Traditional Chinese Solaris User's Guide

Sun Microsystems, Inc.
901 N. San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part No: 806–3488–10
March 2000

Contents

Preface

Traditional Chinese Solaris User's Guide describes product behavior unique to the Traditional Chinese Solaris™ operating environment and answers many questions commonly asked during initial experience with the software. This guide introduces the general appearance and properties of a variety of localized Desktop Tools™ and utilities offered with the Traditional Chinese Common Desktop Environment (CDE) and OpenWindows™ environment.

Who Should Use This Book

This user's guide is for someone who wants to use the Traditional Chinese features of Solaris software to manage files, calendar, and e-mail, write or print Traditional Chinese files, and so forth. Tools for these and many other applications run under Traditional Chinese Solaris software. This guide helps you easily find, access, and get started with these tools. You should read this guide:

- If you have not used Traditional Chinese Solaris software before
- For information on using product features
- If you need instructions for starting up your Traditional Chinese Solaris operating environment (see especially Chapter 2, "Starting the Traditional Chinese Solaris Software.")

Before You Read This Book

Become familiar with the basics of the Solaris base release user documents, particularly the ones listed under "Related Books" on page xiii. This user's guide focuses on using the Traditional Chinese features of the Desktop Tools and other features of Traditional Chinese Solaris software.

How This Book Is Organized

Each chapter of this guide addresses a different aspect of using Traditional Chinese Solaris software. The chapters tell how to check your set up before you begin using the facilities of Traditional Chinese Solaris software and give step-by-step instructions for using Traditional Chinese facilities.

Chapter 1, "Introduction to Traditional Chinese Solaris Software," briefly describes general modifications made to Solaris software, including CDE, to internationalize and localize it for Traditional Chinese.

Chapter 2, "Starting the Traditional Chinese Solaris Software," gives the step-by-step instructions you must follow to start your Solaris user environment. It also describes Traditional Chinese Solaris-specific features you must use to turn Traditional Chinese facilities OFF/ON by using dtlogin.

Chapter 3, "Using the `htt` Input Method Server," introduces the startup, appearance, and use of `htt`.

Chapter 4, "Entering Traditional Chinese Text," describes different Traditional Chinese character entry modes and provides a step-by-step tutorial in their use. (Further information on customizing commands and other advanced user topics are covered in *International Language Environments Guide* and *Traditional Chinese Solaris System Administrator's Guide*.)

Chapter 5, "Localized Applications," describes uses of two desktop tools localized for Chinese: `mailx`, `talk`, and tools to convert file codes.

Chapter 6, "Font Editor," explains how to customize fonts used in your Traditional Chinese Solaris applications.

Chapter 7, "Traditional Chinese Printing Facilities," discusses Traditional Chinese Solaris support for line printers with built-in Chinese fonts or using the Traditional Chinese Solaris `xetops` filter package.

Appendix A, “Open Windows Information,” describes the special requirements of the OpenWindows environment.

Appendix B, “Binary Compatibility Package,” discusses running compiled binary code of earlier SunOS™ 4.x/Solaris 1.x/Asian OpenWindows 2.x applications without recompilation.

Appendix C, “Running Networked Applications,” discusses running localized applications that reside on another machine across your network.

Appendix D, “Mapping Traditional Chinese Keyboard Functions,” discusses how to configure a Sun Chinese keyboard to make selected key functions when you need them.

The Glossary contains a list of words and phrases found in the Traditional Chinese Solaris documentation set, and their definitions.

Related Books

You should become familiar with the following basic documentation:

- *Solaris Introduction*
- *Solaris User's Guide*
- *Solaris 8 (SPARC Platform Edition) Asian Release Notes*
- *Solaris 8 (Intel Platform Edition) Asian Release Notes*

Advanced users may want to read *Solaris Advanced User's Guide*. Advanced users wanting to customize their system environment or the operations of their Sun tools will find much pertinent information in *International Language Enviornments Guide* and *Traditional Chinese Solaris System Administrator's Guide*. These books give information on setting up, administering, programming, and customizing product features for advanced users, developers/programmers, and system administrators.

What Typographic Changes Mean

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

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`

`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root 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.

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	`#`

CHAPTER 1

Introduction to Traditional Chinese Solaris Software

Introduction

This software is a Traditional Chinese localization of the Sun™ Solaris operating environment. The Traditional Chinese Solaris software includes the Traditional Chinese Common Desktop Environment (CDE) for windowed applications that are built on Sun's Solaris operating environment.

