Press Control-spacebar to turn on Hangeul input conversion.**
3. **Press Control-N to use step mode or Control-W to use the lookup mode to see the available Hanja choices.**

The choices consist of Hanja words as well as Hanja characters for the last syllable of a Hangeul word.

The following figure shows the Hanja word choices that are available for the Hangeul word in the preedit area.



4. **Press Control-K or any other nonprinting character except Control-N or Control-P to commit the highlighted Hanja. The preedit area next to the committed Hanja is then open for new Hangeul input.**

You can type a space, a number, or another printable character to commit the character along with the Hanja. When the highlighted Hanja and the printable character are committed, the preedit area next to the character is open for new Hangeul input.

Entering Special Symbol Characters

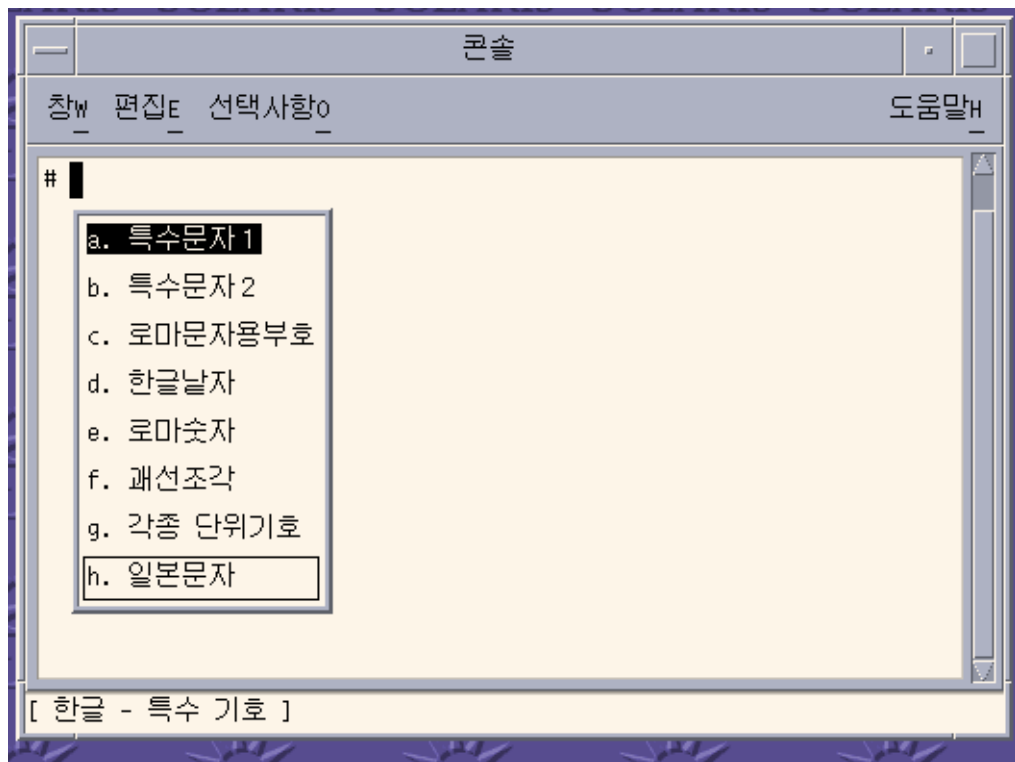
In the symbol input mode, you can enter non-Hangul/Hanja characters that you cannot type directly on the keyboard. Once you enter symbol input mode, the process you use to page through lookup choices and to make selections is the same as the one you use in Hanja lookup mode.

- Press Control-N or Control-P to move forward or backward through lookup choice pages.
- Type the letter of the choice to commit the symbol and to return to the Hangul input mode.
- Press Control-A at any point to abort the symbol lookup without making a selection, and to return to the Hangul input mode.

▼ How to Enter Special Symbol Characters

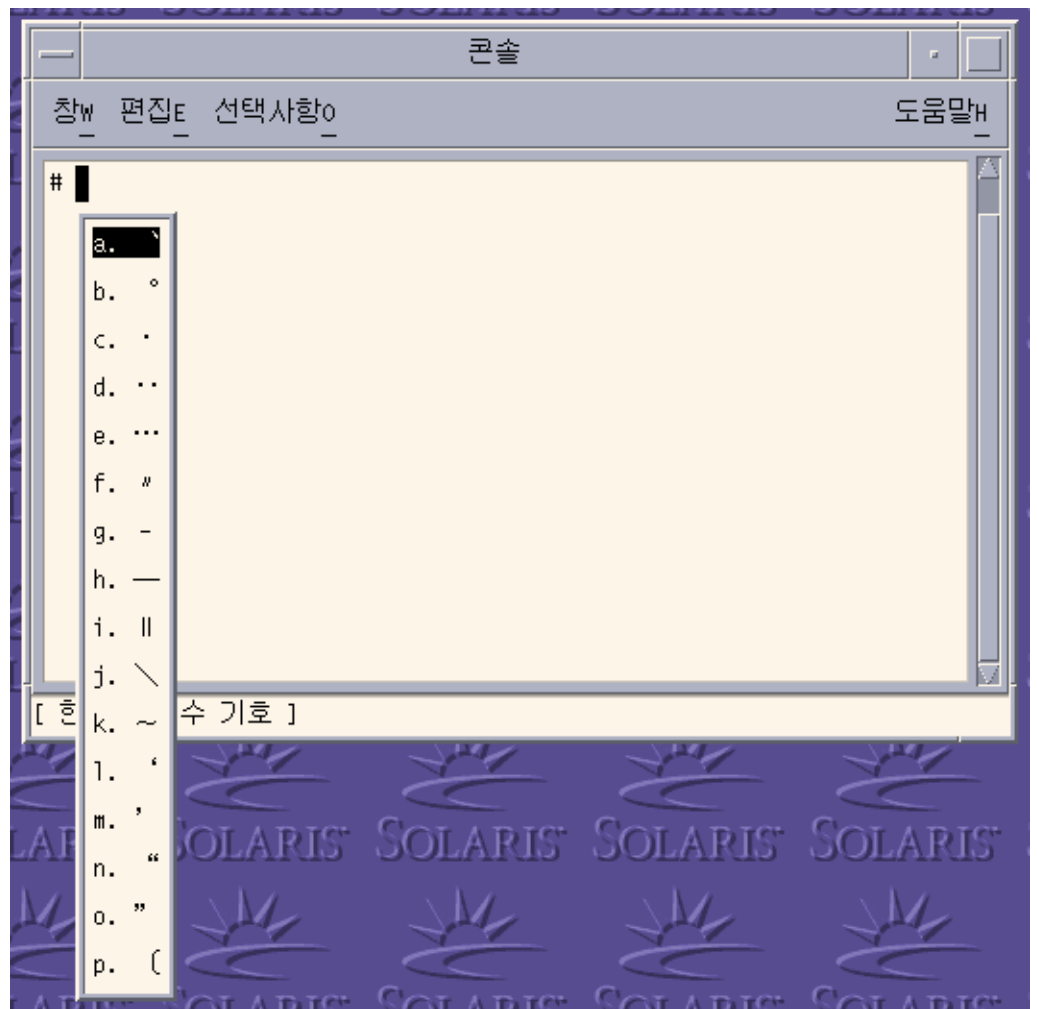
1. **Press Control-spacebar to turn on Hangul input conversion.**
2. **Press Control-J to turn on symbol input mode.**

The input method server displays a page of category choices and the status area indicates that you are in symbol input mode.

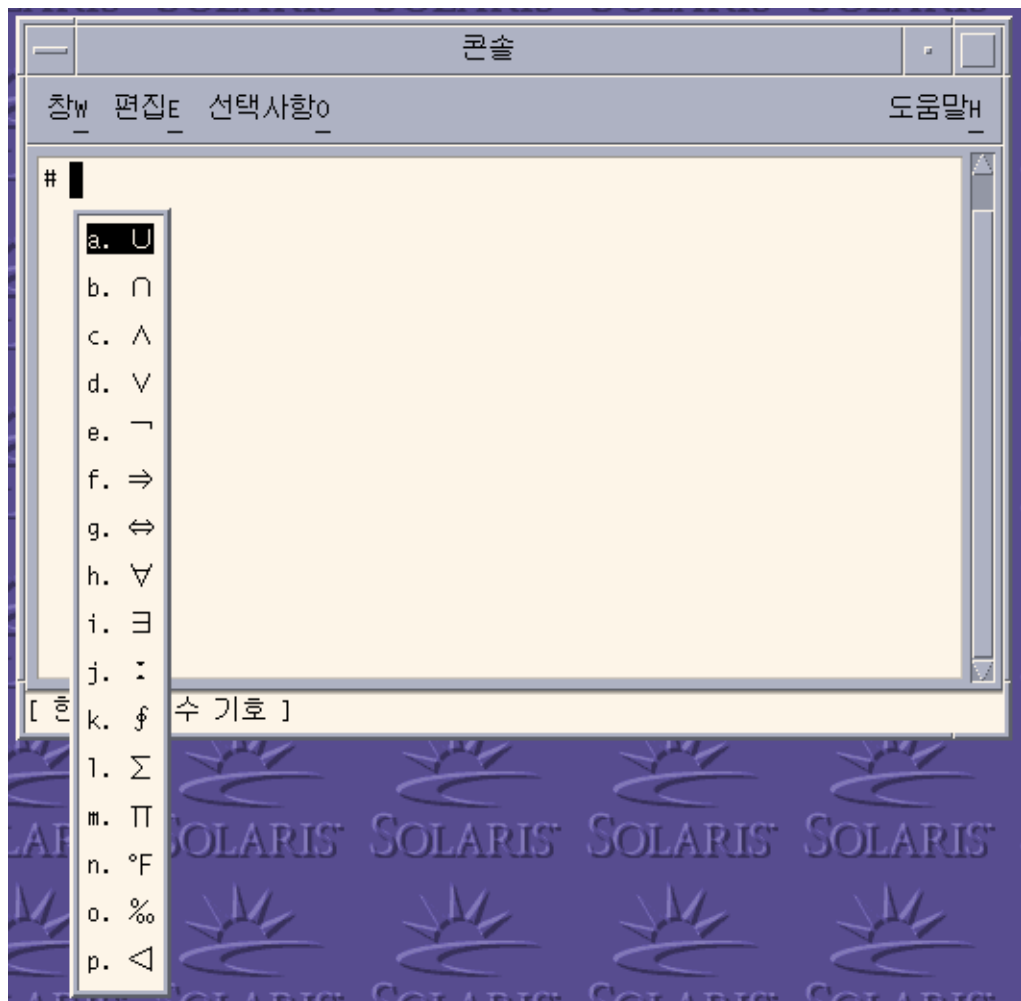


3. Type the letter of a category for the page of choices you want to display.

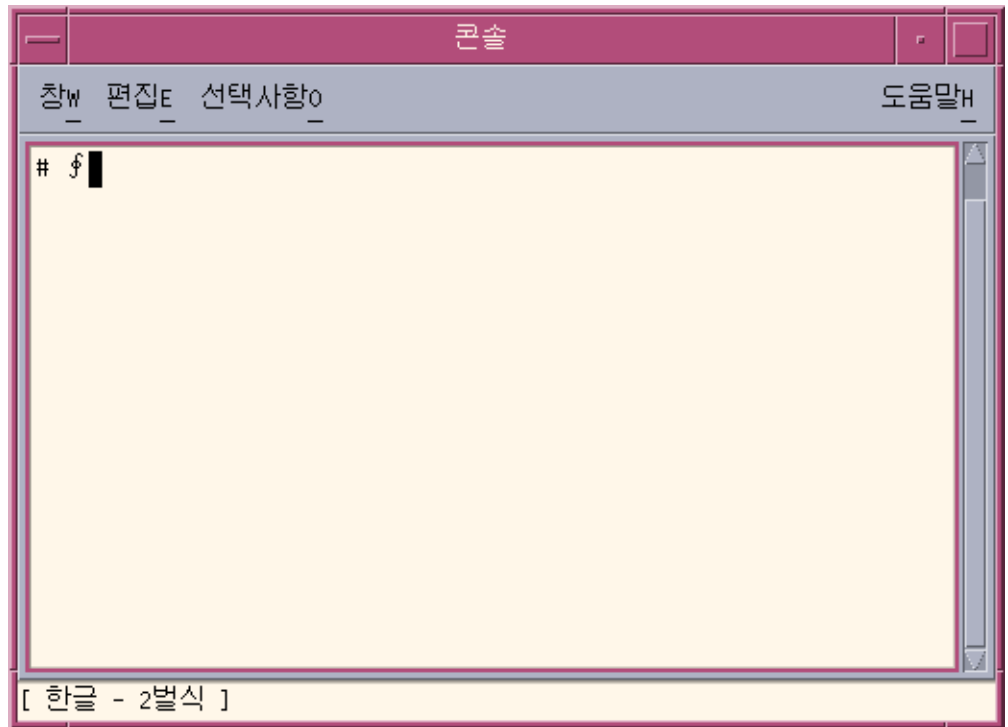
Category a is selected in the following figure. You can also scroll down the list of choices with the spacebar and press Enter to make a selection.



4. Press Control-N six times in category you selected in previous step to see the following page.



5. Type the letter **k** to commit the corresponding character of the page displayed in the previous step.



Entering Hangul Characters Using Hex Input Mode

You can enter any character defined in a character set supported for a locale by typing the encoding value of the character.

- In the `ko_KR.EUC` locale, you can use the KS X 1001 encoding values to enter the characters that are defined in the character set.
- In the `ko_KR.UTF-8` locale, you can use the encoding values that are defined in the following character sets to input the characters of each set.
 - KS X 1001, known as KS C 5601–1987
 - UTF-8 encoded values from Unicode 3.2

The following table shows you the key sequences you can use to select character sets in each locale.