Design of Traditional Chinese Solaris Software

Traditional Chinese Solaris software is an extension of base Solaris software. Virtually all utilities and features of the U.S. and International Solaris standard releases are incorporated in Traditional Chinese Solaris software. These products introduce Solaris input methods for the input and output of Traditional Chinese. Application programs and CDE use the features of Traditional Chinese CDE to communicate with users in Traditional Chinese.

This Traditional Chinese localization of Sun's internationalized CDE includes enhancements for handling appropriate linguistic and cultural conventions, which it provides to two broad working environments:

- A localized user environment, which includes localized desktop tools and window manager (`dtwm`) that communicate with users in Traditional Chinese.
- A localized development environment, which programmers use to develop localized applications, with `Xlib` and Motif, which have been internationalized for

this use. Programmers and developers should refer to *Solaris Internationalization Guide for Developers*.

Traditional Chinese Graphical User Interface

This Traditional Chinese Solaris release uses the Traditional Chinese CDE Motif graphical user interface, which is similar in layout and design to the U.S. release of CDE. Traditional Chinese CDE supports multibyte characters and Traditional Chinese 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 Traditional Chinese input include a *status area* associated with their input window to show the current conversion mode. With an input conversion mode on, as Traditional Chinese is being typed its entry point becomes a highlighted (reverse video) *preedit area* until the input is converted to Traditional Chinese or special characters and committed. Some input modes also provide conversion choices among several Traditional Chinese characters on menus.

Traditional Chinese Input/Output

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

Traditional Chinese input at the keyboard is stored temporarily in an intermediate representation. The conversion manager program, with the help of user interaction, then transforms this intermediate representation into a displayed character string.

The Localized CDE Desktop

The following desktop tools are provided in this Solaris release. All can handle Traditional Chinese input and output. A manual page is provided for each.

Address Manager – Carries out remote operations and finds information about the systems and users on your network. Can speed up such tasks as sending email, logging in remotely, and setting appointments on someone else's calendar.

Application Manager – Contains the applications available on your system. You can launch these applications through the Application Manager interface.

Audio Tool – Tool for recording, playing, editing, and controlling workstation audio parameters.

Calculator – Visual calculator for use with mouse or keyboard.

Calendar Manager – Manages business and social appointments; can use electronic mail to send automatic reminders.

Clock – Displays current analog or digital time.

Console – Standard Motif scrolling window terminal emulator.

File Finder – Tool for searching for a folder and subfolders that returns a list of files or folders that match your search criteria. You can also specify the size, owner, date modified, type, and permissions in your search criteria.

File Manager – Graphical tool for accessing files and directories. Represents file types with varying colors and icons. Navigates through the file system with the mouse.

Front Panel – A centrally-located window containing controls for accessing applications and utilities, including the workspace switch. The Front Panel occupies all workspaces.

Help – On-line searchable help for CDE.

Icon Editor – Visual tool for editing icon appearance and creating new icons.

Image Tool – Interactive image viewer. Image Tool can be used to view the contents of file types such as GIF, TIFF, JPEG, PostScript, and others.

Mailer – Tool for handling electronic mail.

Performance Meter – Real-time system performance meter that can display a variety of data.

Print Manager – Graphical front-end to the print command. It supports *drag-and-drop* file transfer operations.

Process Manager – Tool for displaying and performing actions on the processes that are currently running on your workstation.

Snapshot – Tool to *snap* or capture picture of a window or region of a screen in a bitmap (raster file). Used for capturing screen image displays in this user's guide.

Text Editor – Visual text editor used in CDE tools such as the Mailer composition window.

Style Manager – Tool for setting workstation preferences, such as audio feedback from keyboard, mouse response, and so on.

Terminal – Standard Motif window terminal emulator. The window behaves like an ASCII character terminal for entry of UNIX® commands at a system shell prompt and other terminal operations.

CHAPTER 2

Starting the Traditional Chinese Solaris Software

The Traditional Chinese Solaris operating environment must be specially set up for using Traditional Chinese text facilities. This chapter describes the steps required to set up the Traditional Chinese environment and to start Traditional Chinese Solaris operation.

The Traditional Chinese Solaris operating environment provides two window environments, CDE and the OpenWindows environment. CDE is a fully internationalized environment; it does not require most of the administration tasks that the OpenWindows environment requires to handle Traditional Chinese. For information on starting up OpenWindows, see Appendix A, “Open Windows Information.” The Traditional Chinese Solaris product includes the following locales:

- `C` – ASCII English environment
- `zh_TW` – Traditional Chinese environment in extended UNIX code (EUC) CNS 11643-1992 standard
- `zh_TW.BIG5` – Traditional Chinese environment in the Big5 codeset
- `zh_TW.EUC` – Symbolic link to zh_TW locale
- `zh_TW.UTF-8` – Traditional Chinese environment in Unicode 3.0 standard

Note – The zh_TW.BIG5 locale runs under CDE, but does not support OpenWindows.