Current Locale	Key Sequence	Character Set
ko_KR.EUC	Control-X	KS X 1001
ko_KR.UTF-8	Control-X	KS X 1001
ko_KR.UTF-8	Control-Shift-X	UTF-8

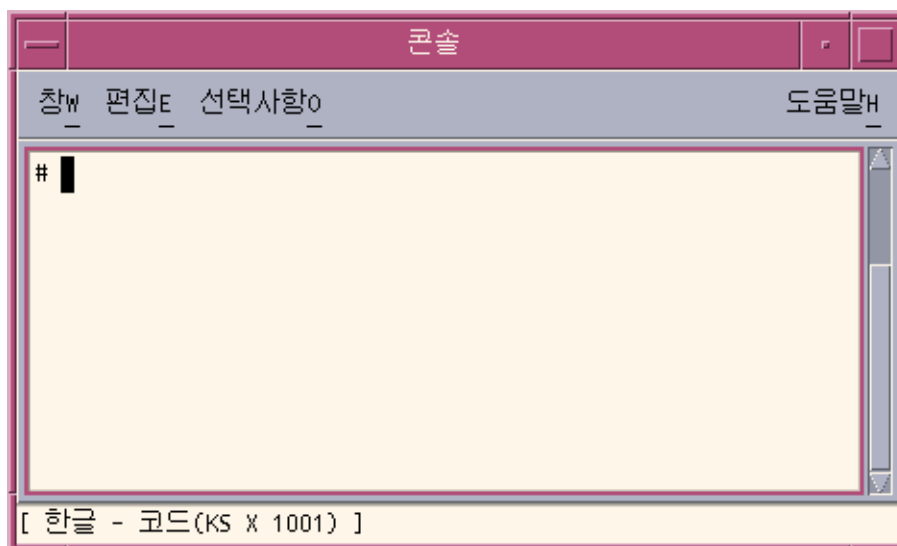
▼ How to Use the Hex Input Mode

1. Press Control-spacebar to turn on Hangul input conversion.
2. Press Control-X to turn on hex mode in the ko_KR.EUC locale or the ko_KR.UTF-8 locale.

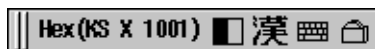
Alternatively, you can select the hex input mode from the palette auxiliary window.



The status area indicates that you are in hex input mode and that you can use KS X encoding values to input Hangul characters.



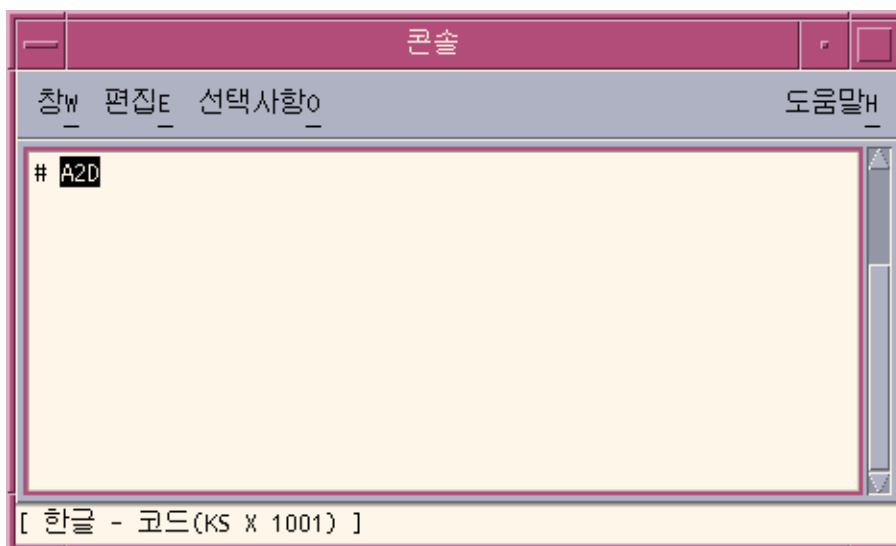
The palette window changes to indicate that you are in hex mode.



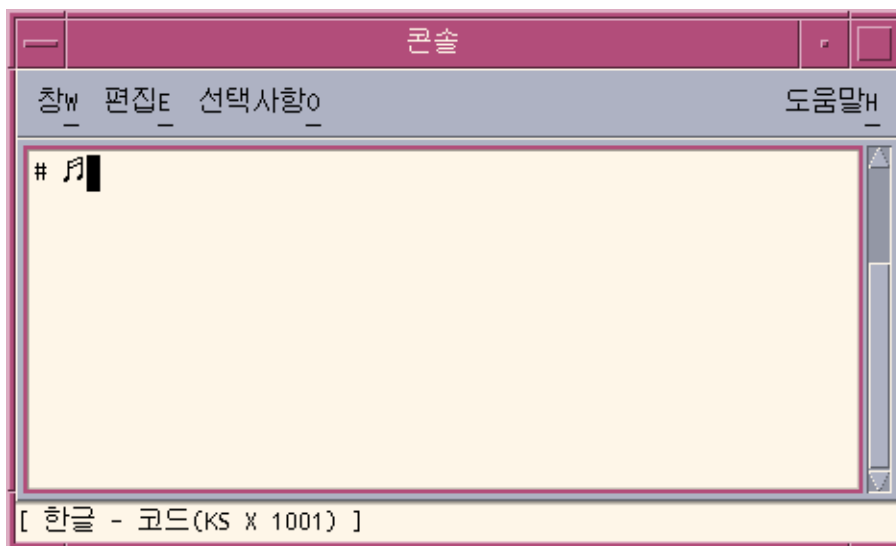
The only keys that are available in hex mode are the letters a through f, the numbers 0 through 9, and the Delete key. You can also use Control-X to exit hex mode.

Note – The Control-X option appears in this procedure to access the KS X 1001 character set for example purposes. You can press Control-Shift-X to enter a hex code from the UTF-8 character set.

3. Type the first three keys of the hex code for the character. For example, you could type a2d in the preedit area as shown in the following figure.



4. Type the last key of the hex code to display and to automatically commit the character. For example, if you type d in the preedit area, the glyph for two notes of music appears.



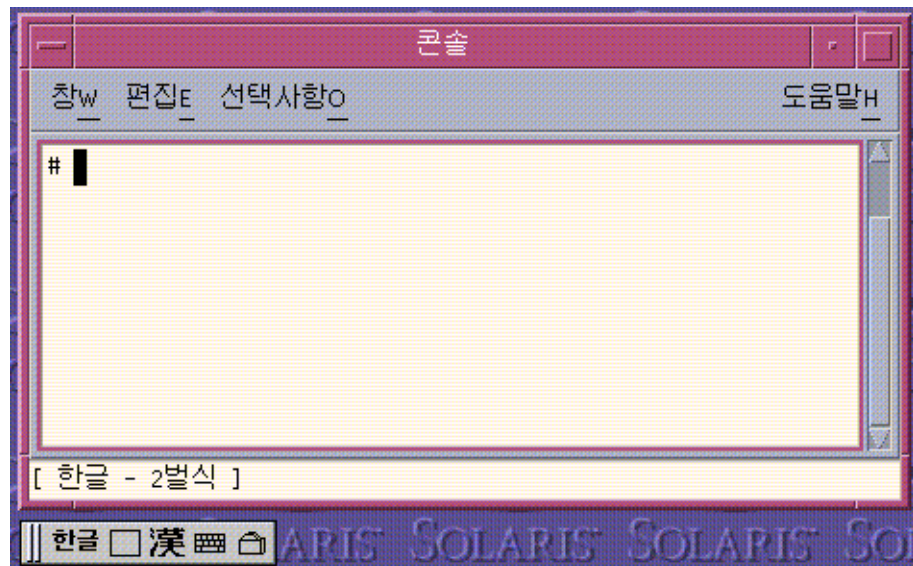
5. Press Control-X to turn off hex mode.

Switching Between the Half_Width and the Full_Width Character Modes

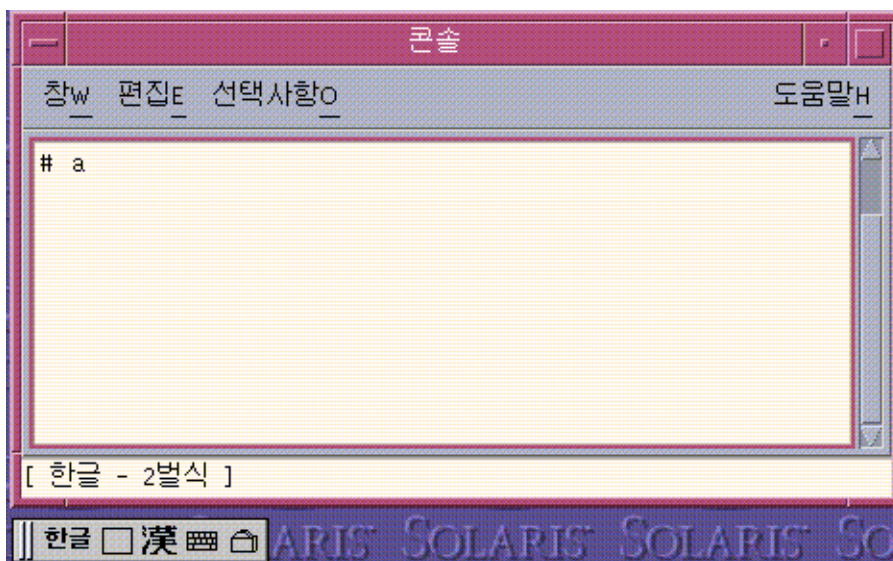
You can toggle between full_width and half_width character modes in the Korean Solaris environment. In full_width mode, the full_width character of the input key is committed in the application subwindow. In half_width mode, the half_width character is committed.

▼ How to Switch Between Half_Width and Full_Width Character Modes

- In Korean input mode, you can press Shift-spacebar to switch between the half_width and the full_width character modes.



For example, if you press the a key in full_width mode, the full_width form of the a character is committed.



Auxiliary Windows for the Korean Input Method

This section describes the four auxiliary windows available for the Korean input method.

Palette Auxiliary Window

The following buttons are available on the Palette Auxiliary window:

- Grabber – You click the left mouse button to move the palette from the grabber bar on the left.
- Input mode selector – Click this button to display a submenu with the following input mode options.
 - Normal Hangul input mode
 - Hex input mode (in KS X 1001)
 - Symbol input mode (in UTF-8)
 - Symbol input mode
- Half_width/Full_width mode selector. – Click this button to toggle between the half_width and the full_width input modes.

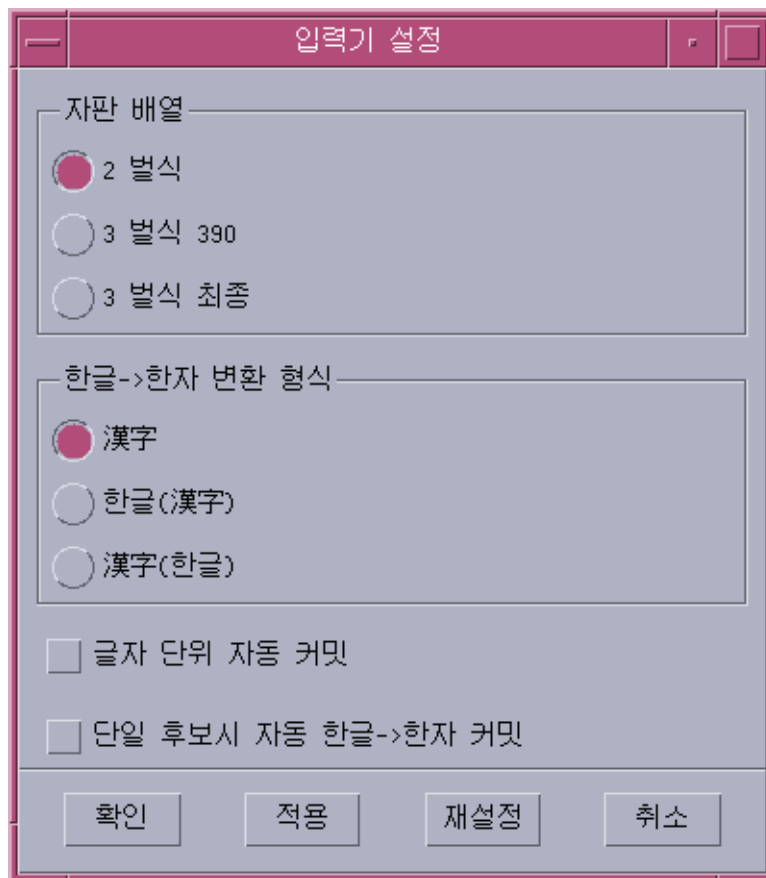
- Hangul to Hanja convertor – Click this button convert the Hangul characters in the preedit area to Hanja.
- Virtual Keyboard launcher – Click this button to show the virtual keyboard window.
- Utilities – Click this button to activate the Input Method Options Auxiliary window.

Input Method Options

Use the Options Auxiliary window to save your preference settings and to make the preferences available by default. To launch this window, click the Utilities button on the Palette Auxiliary window and select the Input Method Property Configuration menu.



The Options Auxiliary window appears.



You can specify the following options the Options Auxiliary window:

- Default Keyboard Layout
 - 2 Beol Sik
 - 3 Beol Sik 390
 - 3 Beol Sik final
- Default Hangul to Hanja conversion format
 - Hangul to Hanja
 - Hanja to Hangul (Hanja)
 - Hanja to Hangul (Hangul)
- Automatic commit on a character basis

The option enables you to commit a character in the preedit buffer as soon as you completely enter the character.
- Automatic commit on a single-candidate conversion

This option enables you to automatically commit Hanja when only a single candidate is available for the characters you want to convert.

Virtual Keyboard Auxiliary Window

To launch this window, click the Virtual Keyboard launcher in the Palette Auxiliary window. The auxiliary window appears for one of the following three virtual keyboards that you have selected as your default.

- 2 Beol Sik



- 3 Beol Sik 390



- 3 Beol Sik Final



Lookup Table Auxiliary Window

To launch this window, click the Utilities button in the Palette Auxiliary window and select the Code Point Table menu.



The Lookup Table Auxiliary Window appears.



To input any character of a table, double click the character or select the character and press the Insert button on the right side of the window.

Input Method Help

To launch this window, click the Utilities button in the Palette Auxiliary window and select the Input Method Help item. Information on the Korean input methods will appear in your web browser.

Hanja Tool

The following sections describe the features of the Hanja tool and provide procedures that you can use to create and edit local Hangul-Hanja conversion dictionaries.

- “Hanja Tool Features” on page 63
- “Using Hanja Tool” on page 64

Hanja Tool Features

The Hanja Tool enables you to add or delete entries in a local Hangul-Hanja dictionary. This capability controls the available Hanja choices in the Hangul-Hanja conversion mode.

Note – The system-wide Hangul-Hanja dictionary is read-only and cannot be edited.

The Hanja Tool is a viewer for the Hangul-Hanja dictionary. Because the dictionary file is in binary format, you would be unable to view it without the Hanja Tool.

Hanja Tool allows you to add an user-based Hangul-Hanja entry. The Hanja Tool adds the user-based Hangul-Hanja entry as an extension to the read-only, system-wide Hangul-Hanja dictionary, `/usr/lib/im/locale/ko_KR/common/data/syshjd`. Each user’s entry for Hangul-Hanja conversion works seamlessly together with the system-wide Hangul-Hanja entry.

Using Hanja Tool

The Hanja Tool initially appears in ASCII input mode, as shown in the status area of the figure. Before you can type Korean characters in the Hanja Tool fields, you must type Control-Space to turn on the Hangul input mode.

The Hanja Tool input fields, scrolling list areas, and buttons are labelled and used as described in the following.

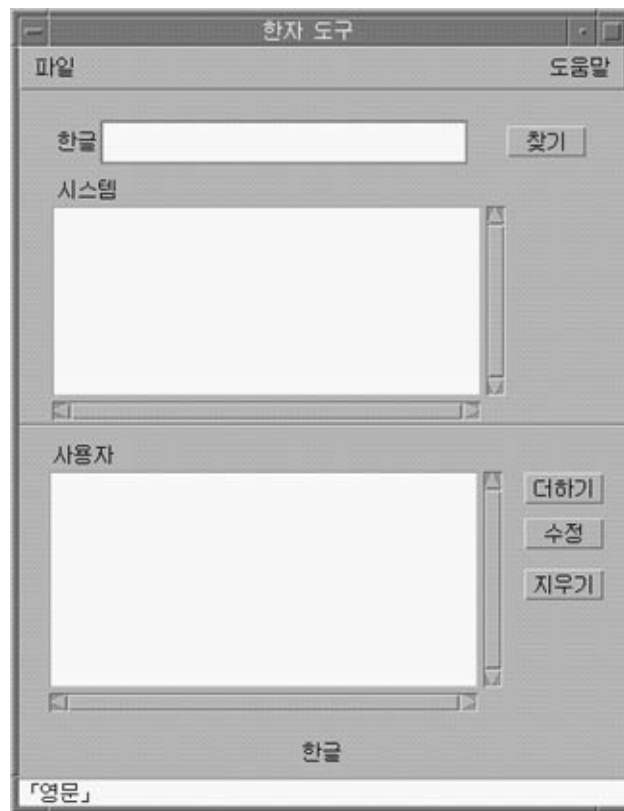
- **Hangul** – In this field you type the Hangul associated with certain Hanja characters in the Hangul–Hanja dictionaries.
- **System Dictionary** – This scrolling-list area displays the system-wide Hangul–Hanja dictionary entries. When you click the Find button, this area displays the System Dictionary’s Hanja entries associated with the Hangul characters committed in the Hangul field. You can scroll through the available Hanja choices for the Hangul characters or you can scroll through the entire System Dictionary.
- **User Dictionary** – This scrolling-list area displays the Hanja entries defined by a user. When you click the Find button, this area displays the user-defined Hanja entries associated with the Hangul in the Hangul field. You can scroll through the available Hanja choices for the Hangul characters.
- **Find** – Click this button to search the Hangul–Hanja dictionary for Hanja associated with the Hangul entry committed in the Hangul field.
- **Add** – Click this button bring up the Add/Modify dialog box. You can use this box to add a Hangul-Hanja conversion to the User Dictionary.
- **Modify** – Click this button to modify a Hangul-Hanja conversion in the User Dictionary. The Add/Modify dialog box appears only when the Hanja you select is from the User Dictionary. Otherwise, a system message is displayed.
- **Delete** – Click this button to delete the selected Hanja from the User Dictionary. A delete confirmation dialog box is displayed before the Hanja is deleted.
- **Quit** – Click this button to quit the Hanja Tool.

▼ How to Launch the Hanja Tool

- **Enter the following command to launch the Hanja Tool.**

```
system% sdthanja
```

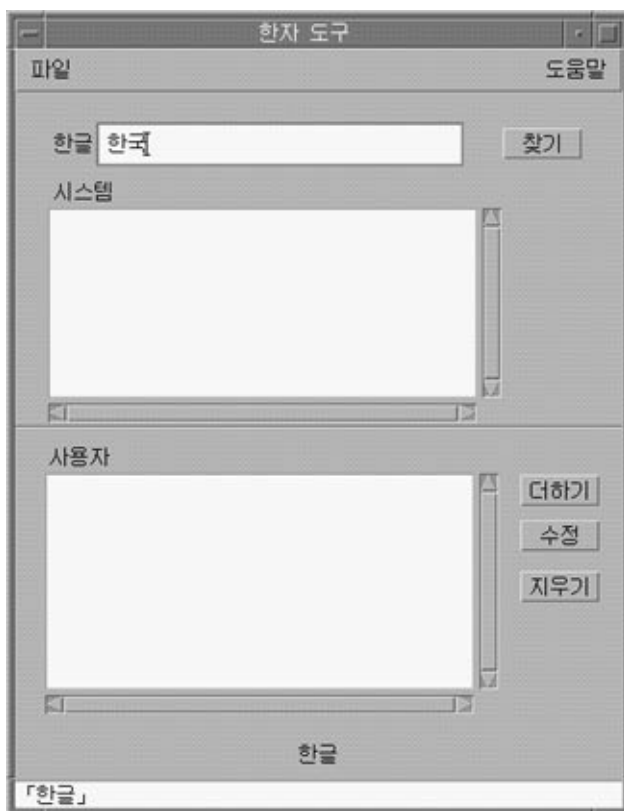
This command displays a Hanja Tool window, such as the following.



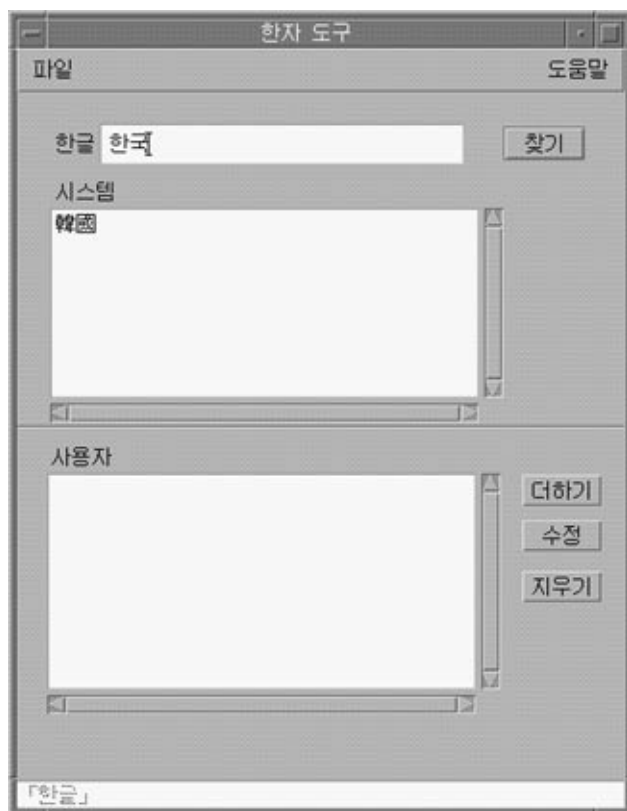
▼ How to Find the Hanja for a Hangul word

Use the following procedure to find the Hanja associated with a Hangul word in the system and user parts of the Hangul-Hanja dictionary.

1. **Use the Launch the Hanja Tool.**
2. **Type the Hangul word in the Hangul field.**
The entry appears highlighted in the preedit area as you type
3. **Type Control-K to commit the entry.**



4. Click the Find button to search for the Hanja to associate with the committed Hangul.



The line beneath the status area shows the number of the word in the User and System dictionaries.

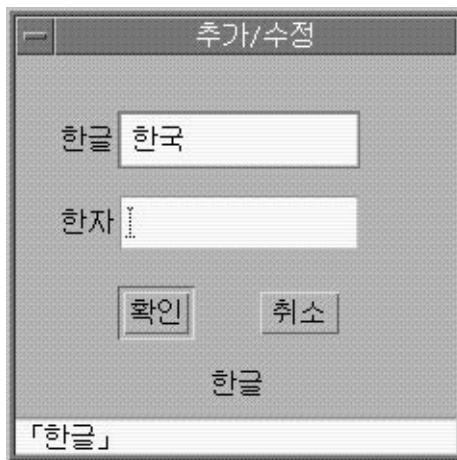
▼ How to Add Hanja to the Hangul-Hanja Dictionary

Use the following procedure to add a Hanja word to the Hangul-Hanja dictionary.

1. Click the Add button of the main screen to bring up the Add/Modify dialog box.



2. Type a new word or change the existing word in the Add/Modify dialog box. Then use Control-K to commit the Hangul characters you want to associate with the Hanja in your dictionary.



3. In the Hanja field, type the Hanja associated with the Hangul that you want to add to your dictionary.
Use the character-by-character methods described in [“Using Hangul/Hanja Conversion to Enter Hanja”](#) on page 41.



4. Click the Add button.

A display such as the following appears.



The line beneath the status area tells you the word was added to the User Dictionary. The User Dictionary is changed. The read-only System Dictionary cannot be changed.



▼ How to Modify Hanja in the Hangul-Hanja Dictionary

Modifying a Hanja word is similar to the process for adding a Hanja word.

1. Select a Hanja word from the User Dictionary and click the Modify button.
2. Edit the Hangul associated with the Hanja in the Add/Modify dialog box. Then use Control-K to commit the Hangul characters you want to associate with the Hanja.
3. In the Hanja field, type the Hanja associated with the modified Hangul.
Use the character-by-character methods described in [“Using Hangul/Hanja Conversion to Enter Hanja”](#) on page 41.
4. Click the Modify button.

▼ How to Delete Hanja from the Hangeul-Hanja Dictionary

To delete Hanja from the User Dictionary, use the following procedure.

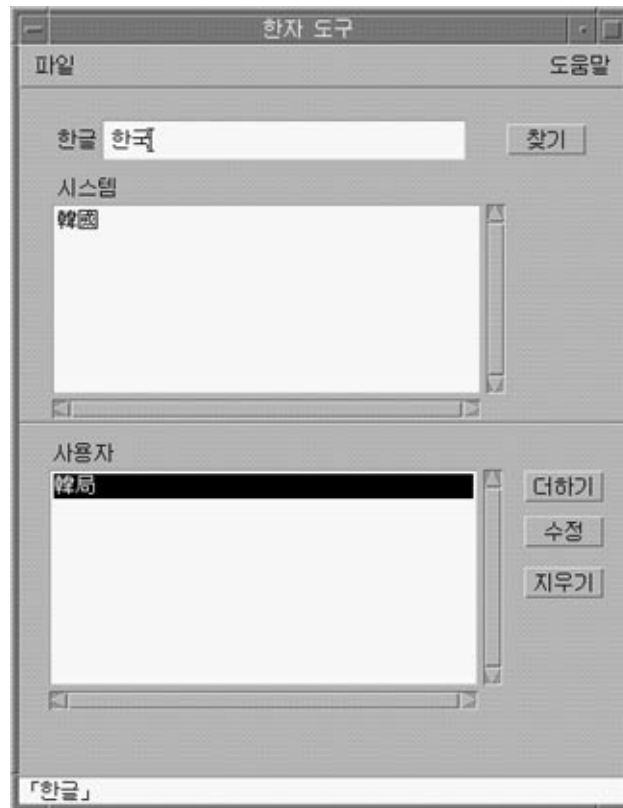
1. In the Hangeul field type the Hangeul word associated with the Hanja in your User Dictionary.

2. Type Control-K to commit the Hangeul input.

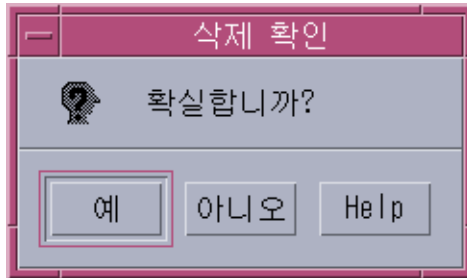
3. Click the Find button.

This displays the Hanja in the User Dictionary list area. Scroll through the list area to find the Hanja you want to delete.

4. Click the Hanja you want to delete.



5. Click the Delete button to delete the Hanja from the User Dictionary.
A delete confirmation dialog box appears.



6. **Click OK if you are sure you want to delete the Hanja. Otherwise, click Cancel.**

The line beneath the status area tells you that the word was deleted from the User Dictionary. The User Dictionary is changed. The System Dictionary is read-only and cannot be changed.

Localized Applications

The following sections in this chapter describe the utilities and applications that you use in the Korean Solaris Operating System.

- “Using the `mailx` Utility with Korean Characters” on page 73
- “Using `talk` with Korean Characters” on page 74
- “The `xtobdf` Utility” on page 74
- “The `Sdtconvtool` Utility” on page 74
- “The `iconv` Utility” on page 75

Using the `mailx` Utility with Korean Characters

The *encoding* variable in `mailrc` does not affect Mailer behavior, but it does set `mailx` function. To send email in formats other than 7-bit ASCII, such as to send (8-bit) Korean characters you must use `/usr/SUNWale/bin/mailx`. The *encoding* variable in `.mailrc` sets encoding formats for `mailx`.

International transmission conventions require that header information use only ASCII characters. Korean characters should not be used in the header (including the Subject line) with Mailer or with `mailx`.

Note – The `/bin/mailx` application has not been localized and cannot send or receive Korean characters.

Using `talk` with Korean Characters

To use Korean characters with `talk`, the `/usr/SUNWale/bin/talk` application is required because `/bin/talk` is not localized.