Setting the Default Locale

You can change your default locale using the following procedure.

- **Choose the language button on the dtlogin window.**
- **Select the** C**,** zh_TW**,** zh_TW.UTF-8**, or** zh_TW.BIG5 **locale.**

  Your new locale is now in effect

Using Aliases to Set Locales

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

Setting Up Locale Aliases for the Korn Shell

- Put aliases like the following in your .profile file.

```
alias twn_EUC "export LANG=zh_TW; /bin/stty cs8 -istrip defeucw"
alias twn_BIG5 "export LANG=zh_TW.BIG5; /bin/stty cs8 -istrip defeucw"
alias asc "export LANG=C; /bin/stty cs7 istrip defeucw"
alias twn_UTF-8 "export LANG=zh_TW.UTF-8; /bin/stty cs8 -istrip defeucw"
```

Setting Up Locale Aliases for the C Shell

- Put aliases like the following in your .cshrc file.

```
alias twn_EUC "setenv LANG zh_TW; /bin/stty cs8 -istrip defeucw"
alias twn_BIG5 "setenv LANG zh_TW.BIG5; /bin/stty cs8 -istrip defeucw"
alias asc "setenv LANG C; /bin/stty cs7 istrip defeucw"
alias twn_UTF-8 "setenv LANG=zh_TW.UTF-8; /bin/stty cs8 -istrip defeucw"
```

Changing Font Directories

The Xsession script (located in /usr/dt/bin) that comes with the Traditional Chinese Solaris operating environment includes the following font path: /usr/openwin/lib/locale/*locale*/X11/fonts, where *locale* is either zh_TW or zh_TW.BIG5. To add a different font directory path dynamically, type:

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

CHAPTER 3

Using the `htt` Input Method Server

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

This chapter explains the basic display features of the `htt` input server. Normally, you do not need to change any `htt` settings, as discussed in this chapter, to operate any Traditional Chinese Solaris application.

You need to be familiar with the input method terminology in Chapter 4, "Entering Traditional Chinese Text," before reading this chapter. However, to use Solaris applications for Traditional Chinese character input you do not need to read beyond the first section of this chapter.

The Traditional Chinese Solaris `htt` server icon looks like this:

Input Method Server Basic Properties

The Traditional Chinese Solaris operating environment starts the `htt` input method server automatically when you start the Solaris operating environment in a locale that requires an input method. htt continues to run and service applications that are started and connected to it. If you need to restart htt, refer to "How and When htt Is Started," on page 15.

So each Traditional Chinese Solaris application that uses htt for Traditional Chinese character input typically finds htt running when it starts. To get htt's service for language input, htt should be running before an application is started. If an application does not find htt running when it starts, that application may not be able to get the input service even if htt is started later.

Three htt Processes

The input method server comprises three related programs (htt_props, htt, and htt_server) processes running together. One process controls the input method server properties, another controls the population of the input method server (that is, a "watch dog" process), and the third handles input methods of clients.

Usually, you don't need to know the details of these three processes. When htt is started, htt_props and htt_server are started automatically.

htt Property Manager

The htt input method server is started as an icon. Double click SELECT on the icon to start htt Property Manager to configure the behavior of htt.

Resetting and Terminating htt

The input method server property manager includes the following menus:

File Menu

Reset Input Manager – resets and restarts the htt input method server. This operation is not needed unless the input method server requires resetting (for example, when an application stops getting the language input). Selecting this operation destroys any intermediate data (such as preedit texts).

Exit – terminates htt. When you select this operation, htt Property Manager offers three options.

- Exit – terminates htt completely. If you need to restart htt, start it from a shell window:

  ```
  system% htt &
  ```

- Background – terminates htt Property Manager only. The htt icon and htt Property Manager window are no longer visible. However, the htt input method server still functions. If you need to start htt Property Manager again, start htt_props from a shell window.

  ```
  system% htt_props &
  ```

- Cancel – cancels the termination.

Help Menu

The htt Property Manager provides four categories of help messages to guide its operation:

- General – describes the operations in the general dialog
- Preedit Status – describes the operations in the Preedit/Status dialog
- Lookup Table – describes the operation in the Lookup table dialog
- About – describes the menu operations from the main Property Manager window

Using htt Command Line Options.

htt's command line options can be referenced from its man page or by typing:

```
system% htt -help
```

Customizing the on/off Key From General Dialog

You can customize the key sequence to turn on/off locale specific input. To add a key sequence, perform the following steps:

1. **Press the Add button.**

   You will see the Add Key popup.