The `xtobdf` Utility

Korean Solaris software provides the following BDF font generator:

- `xtobdf` — Converts from an X server font to a BDF font
-

The `Sdtconvtool` Utility

`Sdtconvtool` is a graphic user interface utility that enables file conversion between various code sets. Its functionality is similar to `iconv`.

▼ How to Use the `Sdtconvtool` Utility

The following procedure shows how to convert a file encoded in UTF-8 to `ko_KR.EUC` encoding.

1. Select the code set of the file to be converted.

Scroll through the pull-down list and select the code set of the file to be converted. In this case, select UTF-8.

2. Enter the path of the file to be converted.

You can enter the path manually in the source file path area, or you can use the browse button to find and select the file. In this case. For example, you could enter or select `/tmp/ko_KR.UTF-8`.

3. Select the code set to for the converted file.

Select the target code set. For example, you could select `ko_KR-euc`.

4. Enter the path for the target file.

You can enter the path manually in the target file path area, or you can use the browse button to find and select the file. For example, you could enter or select

/tmp/ko-euc.txt.

5. Click the start conversion button.

The `iconv` Utility

The `iconv` command converts the characters or sequences of characters in a file from one code set to another and writes the results to standard output. Korean Solaris software includes special filters for the `iconv` command.

If no conversion exists for a particular character, it is converted to the underscore `_` in the target code set. The following options are supported:

- `-f from-code` - Symbol of the input code set
- `-t to-code` - Symbol of the output code set

The following code set conversion modules are supported in Korean Solaris software. For more information, see `iconv(1)`.

TABLE 6-1 Korean `iconv` Code Conversion Modules (ko locale)

Easy Name <i>from-code</i>	Actual Name <i>from-code</i>	Easy Name <i>to-code</i>	Actual Name <i>to-code</i>
IBM CP933	ko_KR.cp933	UTF-8 (Unicode 3.2)	ko_KR-UTF-8
IBM CP933	ko_KR.cp933	UTF-8	UTF-8
ISO646	646	KS X 1001	5601
ISO2022-KR	ko_KR.iso2022-7	KS X 1001	ko_KR-euc
ISO2022-KR	ko_KR.iso2022-7	UTF-8 (Unicode 3.2)	ko_KR-UTF-8
ISO2022-KR	ko_KR.iso2022-7	UTF-8	UTF-8
KS X 1001	5601	UTF-8	UTF-8
KS X 1001	EUC-KR	UTF-8	UTF-8
KS X 1001	KSC5601	UTF-8	UTF-8
KS X 1001	ko_KR-euc	UTF-8 (Unicode 3.2)	ko_KR-UTF-8
KS X 1001	ko_KR-euc	UTF-8 (Unicode 3.2)	ko_KR-UTF-8
KS X 1001	ko_KR.EUC	UTF-8	UTF-8
KS X 1001	ko_KR-euc	KS X 1001	ko_KR-johap
KS X 1001	ko_KR-euc	KS X 1001	ko_KR-johap92

TABLE 6-1 Korean `iconv` Code Conversion Modules (ko locale) (Continued)

Easy Name <i>from-code</i>	Actual Name <i>from-code</i>	Easy Name <i>to-code</i>	Actual Name <i>to-code</i>
KS X 1001	ko_KR-euc	KS X 1001	ko_KR-nbyte
KS X 1001	ko_KR-nbyte	KS X 1001	ko_KR-euc
KS X 1001	ko_KR-johap92	UTF-8 (Unicode 3.2)	ko_KR-UTF-8
KS X 1001	ko_KR-johap92	KS X 1001	ko_KR-euc
UTF-8	UTF-8	KS X 1001	5601
UTF-8	UTF-8	KS X 1001	KS X 1001
UTF-8	UTF-8	KS X 1001	EUC-KR
UTF-8	UTF-8	KS X 1001	KSC5601
UTF-8	ko-KR-UTF-8	IBM CP933	ko_KR.cp933
UTF-8	UTF-8	IBM CP933	ko_KR.cp933
UTF-8	ko-KR-UTF-8	ISO2022-KR	ko_KR-iso2022-7
UTF-8	UTF-8	ISO2022-KR	ko_KR-iso2022-7
UTF-8	UTF-8	ISO2022-KR	ISO2022-KR
UTF-8	ko-KR-UTF-8	KS X 1001	ko_KR-johap
UTF-8	UTF-8	KS X 1001	ko_KR-johap
UTF-8	ko-KR-UTF-8	KS X 1001	ko_KR-johap92
UTF-8	UTF-8	KS X 1001	ko_KR-johap92
Wansung	ko_KR-euc	Johap	ko_KR-johap92
Wansung	ko_KR-euc	Packed	ko_KR-johap
Wansung	ko_KR-euc	N-Byte	ko_KR-nbyte
ILEG=KS X 1001	ko_KR-euc	ISO-2022-KR	ko_KR-iso2022-7
Johap	ko_KR-johap92	Wansung	ko_KR-euc
Packed	ko_KR-johap	Wansung	ko_KR-euc
N-Byte	ko_KR-nbyte	Wansung	ko_KR-euc
ISO-2022-KR	ko_KR-iso2022-7	Wansung	ko_KR-euc

The following modules perform character-based code conversion on the KS C 5700 character set. They convert KSC 5700 characters between Korean UTF-8, completion code (Wansung), and combination code (Johap).

TABLE 6-2 Common Korean `iconv` Code Conversion Modules (`ko` and `ko.UTF-8` locales)

Easy Name <i>from-code</i>	Actual Name <i>from-code</i>	Easy Name <i>to-code</i>	Actual Name <i>to-code</i>
UTF-8	ko_KR-UTF-8	KS X 1001	ko_KR-euc
UTF-8	ko_KR-UTF-8	KS X 1001	ko_KR-euc
UTF-8	UTF-8	KS X 1001	ko_KR-euc
UTF-8	UTF-8	KS X 1001	ko_KR.EUC
UTF-8	ko_KR-UTF-8	Johap	ko_KR-johap92
UTF-8	ko_KR-UTF-8	Packed	ko_KR-johap
UTF-8	ko_KR-UTF-8	ISO-2022-KR	ko_KR-iso2022-7
Wansung	ko_KR-euc	UTF-8	ko_KR-UTF-8
Wansung	ko_KR-euc	UTF-8	UTF-8
Johap	ko_KR-johap92	UTF-8	ko_KR-UTF-8
Johap	ko_KR-johap92	UTF-8	UTF-8
Packed	ko_KR-johap	UTF-8	ko_KR-UTF-8
Packed	ko_KR-johap	UTF-8	UTF-8
ISO-2022-KR	ko_KR-iso2022-7	UTF-8	ko_KR-UTF-8
ISO-2022-KR	ko_KR-iso2022-7	UTF-8	UTF-8
UTF-8	ko_KR.UTF-8	Unified Hangul	ko_KR-cp949
UTF-8	UTF-8	Unified Hangul	ko_KR-cp949
Unified Hangul	ko_KR-cp949	UTF-8	ko_KR-UTF-8
Unified Hangul	ko_KR-cp949	UTF-8	UTF-8

In the following example, a KS C 5601 file (Korean EUC) is converted to KS C 5700 (`ko.UTF-8`).

```
system% iconv -f ko_KR-euc -t ko_KR-UTF-8 ko_euc_file > ko_UTF-8_file
```

For further information, see the `iconv(3C)`, `iconv_ko.UTF-8(5)`, and `iconv_utf(5)` man pages. You can use these utilities to convert files for printing.

Korean Solaris Printing

The Korean Solaris Operating System supports the printing of Korean output to the following types of printers.

- Line printer with built-in Korean fonts
- PostScript-based printer with built-in scalable fonts
- Any PostScript-based printer for bitmap printing

Note – Review the manufacturer’s documentation on installing the printer before you complete the procedures in this chapter.

Line Printer Support

For the Korean Solaris Operating System to run a line printer, the printer must recognize at least one of the supported code sets.

- Completion code, also called Wansung (Korean EUC, based on KS C 5601)
- Combination code, also called Johap (either KS C 5601-1987-3 or KS C 5601-1992-3)
- N-byte code

Printing an EUC (Wansung) File to a Printer that Does Not Support EUC

A printer that does not support EUC requires filters to convert the EUC files before you can print them. Use the commands in this section to print EUC files to a non-EUC printers.

The following command lines install the printer `lp1` on port `ttya` and signal the print service that the `lp1` accepts only Packed format files. Packed is the KS C 5601-1987 combination code.

```
# lpadmin -p lp1 -v /dev/ttys -I PACK
# accept lp1
# enable lp1
```

The following command lines install the printer `lp1` on port `ttya` and signal the print service that `lp1` accepts only Johap format files. Johap is the KS C 5601-1992 combination code.

```
# lpadmin -p lp1 -v /dev/ttys -I JOHAP
# accept lp1
# enable lp1
```

See the `lpadmin(1M)` man page for more information.

You can use an `lpfilter` command line such as the following to print files with formats that are not supported by the printer. The command line signals the print service that a converter called *filter-name* is available through the filter description file named *pathname*.

```
# lpfilter -f filter-name -F pathname
See the lpfilter(1M) for more information.
```

The following example shows the output of *pathname* for a converter called `comptopack`.

```
Input types: simple
Output types: PACK
Command: comptopack
```

The *pathname* filter converts the default input type to Packed format with the `comptopack` converter.

```
Input types: simple
Output types: JOHAP
Command: wansungtojohap
```

To print an EUC file, use a command line such as the following.

```
system% lp euc-filename
```

To print a Packed format file, use a command line such as the following.

```
system% lp -T PACK PACK-filename
```

To print a Johap format file, use a command line such as the following.

```
system% lp -T JOHAP JOHAP-filename
```


Printing a ko_KR.UTF-8 File to Printers that Do Not Support UTF-8

Use the commands in this section to convert ko_KR.UTF-8 files to EUC and Johap files.

You can use command lines such as the following to convert a ko_KR.UTF-8 file and to print the converted file to an EUC printer.

```
system% iconv -f ko_KR-UTF-8 -t ko_KR-euc
ko_KR.UTF-8_filename
>euc-filename
system% lp euc-filename
```

Note – The output will include only the characters that are defined in the EUC code set.

You can use command lines such as the following to convert an ko_KR.UTF-8 file and to print the converted file to a Johap printer.

```
system% iconv -f ko_KR-UTF-8 -t ko_KR-johap92
ko_KR.UTF-8_filename >
johap92-filename
system% lp johap92-filename
```

Note – The output will include only the characters that are defined in the Johap code set.

Laser Printer Support

To print Korean characters to a PostScript-based printer, a Korean Solaris application recognize the mp utility.

Using the mp Utility

The mp utility supports all Asian locales. As a print filter, mp generates a properly formatted version of content in PostScript format. The Postscript output contains scalable or bitmap fonts, depending on the system font configuration formp in the current locale. The mp filter can print a ko file in ko locale or a UTF-8 file in the ko_KR.UTF-8 locale. For more information, see the mp(1) man page.

The following example shows a typical command line that you could use to print a file named *filename* that contains Korean characters, with or without ASCII/English characters.

```
system% mp filename | lp -d printer
```

PART **III**

System Administrator Information

This part includes system administration information.

Configuring a Korean Solaris System

You can use the procedures this chapter to set a system-wide default locale or to add a new locale from the Solaris DVD and CD media. You also change the font path set by the `Xsession` script included in the Korean Solaris Operating System. The following sections are included in this chapter.

- “Setting the Default Locale” on page 85
- “Using Aliases to Set the Locale” on page 86

Setting the Default Locale

You can use the procedure in this section to make a system-wide change to the default locale.

▼ How to Set the Default Locale

1. **Become superuser.**
2. **Edit the `/etc/default/init` file.**
3. **Add or update the `LANG` variable with the appropriate locale.**

```
LANG=locale
```

```
locale
```

```
C, ko, or the ko.UTF-8
```

4. **Instruct all users to exit CDE.**
5. **Kill the `dtconfig` process.**

```
# /usr/dt/bin/dtconfig -kill
```

6. Reboot the system.

```
# reboot
```

Using Aliases to Set the Locale

You can use aliases to change a terminal-emulation window between the Korean locales and ASCII/English locale without typing long command lines or editing your `.dtlogin` file and running `source` every time.

▼ How to Set Locale Aliases for the Korn Shell

- Put aliases such as the following in your `.profile` file.

```
alias kor_EUC "export LANG=ko; /bin/stty cs8 -istrip defeucw"
alias kor_UTF "export LANG=ko.UTF-8; /bin/stty cs8 -istrip \defeucw"
alias asc "export LANG=C; /bin/stty cs7 istrip defeucw"
```

How to Set Locale Aliases for the C Shell

- Put aliases such as the following in your `.cshrc` file.

```
alias kor_EUC "setenv LANG ko; /bin/stty cs8 -istrip defeucw"
alias kor_UTF "setenv LANG ko.UTF-8; /bin/stty cs8 -istrip \defeucw"
alias asc "setenv LANG C; /bin/stty cs7 istrip defeucw"
```

Managing the `htt` Input Method Server

The input method server (IM Server) handles Korean input for Korean Solaris software. The IM Server receives keyboard input and makes the input available as Korean characters to Korean Solaris system applications. The IM Server can serve any internationalized X Window application that uses the X Windows Input Method (XIM) application program interfaces (API) to receive language input.

The following sections in this chapter explain the basic display features of the IM Server.

- “Input Method Server Basic Properties” on page 87
- “`htt` Application Display” on page 88
- “How and When `htt` is Started” on page 89
- “Using the `iiim` Server in the `ko_KR.UTF-8` Locales” on page 89

Input Method Server Basic Properties

The Korean Solaris Operating System starts the IM Server automatically when you log in to a locale that requires an input method.

The Korean Solaris applications that use the IM Server for Korean character input typically find the IM Server running when the applications starts. To connect to the IM Server for language input, the IM Server should be running before the application starts. If the IM Server is not running before the application starts, the application might not be able to connect to the input service. Even if the IM Server starts later, the application might not be able to connect to the input service.

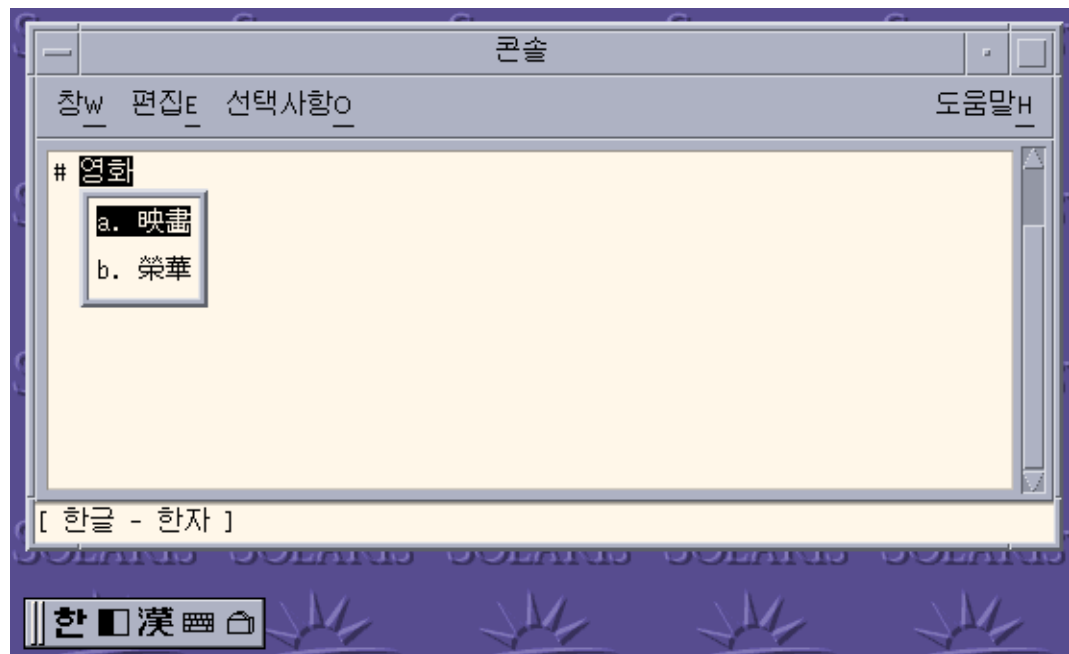
Three http Processes

The input method server comprises three related programs. The `http`, `http_server`, and `http_xbe`, processes are run together. One process controls the properties of the input method server. A watch dog process controls the population of the input method server and a third process handles the input methods of clients.

Usually, you don't need to know the details of these three processes. When `http` starts, the `http_props` and `http_server` start also.

http Application Display

The following figure shows a typical application display when the `http` input method is in use.



How and When `htt` is Started

Any application that uses the IM server to receive Korean characters must find `htt` running before the server can receive Korean character input. If the default locale is `C`, which does not require `htt`, and you change from `C` to another locale in a single Terminal, for example, you must start `htt` with a command line like the following:

```
system% /user/openwin/bin/htt -if kole -xim htt_xbe &
```

With the locale set to `ko_KR.EUC`, `htt` starts automatically when the Korean windowing environment is started. In CDE, `htt` is started from a script, `/usr/dt/config/Xsession.d/0020.dtims`. The script is executed by `Xsession`, which runs at every initialization stage when you log in from `dtlogin`. The script ensures that `htt` is started before other applications in CDE.

Using the `iiim` Server in the `ko_KR.UTF-8` Locales

The Internet Intranet Input Method Server (`iiim`) supports the European Local Input Methods that use the Compose key. The `iiim` server also supports the Asian Remote Input Methods that use IIIM Protocol. The default Language Engine is English/European which supports ASCII and some European Languages such as German and French.

To switch to Korean Input Methods, press Control-spacebar.

To switch to other language engines, click the left mouse button in the Status Area.

The following language engine list appears.

- Cyrillic
- Greek
- Thai
- Arabic
- Hebrew
- Hindi
- Unicode Hex
- Unicode Octal
- Lookup
- Korean

The following language engines appear if the corresponding locales are installed:

- Japanese — if the `ja` locale is installed
- Simplified Chinese — if the `zh` locale is installed
- Traditional Chinese — if the `zh_TW` locale is installed
- Traditional Chinese (Hong Kong) — if the `zh_HK` locale is installed

For more information about how to use Korean input methods, see [Chapter 4](#).

Fonts

The following sections in this chapter describe how to use and edit the Korean Solaris fonts.

- “Display PostScript System (DPS)” on page 91
- “BDF and PCF Formats” on page 95
- “Changing Font Directories” on page 96
- “Bitmap and TrueType Fonts” on page 96

This chapter also describes how to convert BDF font files to Portable Compiled Format (PCF) format and how to make the fonts usable on your system.

Display PostScript System (DPS)

The Korean Solaris Operating System provides PostScript fonts in the Display PostScript System (DPS). This section describes what you need to use DPS in Korean Solaris software. For further details, see *Programming the Display PostScript System with X*, published by Adobe Systems.

Using Korean PostScript Fonts and DPS Facilities

The Korean Solaris Operating System DPS provides the fonts listed in the following table.

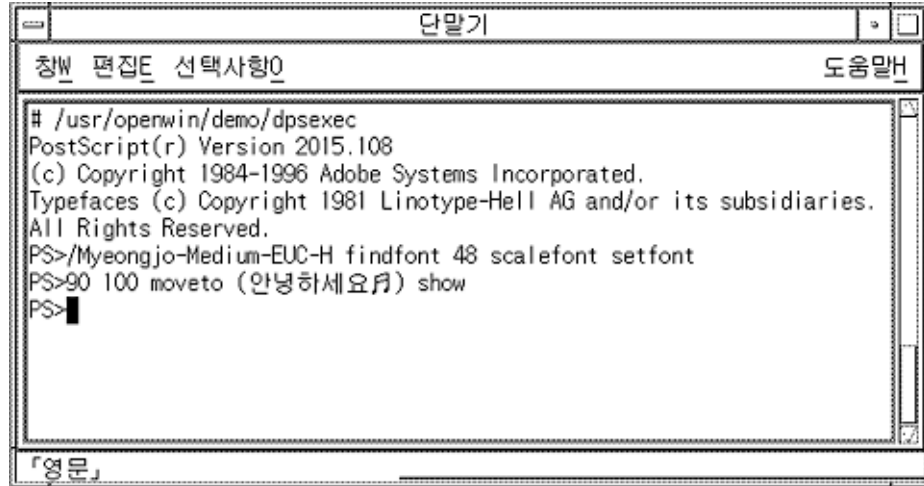
TABLE 10-1 Korean Solaris Operating System DPS Fonts

Font Name	Description
Kodig-Medium-COMB-H	Kodig-Medium font, 9/7 composite font encoding for horizontal display of Johap encoded Hangul and Roman text.
Kodig-Medium-COMB-V	Kodig-Medium font, 9/7 composite font encoding for vertical display of Johap encoded Hangul and Roman text.
Kodig-Medium	Kodig-Medium font, an alias of Kodig-Medium-EUC-H font; can be used like a Roman font.
Kodig-Medium-EUC-H	Kodig-Medium font, 9/7 composite font encoding for horizontal display of EUC text, can be used like a Roman font.
Kodig-Medium-EUC-V	Kodig-Medium font, 9/7 composite font encoding for vertical display of EUC text; can be used like a Roman font.
Kodig-Medium-H	Kodig-Medium font, 8/8 composite font encoding for horizontal display of shifted out ISO2022 text.
Kodig-Medium-V	Kodig-Medium font, 8/8 composite font encoding for vertical display of shifted out ISO2022 text.
Myeongjo-Medium-COMB-H	Myeongjo-Medium font, 9/7 composite font encoding for horizontal display of Johap encoded Hangul and Roman text.
Myeongjo-Medium-COMB-V	Myeongjo-Medium font, 9/7 composite font encoding for vertical display of Johap encoded Hangul and Roman text.
Myeongjo-Medium	Myeongjo-Medium font, an alias of Myeongjo-Medium-EUC-H font; can be used like a Roman font.
Myeongjo-Medium-EUC-H	Myeongjo-Medium font, 9/7 composite font encoding for horizontal display of EUC text; can be used like a Roman font.
Myeongjo-Medium-EUC-V	Myeongjo-Medium font, 9/7 composite font encoding for vertical display of EUC text; can be used like a Roman font.
Myeongjo-Medium-H	Myeongjo-Medium font, 8/8 composite font encoding for horizontal display of shifted out ISO2022 text.
Myeongjo-Medium-V	Myeongjo-Medium font, 8/8 composite font encoding for vertical display of shifted out ISO2022 text.

You can use the following Korean fonts just as you would use Roman fonts.

- Kodig-Medium-EUC-H
- Kodig-Medium-EUC-V
- Kodig-Medium
- Myeongjo-Medium-EUC-H
- Myeongjo-Medium-EUC-V
- Myeongjo-Medium

The following figure shows a sample of Kodig-Medium and Myeongjo-Medium text.



```
# /usr/openwin/demo/dpsexec
PostScript(r) Version 2015.108
(c) Copyright 1984-1996 Adobe Systems Incorporated.
Typefaces (c) Copyright 1981 Linotype-Hell AG and/or its subsidiaries.
All Rights Reserved.
PS>/Myeongjo-Medium-EUC-H findfont 48 scalefont setfont
PS>90 100 moveto (안녕하세요) show
PS>
```



You can also use the following Korean fonts for pure Hangul between the SO and SI characters with no intermediate ASCII (0x20) characters.

- Kodig-Medium-H
- Kodig-Medium-V

Creating Composite Korean Fonts

You can create composite fonts using one Roman font and one of the following Korean fonts.

- Kodig-Medium-COMB-H
- Kodig-Medium-COMB-V
- Myeongjo-Medium-COMB-H
- Myeongjo-Medium-COMB-V

For example, the following PostScript code defines a sample composite font. The ASCII characters of the Times-Italic+Kodig-Medium font are combined with the Korean characters of the Kodig-Medium horizontal font.

```
/Times-Italic+Kodig-Medium
13 dict begin
    /FontName 1 index def
    /FMapType 4 def
    /Encoding [ 0 1 ] def
    /WMode 0 def
    /FontType 0 def
    /FontMatrix [1.0 0.0 0.0 1.0 0.0 0.0] def
    /FDepVector [
        /Times-Italic findfont
        /Kodig-Medium-COMB-H findfont
    ] def
currentdict
end
definefont pop
```

Using Korean Fonts in DPS Programming

You can use Korean fonts just as you use Roman fonts in DPS wrap definitions. This sample code creates the display which follows.

```
defines PSWDisplayText(char *text)
    /pointSize 50 def
    /Helvetica pointSize selectfont
    (Hello World) stringwidth pop 2 div neg 0 moveto
    (Hello World) show

    /cpSize 40 def
    /Kodig-Medium-KO cpSize selectfont
    (text) stringwidth pop 2 div neg pointSize neg moveto
    (text) show
endps
```

You can set PSWDisplayText in a C program to display the designated Korean text.



BDF and PCF Formats

You must convert BDF fonts to PCF format to make the fonts usable in Korean Solaris applications.

▼ How to Convert BDF Fonts to PCF Format

- Use the `bdf2pcf` command to convert a BDF font to PCF format.

```
system% bdf2pcf -o myfont14.pcf myfont14.bdf
```

The `-o` option enables the matrix encoding used for Asian PCF font files.

▼ How to Install and Check an Edited Font

1. Compress the `.pcf` font file and move it to your font directory.

```
system% compress myfont14.pcf
```

2. Run the following commands in your font directory.

The `.bdf` file should not be in the font directory.

```
system% cat >> fonts.alias
-new-myfont-medium-r-normal--16-140-75-75-c-140-ksc5601.1987-0
Myfont-Medium14
```

```
^D
system% mkfontdir
system% xset +fp `pwd`
```

3. Use a command line such as the following to view your font.

```
system% xfd -fn Myfont-Medium14
```

Changing Font Directories

The Xsession script in `/usr/dt/bin` directory of the Korean Solaris Operating System includes the following font path:

`/usr/openwin/lib/locale/locale/X11/fonts`, where *locale* is either `ko` or `ko.UTF-8`. To dynamically add a different font directory path, type the following commands:

```
system% xset +fp font_directory-path
system% xset fp rehash
```

Bitmap and TrueType Fonts

The tables in this section list the bitmap and TrueType fonts that are supported for the `ko`, the `ko.UTF-8`, and the `ko/ko.UTF-8` locales.

The following table shows the Korean bitmap fonts for the `ko` locale.

TABLE 10-2 Korean Bitmap Fonts for the `ko` Locale

Full Family Name	Subfamily	Format	Encoding
Gothic	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Graphic	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Haeso	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Kodig	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Myeongijo	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Pilki	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Round gothic	R/B	PCF (12,14,16,18,20,24)	KS X 1001

The following table shows the Korean bitmap fonts for the `ko.UTF-8` locale.

TABLE 10-3 Korean Bitmap Fonts for the ko.UTF-8 Locale

Full Family Name	Subfamily	Format	Encoding
Gothic	R/B	PCF (12,14,16,18,20,24)	KS X 1001 (Johap)
Graphic	R/B	PCF (12,14,16,18,20,24)	KS X 1001 (Johap)
Haeso	R/B	PCF (12,14,16,18,20,24)	KS X 1001 (Johap)
Kodig	R/B	PCF (12,14,16,18,20,24)	KS X 1001 (Johap)
Myeongijo	R/B	PCF (12,14,16,18,20,24)	KS X 1001 (Johap)
Pilki	R/B	PCF (12,14,16,18,20,24)	KS X 1001 (Johap)

The following table shows the Korean TrueType Fonts for the ko/ko.UTF-8 locales.

TABLE 10-4 Korean TrueType Fonts for the ko/ko.UTF-8 Locales

Full Family Name	Subfamily	Format	Vendor	Encoding
Kodig/Gothic	R	TrueType	Hanyang	Unicode
Myeongijo	R	TrueType	Hanyang	Unicode
Haeso	R	TrueType	Hanyang	Unicode
Round gothic	R	TrueType	Hanyang	Unicode

Korean Printing Facilities

This chapter includes the following sections:

- “Printing Korean Output to a Line Printer” on page 99
- “Using the mp Utility” on page 100

Printing Korean Output to a Line Printer

You can use the commands in this section to print files of different formats to a line printer.

- **Use this command to print an EUC file.**

```
system% lp euc-filename
```

- **Use this command to print a Packed format file to a printer that supports Packed format.**

```
system% lp -T PACK PACK-filename
```

- **Use this command to print a Johap format file to a printer that supports Johap format.**

```
system% lp -T JOHAP JOHAP-filename
```

- **Use commands such as the following to print a ko.UTF-8 file to an EUC printer.**

The first command line converts the file to an EUC format. The second line prints the file.

```
system% iconv -f ko_KR-UTF-8 -t ko_KR-euc  
ko.UTF-8_filename  
> euc-filename  
system% lp euc-filename
```

Note – The printed output includes only the characters that are defined in the EUC code set.

- **Use commands such as the following to print a ko.UTF-8 file to a Johap (KS C 5601-1992) printer.**

The first command line converts the file to a Johap format. The second line prints the file.

```
system% iconv -f ko_KR-UTF-8 -t ko_KR-johap92ko.UTF-8_filename >\  
johap92-filename  
system% lp johap92-filename
```

Note – The printed output includes only the characters that are defined in the Johap code set.

Using the mp Utility

The enhanced mp print utility is available in the current Korean Solaris release. You can use mp to print ko files in ko locale or UTF-8 files in ko.UTF-8 locales.

The output from the utility is standard PostScript that you can send to any PostScript printer.

Use the following command to print a file with the mp utility.

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

You can also use the utility as a filter because mp accepts stdin stream.

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

You can set the utility as a print filter for a line printer. For example, the following command sequence signals the print service that the printer lp1 accepts only mp format files. The command also installs the printer lp1 on port /dev/ttya. See the lpadmin(1M) man page for more details.

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

You can use the following command to add a filter for an LP print service.

```
system% lpfilter -f filtername -F pathname
```

The filter converts the default input type to PostScript output with the /usr/bin/mp command.

To print a UTF-8 text file, use a command line such as the following.

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

For more details on the `mp(1)`, see the man page.

Using `mp` as an Xprt (X Print Server) Client

Using `mp` as an Xprt client enables `mp` to print the output of any printer connected to a network supported by an Xprt print service. As an Xprt client, `mp` supports PostScript and many versions of PCL.

The Xprt client attempts a connection to an Xprt server based on the following rules:

- When the `-D printer_name@machine[:dispNum]` or `-P printer_name@machine[:dispNum]` options are used with the `mp` command, `mp` attempts to connect to an Xprt print service on `machine[:dispNum]` with `printer_name`.

If the above attempted connection to `machine[:dispNum]` fails or if the argument given to `-D` or `-P` is just `printer_name`, then the `mp` command checks the `XPSEVERLIST` for Xprt servers that support the `printer_name` argument. For example:

```
system% setenv XPSEVERLIST "machine1[:dispNum1] machine2[:dispNum2] machine3[:dispNum3]"
```

- If no server is found using above rules, `mp` checks for an `XPDISPLAY` environment variable set to `machine[:dispNum]`. For example:

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

- If the `XPDISPLAY` variable is not set or if the variable is invalid, `mp` tries to connect to the default display `:2100`. If the default display value is also invalid, `mp` exits with an error message.

The `/usr/lib/lp/locale/C/mp` directory contains `.xpr` print page sample layout files for Xprt client. The sample files are for 300 dpi printers. If the target printer has a different dpi value, the dpi value of the sample files is automatically converted to the resolution of the target printer.

TTY Environment and Support

This chapter assumes that you are familiar with the following information.

- The Solaris Operating System communicates with external devices using STREAMS and `ioctl`.
- Different terminal types are supported by `termcap` and `terminfo`.

This chapter includes the following sections:

- “TTY STREAMS” on page 103
- “TTY Utilities” on page 105
- “TTY Commands” on page 106
- “TTY Setup Examples” on page 107
- “Terminal Support” on page 110
- “Installing a Terminal” on page 110

Refer to the `termio(7I)` man page for background information on STREAMS and TTY drivers.

TTY STREAMS

The data path between a user’s shell and the terminal is called a *stream*. The data on a stream contain characters and control information that affect data handling, such as the control sequences that precede a change in code set or communication protocols. Data that enters the stream from the terminal is raw or unprocessed. Data is sequentially processed by STREAMS modules for appropriate use by the shell or an application.

STREAMS provides a way to modularize the processing on a line, allowing processing instructions to be grouped in functional modules. These modules can be added or removed from the line so that different environments can be provided to a terminal according to the user’s needs.

Traditional STREAMS

The traditional STREAMS TTY environment contains a raw device driver, a line discipline module, and a stream head. The raw device driver provides an I/O interface between the kernel and the hardware. Because it is closest to the physical hardware, it provides basic communication protocols, baud rate switching, and other low level services. The line discipline module is a set of instructions or disciplines that transforms the raw data to processed data. This includes handling the delete character, line kill character, and others. The stream head provides an interface between the user's process and the stream.

Korean Solaris STREAMS

The Korean Solaris Operating System uses the modular nature of STREAMS to support Korean. In addition to the traditional TTY modules, this product implements code conversion in STREAMS. Hangul-Hanja conversion is typically supplied by many existing Korean TTYs and is not available in the Korean Solaris TTY environment.

The Korean Solaris Operating System enhances the traditional modules. The line discipline of the operating system handles proper cursor movement for wide characters as well as normal protocols. The Korean Solaris code conversion modules convert between two different character code formats. For example, Packed format, combination code of KS C 5601-1987, can be converted to EUC format, completion code.

Code conversion depends on the appropriate flags or parameters being set. For example, if a Packed code terminal is being used, the input from the terminal is converted to EUC and the output to the terminal is converted to Packed code.

The major modules that can be pushed onto the stream are `ldterm`, `kpack` and `kjohap`.

- `ldterm` is a generic EUC line discipline module. It processes all normal line discipline functions and also handles proper cursor movement and backspacing for wide characters (EUC). See `ldterm(7M)` for more information.
- `kpack` controls code conversion between Combination code of KS C 5601-1987 and EUC.
- `kjohap` controls code conversion between Combination code of KS C 5601-1992 code and EUC.

TTY Utilities

`ioctl` (input/output control) calls are low-level routines for handling device input and output.

The `termcap` and `terminfo` databases are used by applications to configure their terminal display appropriately.

EUC `ioctl` Features

The Korean Solaris Operating System uses `ioctl` STREAMS commands for general EUC handling. The table describes the `ioctl` requests.

TABLE 12-1 `ioctl` Requests and Descriptions

<code>ioctl</code> Request	Description
<code>EUC_WGET</code>	Get <code>cswidth</code> values from TTY stream
<code>EUC_WSET</code>	Set <code>cswidth</code> values for TTY stream
<code>EUC_OXLOFF</code>	Set code conversion to OFF
<code>EUC_OXLON</code>	Set code conversion to ON

Character code conversion to and from the terminal is controlled by `EUC_OXLON` and `EUC_OXLOFF`.

The `termcap` database

The `termcap` and `terminfo` databases used to tailor the terminal characteristics for an application. The following table describes the variables of the `termcap` database.

TABLE 12-2 `termcap` Variables and Descriptions

Variable	Description
<code>dv</code>	Device type: language and code set
<code>ci</code>	Init sequence for multiple code sets
<code>s0</code>	Shift into code set 0
<code>s1</code>	Shift into code set 1

TABLE 12-2 `termcap` Variables and Descriptions (Continued)

Variable	Description
<code>s2</code>	Shift into code set 2
<code>s3</code>	Shift into code set 3

The `terminfo` database

The following are extensions to `terminfo`. The `s0-s3` string values are used as data announcement mechanisms for the respective code sets during terminal I/O.

TABLE 12-3 `terminfo` Variables and Definitions

Variable	Capname	Tc	Definition
<code>device_type</code>	<code>devt</code>	<code>dv</code>	Device type: language and code set
<code>code_set_init</code>	<code>csin</code>	<code>ci</code>	Init sequence for multiple code sets
<code>set0_des_set</code>	<code>s0ds</code>	<code>s0</code>	Shift into code set 0
<code>set1_des_set</code>	<code>s1ds</code>	<code>s1</code>	Shift into code set 1
<code>set2_des_set</code>	<code>s2ds</code>	<code>s2</code>	Shift into code set 2
<code>set3_des_set</code>	<code>s3ds</code>	<code>s3</code>	Shift into code set 3

TTY Commands

The two commands for configuring and using the TTY environment are `setterm` and `/bin/stty`. `setterm` is used primarily to build the TTY stream for a particular terminal type, pushing the necessary modules onto the stream. `stty` changes the behavior of the modules in the stream.

`setterm` Command

`setterm` is used to configure the TTY STREAMS environment. It can inquire about and manipulate STREAMS modules for a particular TTY port. `setterm` allows users to tailor their TTY STREAMS environment using system-provided or user-provided STREAMS modules.

`setterm` uses a terminal device name that reflects the `devt` (device type) field in the `terminfo` database for configuring STREAMS modules for a TTY port. This device name is matched with an entry of the same name in the `setterm` configuration file,

`/usr/share/lib/setterm/ko/conf.file` or `/usr/share/lib/setterm/ko.UTF-8/conf.file`. This entry contains detailed instructions on which modules to pop and push in order to properly configure the STREAMS environment.

`setterm` can also take the device type as a direct argument. This device type is similarly matched with an entry in `/usr/share/lib/setterm/ko/conf.file` or `/usr/share/lib/setterm/ko.UTF-8/conf.file`.

The `setterm` configuration file uses a special language for instructions on what actions to take. This language allows users to determine the names of modules on the STREAMS stack, to push or pop modules on the stack, and to do other operations. `setterm` manipulates the STREAMS stack by making `ioctl` calls.

`/bin/stty` Command

The `defeucw` option to the `/bin/stty` command is for modifying STREAMS modules to reflect changes in the user's environment. It does not work with the `/usr/ucb` version of `stty`, which has not been internationalized.

The following command queries the user's environment for information on EUC code set width and sets.

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

For example, if the user has the environment variable `LC_CTYPE` set to `locale`, this option gets information on the number of bytes per character and the screen width per character for the code sets in the `ko` and `ko.UTF-8` environments and then sends this information to relevant modules in the stream.

TTY Setup Examples

The system administrator can add `setterm` in the startup script in `/etc/rcn.d` directory (where `n` is the run level), to run at the system boot time. Also, users can run the `setterm` command at login to configure the stream for their terminal, including the appropriate modules for Korean input code conversion. The following examples using `setterm` work as commands typed at a system prompt or included in system files such as `.cshrc`, `.login`, and the startup script. Such commands can either explicitly set the device type or use the `terminfo` database.

Configuring STREAMS for Korean Solaris Software

You can use the commands in this section to configure STREAMS modules.

- Use the following command to explicitly configure the STREAMS module for the EUC, completion code, terminal type.

```
system% setterm -x EUC
```

- Use the following command to explicitly initialize the Packed STREAMS module that supports the combination code of KS C 5601-1987.

```
system% setterm -x PACK
```

- Use the following command to explicitly initialize the Johap STREAMS module that supports a combination code of KS C 5601-1992.

```
system% setterm -x JOHAP
```

Use this procedure to run a configuration command that references an entry in the terminfo database.

1. Add the following entry for a FAST-15 Packed code terminal to the terminfo database.

```
fast-15 | fast-pack | korean terminal packed mode,
devt=PACK,
use=vt100-w,
```

2. Use the following configuration command to reference the entry for the FAST-15 Packed code terminal.

```
system% setterm -t fast-15
```

For `setterm` to work properly, the `/usr/share/lib/setterm/ko/conf.file` must contain an entry that corresponds to the device type. This entry gives `setterm` instructions for placing the appropriate conversion modules in the TTY stream.

- Add the following entry that corresponds to the device type to the `/usr/share/lib/setterm/ko/conf.file`.

```
#
KoreanPACK|PACKthrow \
                                poppto zs|mcp|mti|ptem \
                                push kpack \
                                push ldterm \
                                push ttcompat \
                                run {stty defeucw} \
                                catch

#
KoreanJOHAP|JOHAP   throw \
                                poppto zs|mcp|mti|ptem \
                                push kjohap \
                                push ldterm \
                                push ttcompat \
                                run {stty defeucw} \
                                catch

#
GenericEUC|EUCthrow \
                                poppto zs|mcp|mti|ptem \
```

```

push ldterm \
push ttcompat \
run {stty defeucw} \
catch

#
ASCIIThrow \

popto zs|mcp|mti|ptem \
push ldterm \
push ttcompat \
catch

#

```

Configuring STREAMS for the ko.UTF-8 Locale

You can use the commands in this section to configure STREAMS modules the ko.UTF-8 Locale.

- Use the following command to explicitly configure the STREAMS module for the ko.UTF-8 locale.

```
system% setterm -x u8
```

- Use the following command to explicitly initialize an EUC terminal to use the ko.UTF-8 locale

```
system% setterm -x ku8euc
```

- Use the following command to explicitly initialize the Johap STREAMS module. This module supports combination code of the KS C 5601-1992 terminal to use the ko.UTF-8 locale.

```
system% setterm -x ku8johap
system% stty defeucw cs8 -istrip
```

Use this procedure to run a configuration command that references an entry in the terminfo database.

1. Add the following entry for a FAST-15 Packed code terminal to the terminfo database.

```
fast-15 | fast-pack | korean terminal packed mode,
devt=PACK,
use=vt100-w,
```

2. Use the following configuration command to reference the entry for the FAST-15 Packed code terminal.

```
system% setterm -t fast-15
```

For setterm to work properly, the /usr/share/lib/setterm/ko.UTF-8/conf.file must contain an entry that corresponds to the device type. This entry gives setterm instructions for placing the appropriate conversion modules in the TTY stream.

- Add the following entry that corresponds to the device type to the `/usr/share/lib/setterm/ko.UTF-8/conf.file`.

```
# Korean specific entries:
#
# Completion/Wansung/EUC code terminal support (KS C 5601-1992)
KoreanU8EUC|KU8EUCthrow \
    popto zs|mcp|mti|ptem \
    push ku8euc \
    push eucu8 \
    push ldterm \
    push ttcompat \
    push u8euc \
    run {stty defeucw} \
    catch
#
# Combination code terminal support (KS C 5601-1992 Annex 3)
KoreanU8JOHAP|KU8JOHAPthrow \
    popto zs|mcp|mti|ptem \
    push ku8johap \
    push eucu8 \
    push ldterm \
    push ttcompat \
    push u8euc \
    run {stty defeucw} \
    catch
#
```

Terminal Support

The Korean Solaris Operating System supports completion-code terminals and Packed-code terminals. The terminals should have built-in automata, Hangul/Hanja fonts, and Hangul/Hanja input methods.

Note – Packed code is also called combination code, KS C 5601-1987 and KS C 5601-1992.

Installing a Terminal

If you have not added a terminal to your system before, first try installing a terminal in ASCII mode only.

▼ How to Install a Terminal to Support KS C 5601

1. Become superuser.

```
system% su
```

2. Type `admintool`.

The `admintool` menu will appear.

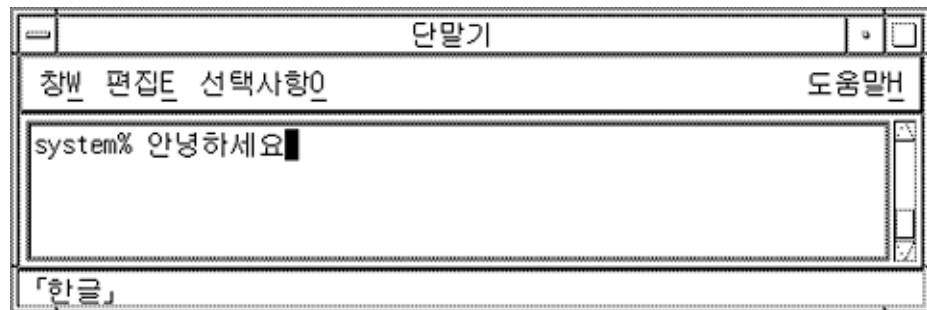
```
system# admintool
```

3. Select the serial ports icon.

4. Click edit on the serial ports menu.

The modify service submenu appears.

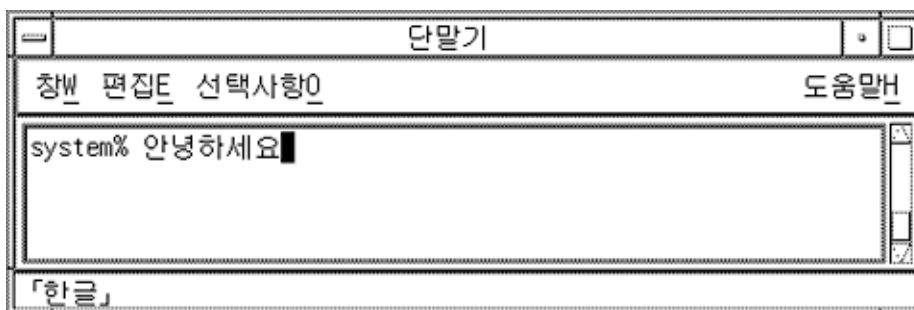
5. From the modify service submenu, select enabled, baud rate 9600. Then enter the terminal type.



6. Under expert options, select the create utemp entry and enter the appropriate module in the streams modules field.

To install a terminal that supports the KS C 5601-1987 version of combination code for the Korean Solaris Operating System, add `kpack` to the streams modules field.

To install a terminal that supports the KS C 5601-1992 version of combination code for the Korean Solaris Operating System, add `kjohap` to the streams modules field.



For more information on `admintool`, see the *System Administration Guide: Solaris Containers, Resource Management, and Virtualization*.

▼ How to Set Up a `ttya` Port from the Command Line

Use the following procedure to set up a terminal on `ttya` port from the command line.

1. Determine the port monitor version number.

```
# ttyadm -v
```

The port monitor version number will display.

2. Enter the following commands and substitute the port monitor version number for *ver*.

```
# pmadm -r -p zsmon -s ttya
# sacadm -a -p zsmon -t ttymon -c /usr/lib/saf/ttymon -v ver
```

For more information, see `pmadm(1M)` and `sacadm(1M)` man pages.

3. Use the `pmadm` command that matches your terminal type to add a login service.

For EUC terminals, use the following command.

```
# pmadm -a -p zsmon -s ttya -i root -fu -v ver -m " `ttyadm -s y \
-T terminal_type -d /dev/ttya -l 9600 -m ldterm,ttcompat -s \
/usr/bin/login`"
```

For Korean Packed code terminals that use the KS C 5601-1987 version of Combination code, type the STREAMS module `kpack` in the `ttadm` command.

```
# pmadm -a -p zsmon -s ttya -i root -fu -v ver -m "`ttyadm -s y \
-T terminal_type -d /dev/ttya -l 9600 -m kpack,ldterm,ttcompat -s \
/usr/bin/login`"
```

For Korean Combination code terminals, which use the KS C 5601-1992 version of Combination code, type the STREAMS module `kjohap` in the `ttadm` command.


```
# pmaadm -a -p zsmom -s ttya -i root -fu -v ver -m ``ttyadm -S y \  
-T terminal_type -d /dev/ttya -l 9600 -m kjohap,ldterm,ttcompat -s \  
/usr/bin/login``
```

4. Turn on the terminal.

Refer to the documentation that accompanies the terminal.

5. Log in the terminal.

6. Check to see that the installation is correct.

```
# setenv LANG ko  
# /bin/stty cs8 -istrip defeucw
```

Note – These values show that the operating system is set to communicate with the terminal in 8-bit no-parity mode. Make sure the terminal is set up in 8-bit no-parity mode. Refer to the terminal’s setup manual for the proper way to set terminal options.

▼ How to Verify TTY Set Up

Use the following procedure to verify that the TTY is properly set up.

1. Type the `/bin/stty` command with the `-a` option.

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

2. If the `cs8`, `-istrip` values set in the previous procedure are not listed, use the following command to set them.

```
system% /bin/stty cs8 -istrip defeucw
```

This is the last step in setting up a terminal. The default setting of a Korean terminal is the completion-code mode. Continue to the next section to install a Packed-code TTY.

▼ How to Use Packed Code and Johap TTY in the ko Locale

If you use a Packed code (KS C 5601-1987) terminal, you must load the `kpack STREAMS` module into the kernel.

1. Become superuser.

```
system% su  
Password: (Type superuser password here if required.)
```

2. Load the `kpack` module.

```
#modload /kernel/strmod/kpack
```

3. Type the following command to enable the module.

```
system% setterm -x PACK
```

▼ How to Use Combination Code and Johap TTY in the ko Locale

If you use a combination code (KS C 5601-1992) terminal , you must load the kjohap STREAMS module into the kernel.

1. Become superuser.

```
system% su  
Password: (Type superuser password here if required.)
```

2. Load the kjohap module.

```
# modload /usr/kernel/strmod/kjohap
```

3. Type the following command to enable the module.

```
system% setterm -x JOHAP
```

▼ How to Use the EUC TTY in the ko.UTF-8 Locale

If you use an EUC (KS C 5601-1987-0) terminal , you must load the ku8euc STREAMS module into the kernel.

1. Become superuser.

```
system% su  
Password: (Type superuser password here if required.)
```

2. Load the ku8euc module.

```
#modload /kernel/strmod/ku8euc
```

3. Type the following command to enable the stream module.

```
system% setterm -x KU8EUC
```

4. Type the following command to enable 8-bit I/O.

```
system% stty defeucw cs8 -istrip
```

5. If you want to use dtterm, type the following command.

```
system% setterm -x u8
```

▼ How to Use the Johap TTY in the ko.UTF-8 Locale

If you use a Johap (KS C 5601-1992-3) code terminal, load the `ku8johap` STREAMS module into the kernel.

1. Become superuser.

```
system% su
Password: (Type superuser password here if required.)
```

2. Load the `ku8johap` module.

```
#modload /kernel/strmod/ku8johap
```

3. Type the following command to enable the module.

```
system% setterm -x KU8JOHAP
```

4. Type the following command to enable 8-bit I/O.

```
system% stty defeucw cs8 -istrip
```

5. If you want to use `dtterm`, type the following command.

```
system% setterm -x u8
```


Binary Compatibility Package

Applications compiled under Korean OpenWindows 2.x or Solaris 1.x or SunOS 4.x systems have different binary formats than the current Korean Solaris release. Older applications can nevertheless be run under the current Korean release without being recompiled by using its included binary compatibility package (BCP).

Note – SUNWowbcp must be included in your system configuration in order for you to run the following commands. See your system administrator for installation.

The following BCP command runs the compiled binary code of earlier SunOS4.x, Solaris 1.x, and Korean OpenWindows 2.x applications without recompilation, although OpenWindows V2 Korean applications display no input server status region. As shown in the following examples, the command calls the application by its old name (*old_application_name*) and sets the basic locale, input language, and display language using the older version's specific locale name (*oldlocale*).

```
system% old_application_name -lc_basicleoleoldlocale -lc_inputlang  
oldlocale -lc_displaylangoldlocale
```

The following example shows the command that you use to run the compiled binary code of an earlier version of the `textedit` application on a Korean current Solaris system.

```
system% textedit -lc_displaylang korean -lc_basicleole korean \  
-lc_inputlang korean
```

Due to incompatibilities between Korean Solaris 2.x and 1.x applications, you cannot cut and paste Korean characters between them.

Korean Test Utilities

Every utility listed in this section is supported, but for this version of Solaris, you are encouraged to use the XPG4 internationalization APIs as described in the *International Language Environments Guide*.

The utilities in the following table test various aspects of the Korean (KS C 5601) national standard character set. Except Korean `isksc`, they also assume that the character being tested is part of the national standard character set.

The arguments for the functions in the following table must be a character in `WC`, `wchar_t`. For more information, see the `kctype(3x)` man page.

TABLE A-1 Korean Character Classification Functions

Utility	Description
<code>isksc</code>	Returns true if it is in the KS C 5601 character set.
<code>iskroman</code>	Returns true if it is a Roman character as defined by the KS C 5636 character set.
<code>iskromannum</code>	Returns true if it is a Roman numeral symbol in the KS C 5601 character set.
<code>isksymbol</code>	Returns true if it is a Latin symbol or special character in the KS C 5601 character set.
<code>iskparen</code>	Returns true if it is a right or left parenthesis in the KS C 5601 character set.
<code>isklatin</code>	Returns true if it is a Latin letter character in the KS C 5601 character set.
<code>iskletter</code>	Returns true if it is a Korean vowel or consonant in the KS C 5601 character set.
<code>iskline</code>	Returns true if it is a ruled line symbol in the KS C 5601 character set.
<code>iskunit</code>	Returns true if it is a unit character in the KS C 5601 character set.
<code>isksci</code>	Returns true if it is a scientific symbol in the KS C 5601 character set.
<code>iskgen</code>	Returns true if it is a graphic or general symbol in the KS C 5601 character set.
<code>iskgreek</code>	Returns true if it is a Greek character in the KS C 5601 character set.
<code>iskrussian</code>	Returns true if it is a Russian character in the KS C 5601 character set.
<code>iskuser</code>	Returns true if the character is in the user-defined area of the KS C 5601 character set.

TABLE A-1 Korean Character Classification Functions (Continued)

Utility	Description
<code>iskhanja</code>	Returns true if it is an ideogram in the KS C 5601 character set.
<code>iskhangul</code>	Returns true if it is a Hangul phonogram in the KS C 5601 character set.
<code>iskkata</code>	Returns true if it is a Japanese Katakana character in the KS C 5601 character set.
<code>iskhira</code>	Returns true if it is a Japanese Hiragana character in the KS C 5601 character set.

Asian-Specific Utilities

This section describes functions for wide character and string input and output, character classification, and conversion functions for the Korean character sets. Asian Solaris software implements a wide character library for handling Korean character codes according to industry standards.

Routines that have Korean language-specific dependency are in their own language-specific library, which is linked with the corresponding C compiler option. In Korean Solaris, `libkle` is linked with `-lk1e`. Refer to the appropriate man page for more information.

Asian Solaris software defines `WC` as a constant-width, four-byte code. `WC` uses the ANSI C data type `wchar_t`, which the Solaris software defines in `wchar.h` as shown in the following example.

```
typedef long wchar_h;
```

In Solaris software, `long` is four bytes.

Conversion Utilities

The conversion utilities described in this section are available, but you should use `iconv()` as a standard function.

Asian Solaris software provides facilities for various conversions.

- Character conversion within a code set, such as converting uppercase ASCII to lowercase.
- Conversions between different conventions of the national standard character sets.
 - Conversions between Combination code and Completion code of KS C 5601-1987 and KS C 5601-1992 characters sets.

- Conversions between GB and EUC.
- Conversions between CNS 11643 code and Big5.
- Conversions between code formats such as EUC and WC.

Programs that use the general multibyte conversion utilities should include the header files `widec.h` and `wctype.h`. Korean Solaris specific routines (such as `iskxxx`) are declared in `ko/xctype.h`.

Programs that use the general multibyte conversion utilities should include three header files: `wctype.h`, `widec.h`, and `ko/xctype.h`.

Conversion Within a Code Set

The multibyte conversion functions are similar to the one-byte conversion functions `toupper()` and `tolower()`. These functions convert wide-characters to other wide characters. For more information on conversion routines, see the man pages for `wconv(3)` for all locales and `kconv(3)` for Korean.

Conversion for Korean Character Codes

The following routines perform character-based code conversion on the KS C 5601 character set. They convert characters in the set between Completion code (or EUC format) and Combination code (or Packed code). To use these routines, the library `k1e` must be linked using the C compiler option `-lk1e`. For more information, see the `kconv(3x)` man page.

TABLE A-2 Korean Code Conversion Functions

Function	Description
<code>comptopack()</code>	Converts a character in Completion code to Combination (packed) code of KS C 5601-1987.
<code>packtocomp()</code>	Converts a character in Combination (Packed) code of KS C 5601-1987 to Completion code.
<code>wasuntojohap()</code>	Converts a character in Completion code to Combination (Packed) code of KS C 5601-1992.
<code>packtocomp()</code>	Converts a character in Combination (Packed) code of KS C 5601-1992 to Completion code.

Mapping Korean Keyboard Functions

This appendix shows you how to configure a Sun Korean keyboard to make selected key functions.

You can use the `xmodmap` command to remap any key to function as the Compose or the AltGraph key. The following example command sequence makes the right Meta key function as the AltGraph key.

```
system% xmodmap -e "remove mod1 = Meta_R"
system% xmodmap -e "remove mod2 = Mode_switch"
system% xmodmap -e "add mod2 = Meta_R"
system% xmodmap -e "keysym Meta_R = Mode_switch"
```

The following command sequence maps the left Meta key () to the Compose key.

```
system% xmodmap -e "keysym Meta_L = Multi_key"
```

The `$OPENWINHOME/share/etc/keytables/Korea4.kt` and `$OPENWINHOME/share/etc/keytables/Korea5.kt` files provide more information.

Glossary

ANSI	American National Standards Institute. ANSI proposes standard definitions for different computing languages. The most recent standard for the C language, prepared by the ANSI C X3J11 Committee, includes library functions for computing with multibyte characters for international usage, as well as a new data type, <code>wchar_t</code> , for dealing with four-byte characters. This standard is not completed, so it is referred to as the “proposed ANSI C standard,” or ANSI C-X3J11.
ASCII	American Standard Code for Information Interchange. A seven bit code containing English upper and lowercase letters, punctuation, numbers and control codes. The eighth bit in each byte is used by different applications for parity checking, communication and message passing protocols, compacting data, or other purposes. Applications that are intended to be internationalized cannot utilize this bit if they are going to use multiple code sets or multibyte characters, and utilities that handle multiple code sets or multibyte characters.
Category	In the Korean Solaris documentation set, category is related to localization. A category is a portion of a country’s language representation and cultural conventions. For instance, the date is often represented in the U.S. as <i>Month, Day, Year</i> ; while in another country it might be <i>Day, Month, Year</i> . The date and time can be thought of as one category of a local language. Categories also refer to the program categories, the environment variables that are related to categories, and the ANSI localization tables for each category.
Character Set	A character set is defined as a set of elements used for the organization, control, or representation of data. Character sets may be composed of alphabets, ideograms, or other units. This may seem a bit open-ended, but character sets may contain other character sets, which makes the boundaries unclear. For example, the KS C 5601 character

set contains English, Greek, Russian, and Japanese character sets, in addition to Hangul syllables (consonant and vowel combinations), Hanja ideograms (Chinese characters), and many other characters.

code set	Also called a coded character set, this is a set of unambiguous rules that establishes a character set and the one-to-one relationship between each character in the character set and its bit representation. For example, the English character set, including punctuation and numbers, can be mapped to the ASCII code set in such a way that each character corresponds to only one bit code, and no bit code corresponds to more than one character.
Combination code	Another name for <i>Packed code</i> or <i>Johap code</i> described below.
Completion code	Also called Wangsung. Completion code is a pre-defined set of Korean character codes, which maps preselected Hangul, Hanja, special symbols, alphabets of other languages and so on into two-byte coding space. This representation is defined in KS C 5601 and used as EUC code set 1 by the Korean Solaris Operating System.
EUC	Extended UNIX Code. Describes four code sets modelled on ISO-2022. Each code set can contain one or more different character sets, like the Hangul and Hanja character sets in KS C 5601. The four code sets are referred to as code sets 0, 1, 2, and 3, and in this text they are sometimes abbreviated as cs0, cs1, cs2, and cs3. Other internationalization efforts sometimes call these g0, g1, g2, and g3. Code set 0 is also called the primary code set, and code sets 1, 2, and 3 are called the supplementary code sets. In the Korean and Chinese implementations of the EUC codes, the primary code set (cs0) contains ASCII and begins with a zero in the most significant bit.
Hangul	Hangul is the phonetic alphabet commonly used in Korea. Each character corresponds to a spoken syllable, usually a consonant-vowel pair or a consonant-vowel-consonant triad. KS C 5601 defines 2350 Hangul characters used in standard computing.
Hanja	Hanja characters are Korean ideograms, which came originally from ancient China (the word itself means Chinese character). They were adopted many centuries ago and have evolved somewhat different meanings in China and Korea. But because they are not phonetically based, Chinese and Korean Hanja have remained closer in meaning than have Italian, French, and Spanish, which evolved into separate languages over the same time span. The Korean Industry Standard defines the 4888 most frequently used Hanja characters in the KS C 5601 standard.

ISO	International Standards Organization. Composed of a number of professional societies and companies, this organization studies and makes recommendations on internationalization issues. ISO 2022 proposes and describes the Extended UNIX Codes. Other ISO proposals include the European 8-bit code and communication protocols for internationalization.
Johap code	Johap code is a Packed code (also called Combination code), which is defined in the KS C 5601-1992 document. Unlike the Packed code defined in KS C 5601-1987 or before, Johap code has a set of Hanja characters and special symbol characters.
KSC	Korean Industry Standard Code Set. This is the Korean analogue to ASCII. The KSC describes standards for computing in the Korean environment. KS C 5601 contains code assignments in Completion code for Hangul and Hanja characters, graphics and punctuation characters, two Japanese phonetic alphabets (Hiragana and Katakana), control codes, and several western alphabets (Roman, Russian, and Greek characters). This standard defines 2350 Hangul characters, 4888 Hanja characters, and 986 additional characters (for punctuation, foreign alphabets, numbers, graphics, and others). Each character is two bytes long, and does not utilize the highest or most significant bit of each byte. In other words, it uses the lower seven bits of each byte for character assignments.
Locale	A locale describes a language or cultural environment. Its setting affects the display or manipulation of language-dependent features. Korean Solaris software provides C for U.S.A, ko for Korean extended UNIX code, and ko.UTF-8 for Korean Universal Multiple Octet Coded Character Set Transmission Format.
N-byte code	This coding system assigns each Korean alphabetic consonant or vowel a one-byte code. These are built up into Hangul syllabic characters with the Hangul automata.
Packed code	Packed code (also called Combination code) is a systematic method for coding Hangul syllabic characters in a two-byte code. Each 16-bit (two-byte) character contains a high or most-significant bit (1) and three 5-bit fields. These fields contain the codes for the beginning consonant (<i>x</i>), a middle vowel (<i>y</i>), and an optional ending consonant (<i>z</i>), as follows: <i>1xxxxxyyyyzzzzz</i> . Hanja characters cannot be represented in Packed code, because many Hanja characters may be represented by one phonetic pronunciation. Packed code is defined in KS C 5601-1987 and earlier as a supplementary code set.
POSIX	Portable Operating System for Computer Environments. An IEEE standards group comprising seven committees that create documents for standardizing and internationalizing UNIX. POSIX document

1003.1 deals with the kernel and system calls. 1003.2 concerns the C-shell and standard libraries. The other five deal with real-time computing, communications and networking, and other issues.

UTF-8

Universal Multiple Octet Coded Character Set (UCS) Transmission Format. `ko-UTF-8` provides the Korean-related characters in this standard. UTF-8 is a representation of Unicode.

Unicode

The international character set and encoding developed by the Unicode Consortium.

Wide Character Code (WC)

A constant-width four-byte code, called WC in Asian Solaris documentation, for the internal representation of EUC codes using the new ANSI-C data type `wchar_t`. Although EUC does not specify limits on the size of the supplementary code sets (code set 0 is always one byte), WC specifies a character as four bytes. Standardizing on four bytes takes up more memory space than necessary if the environment is primarily ASCII, but it also speeds processing time for strings of mixed characters; the 1000th character always begins at byte 4000 (and the 0th character starts at byte 0). This is useful for any type of indexing in applications.

X/Open

X/Open started as a consortium of international UNIX vendors from Europe, USA, and Asia. It is now one of the major standards organizations like POSIX and ANSI; source of the *X/Open System Interface Portability Guide*.

Index

A

- admintool, configuring serial ports, 111-112
- auxiliary window, 33-34
 - input method help, 61
 - input method options, 57-59
 - lookup table, 60-61
 - palette, 56-57
 - virtual keyboards, 59-60

B

- BDF conversion
 - to PCF, 95-96
 - with xtobdf, 74
- binary compatibility package (BCP), 117
- bitmap fonts
 - ko locale, 96
 - ko.UTF-8 locale, 97

C

- CDE, 25
 - Korean, 27
- character mode, switching, 55-56
- characters
 - committing, 38-39
 - deleting, 36-38
 - full_width, 55-56
 - half_width, 55-56
 - mode switching, 55
 - special symbols, 47-51

- characters (Continued)

- wide, 119
- code set
 - conversion, 119
 - locale list, 20
- commands
 - ioctl, 105
 - mailx, 73
 - pmadm, 112
 - setterm, 106-107
 - stty, 107, 113
 - talk, 74
 - TTY, 106-107
- composite fonts, 94
- configuring a terminal port, 112-113
- conversion
 - files, 75
 - utilities, 119

D

- desktop tools
 - Address Manager, 28
 - Application Manager, 28
 - Audio Tool, 28
 - Calculator, 28
 - Calendar Manager, 28
 - Clock, 28
 - Console, 28
 - File Manager, 28
 - Find Files, 28
 - Front Panel, 28

desktop tools (Continued)

- Help, 28
- Icon Editor, 28
- Image Viewer, 28
- Mailer, 28
- Performance Meter, 28
- Process Manager, 29
- Style Manager, 29
- Terminal, 29
- Text Editor, 29

DPS, 91-95
Korean fonts, 94-95

E

EUC (Wansang) files, printing, 79-80

F

fonts
BDF to PCF conversion, 95-96
directories, 96
installing, 95-96
Kodig, 91-95
Myeongjo, 91-95
full_width characters, 55-56

G

Graphical User Interface (GUI), Korean, 27

H

half_width characters, 55-56
Hangul
finding Hanja, 65-67
Hanja conversion, 44-45, 63
hex input mode, 51-54
Hangul-Hanja dictionary
adding entries, 67-70
modifying entries, 70
Hanja
Hangul conversion, 41-46
lookup mode, using, 42

Hanja (Continued)

- System Dictionary, 64-72
- User Dictionary, 64-72

Hanja tool, launching, 64-65
htt, 87-90
starting, 89

I

iconv utility, 75
input, Korean, 32
input conversion, Hangul to Hanja, 34
input methods
auxiliary window, 57-59
help auxiliary window, 61
iiim server, 89-90
IM server, 87-90
locale list, 20
new, 17
options, 57-59
input mode
ASCII, 39
Hangul, starting, 34
hex, using, 51-54
Internet Intranet Input Method (iiim), 89-90
ioctl command, 105

J

Johap
kjohap, code conversion, 104
ko_KR.UTF-8 locale, 32
printing, 79-82, 99-101
Johap TTY, 113-114

K

keyboard
2 Beol Sik, 59
3 Beol Sik 390, 59
3 Beol Sik Final, 60
input, 27
mapping, 121
keyboards, virtual, 40-41
kjohap, Johap code, conversion, 104

- ko_KR.EUC (ko) locale
 - code conversion, 106-107
 - KS X 1001, KS C 5601-1987 code set, 32
 - printing, 79-80, 99-101
- ko_KR.UTF-8 (ko.UTF-8) locale
 - code conversion, 106-107
 - KS C 5601-1992-3 code set, 32
 - printing, 81, 99-101
- ko_KR.UTF-8 (ko.UTF-8), printing, 79-82
- ko locale, conversion, 75
- ko.UTF-8 locale, conversion, 76
- Kodig fonts, 91-95
- kpack, Packed code, conversion, 104

L

- laser printer, 79-82
- line printer, 79-82
- locales
 - categories, 20-21
 - ko fonts, 96
 - ko_KR.EUC, 79-80, 89, 99-101, 106-107
 - ko_KR.UTF-8, 81, 89-90, 99-101, 106-107
 - ko.UTF-8 fonts, 97
 - modifier, 21
 - setting, 26, 85-86
- localization, Korean, 19
- lookup table, auxiliary window, 60-61

M

- mailx command, 73
- mapping keys, 121
- Motif, 25
 - Korean, 27
- mp
 - print filter, 79-82
 - utility, 18, 81-82, 100-101
 - Xprt client, 101
- Myeongjo fonts, 91-95

O

- OpenWindows, 117

P

- Packed
 - ko_KR.UTF-8 locale, 32
 - kpack, code conversion, 104
 - printing, 79-82, 99-101
- Packed TTY, 113-114
- palette auxiliary window, 56-57
- PCF format, conversion from BDF, 95-96
- pmadm command, 112, 113
- PostScript
 - fonts, 91-95
 - printer, 79-82
- preedit area, auxiliary window, 33-34
- printing
 - EUC (Wansang) files, 79-80
 - from a command line, 79-82
 - using mp, 79-82, 100-101
 - UTF-8 files, 81

S

- Serial Ports, 111-112
- setterm command, 106-107
- Solaris, Korean localization, 19-20
- special characters, 47-51
- status area, auxiliary window, 33-34
- stdconvtool, using, 74
- STREAMS
 - Korean Solaris, 104
 - traditional, 104
- stty command, 107, 113

T

- talk command, 74
- termcap database, 105-106
- terminal support, 110
- terminfo database, 106
- TrueType fonts, ko/ko.UTF-8 locale, 97
- TTY
 - commands, 106-107
 - set up, 107-110, 113
 - STREAMS, 103-104
 - utilities, 105-106

U

Unicode 3.2, support, 18
UTF—8, files, printing, 81
utilities
 iconv, 75, 119
 mp, 18, 81-82, 100-101
 test, 118
 xetops, 18
 xtobdf, 74
 xutops, 18

V

virtual keyboards
 auxiliary window, 59-60
 layouts switching, 40-41

W

Wansang code, ko_KR.EUC locale, 32
Window Manager, 25

X

X Print Xerver (Xprt), mp, 101
xetops utility, 18
XIM, 87-90
xtobdf utility, 74
xutops utility, 18