2. **Press the key to be added and OK.**

   The key pressed is added to the conversion key list.

   To remove any key from the list:

   - Select the key from the list and press Remove.

Note – The customized key sequence will take effect when you restart `htt`.

Customizing the Preedit/Status Window From the Preedit/Status Dialog

You can control the Preedit/Status window placement in this panel if your application specifies the root window style Preedit or Status.

1. Selecting "Position on the screen" places the window in a fixed location of the workspace. This window is shared by multiple applications whose input style is root window style.
2. Selecting "Position relative to the cursor" places the window close to the mouse cursor.
3. Selecting "Attach to client input window" places the window near the application's input window.

   In addition to the placement of the Preedit/Status window for the root window style, you can configure the behavior of the preedit string when it reaches the end of a line. By default, the preedit string will wrap around to the second line. But if you de-select "Wrap long lines in on-the-spot preedit," the preedit string will not exceed one line and it will scroll when reaching the end of a line.

Customizing the Lookup Window From the Lookup Table Dialog

You can control the appearance of the lookup window by choosing one of the four options listed:

- Client window status area – places the lookup choices in an area where the input status is shown on the application's window
- Preedit/status window – places the lookup choices in the window placed on the workspace, which is used by root window style applications
- Popup window near cursor – places the popup near to the current position of the mouse
- Popup window near client input window – places the popup near the application's input window

  You can also specify the maximum rows and columns for the lookup choices that can be displayed on a single popup. You can specify the labels used for lookup choices. The options are uppercase or lowercase letters or numbers.

How and When `htt` Is Started

An application that uses the `htt` input method server to receive Traditional Chinese characters must find `htt` running before it can receive such input. So if your default locale is the `C` locale (which does not require `htt`) and you then change from `C` to another locale in a single Terminal, for example, you must start `htt` with a command line like the following:

```
system% htt &
```

With the locale set to `zh_TW` or `zh_TW.BIG5`, `htt` is started automatically when the Traditional Chinese windowing environment starts. In CDE, `htt` is started from a script, `/usr/dt/config/Xsession.d/0020.dtims`. This script is executed by `Xsession`, which runs at every initialization stage upon a user's login from `dtlogin`. This script ensures that `htt` is started before other applications in CDE.

Using the `iiim` Server in `zh_TW.UTF-8` Locale

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

To switch to other Language Engines, press CTRL+Space, as with other Asian Locales.

To switch to Chinese Input Methods, click the left mouse button in the Status Area. The following Language Engine List appears:

- Cyrilllic
- Greek
- Thai
- Arabic
- Hebrew
- Unicode Hex
- Unicode Octal
- Lookup
- Traditional Chinese

The following Language Engines appear if their corresponding locales are installed:

- Japanese — if `ja` locale is installed
- Korean— if `ko` locale is installed.
- Simplified Chinese — if `zh` locale is installed.

After the Traditional Chinese Language Engine is selected, you can use Function Keys to switch between different Input Methods:

For more detailed information about each Input Method, see "Chapter 4, Entering Traditional Chinese Text".

CHAPTER 4

Entering Traditional Chinese Text

About This Chapter

This chapter describes Traditional Chinese Solaris input modes for typing Traditional Chinese characters. You can type any of the following kinds of characters:

- ASCII/English
- Chinese
- Special symbols

You can type any of these characters in the input areas of the following application subwindows:

- Terminal emulation (TTY) windows, such as a Terminal
- Text entry subwindows, such as those used by Text Editor or Mailer
- Control panel subwindows, such as used by File Manager for typing a file name
- Other special use subwindows, such as pop-ups

Character Sets

The Traditional Chinese Solaris operating environment conforms to the CNS 11643 standard, updated in late 1992, and the Big5 character sets. CNS 11643-1992 is a Chinese national standard in Taiwan. It defines 16 planes, 7 planes of which are assigned:

TABLE 4–1

Plane	Contents
Plane 1	Miscellaneous symbols, Hanzi radicals, and Roman and Greek alphabets, total 684 symbol characters in the range of 0x2121 to 0x427E, and 5,401 most commonly used Hanzi characters in the range of 0x4421 to 0x7D4B.
Plane 2	A total of 7,650 secondary commonly-used Hanzi characters in the range of 0x2121 to 0x7244.
Plane 3	A total of 6,148 other Hanzi characters, including some user-defined characters from the original plane 14 characters and different shaped characters in the range 0x2121-0x6246 from the Republic of China's (ROC) Department of Education.
Plane 4	This plane contains a total of 7,298 characters, including some of ISO/IEC 10646 defined CJK Unified Han characters (range: 0x2121-0x6E5C).
Plane 5	This plane contains a total of 8,603 characters that the ROC Department of Education defined as currently-used characters but not included in planes 1 through 4 (range: 0x2121-0x7C51).
Plane 6	This plane contains a total of 6,388 characters that the ROC Department of Education defined as different shaped characters but not included in planes 1 through 5 (range: 0x2121-0x647A).
Plane 7	This plane contains a total of 6,539 characters that the ROC Department of Education defined as different shaped characters but not included in planes 1 through 6 (range: 0x2121-0x6655).
Planes 8 to 11	These planes are not yet assigned.
Planes 12 to 16	These planes are for user-defined characters.

Big5 was defined by five major Taiwanese computer vendors (including the Institute of Information Industry) in May, 1984. Although Big5 is not the national standard, it is more widely used than the CNS 11634-1992

The total number of characters defined in Big5 is 13,523. It is a subset of CNS 11643-1992.

Traditional Chinese Solaris software provides code conversion between Chinese code conventions at three levels of support:

- User commands support file transfers for existing files in different codes.
- Library functions support application development for existing codes.
- STREAMS modules support existing TTY devices using different codes.

Input Window Areas

Three separate areas of an application subwindow are involved in entering characters. These areas are typically displayed, named, and used as follows:

- Preedit area—Highlighted (reverse video or underlined) entry display area
- Status area—Area indicating the current input/conversion mode
- Lookup choice area—Area displaying multiple character choices

Preedit Area

The highlighted (reverse video or underlined) preedit area displays characters as they are typed or converted. It holds formations of text before it is converted to an ideogram or symbol and put in the text block being assembled for the application.

Lookup Choice Area

The lookup choice area displays multiple Chinese or special character choices available for conversion of the character(s) or radical(s) in the preedit area. In the above example, it is a pop-up menu.

Status Area

The status area shows which input conversion mode is in effect. In the above example, it is located in the lower left corner of the window margin.

Input and Conversion Modes for Entering Characters

The following input and conversion modes are available for entering ASCII/English and Chinese characters:

- ASCII input mode (Control-spacebar)
- Array input mode (F2)
- BoShiaMy input mode (F3)
- DaYi input mode (F4)
- ChuYin input mode (F5)
- I-Tien input mode (F6)
- Telecode input mode (F7)
- TsangChieh input mode (F8)
- ChienI input mode (F9)
- NeiMa input mode (F11)
- ChuanHsing input mode (F12)

Press Control-spacebar to toggle Chinese input conversion on or off. The listed function keys and Control keys (such as F2 and F3) toggle on the corresponding conversion modes.

TABLE 4–2 Traditional Chinese Solaris Function Keys Default Settings for Input Conversion Modes

Default Key	Function Description
Control-spacebar	Toggle ASCII/Chinese input conversion mode on/off.
F2	Turn on Array mode.
F3	Turn on BoShiaMy mode.
F4	Turn on DaYi mode.
F5	Turn on ChuYin mode.
F6	Turn on I-Tien mode.
F7	Turn on Telecode mode.
F8	Turn on TsangChieh mode.
F9	Turn on CheinI mode.
F11	Turn on NeiMa mode.
F12	Turn on ChuanHsing mode.

TABLE 4–2 Traditional Chinese Solaris Function Keys Default Settings for Input Conversion Modes *(Continued)*

Default Key	Function Description
spacebar	Convert and display the preedit area choice.
Control-H	Erase one character.
Control-N	Display the next group of choices.
Control-P	Display the previous group of choices.

The /usr/lib/mle/zh_TW/keybind.dat file contains the settings for these keys. Refer to *Traditional Chinese Solaris System Administrator's Guide* for information about changing these default keybinding settings.

Typing ASCII Characters

Each tool first starts in ASCII input mode, which you can toggle on or off by pressing Control-spacebar or the Chinese/English key on a Chinese keyboard. Use this mode to type ASCII keyboard characters, such as the echo command shown in the following figure:

[ASCII] is displayed in the window's status area when this mode is on. When ASCII input mode is off, the current conversion mode symbol appears.

Typing Non-ASCII Characters

As previously mentioned, each tool starts in ASCII mode, as shown by [ASCII] in the status area.

The first step in typing Chinese characters or other non-ASCII symbols is to toggle ASCII input mode off by pressing Control-spacebar (or the Chinese/English key on a Chinese keyboard). Turning ASCII input mode off also turns conversion mode on, which is shown in the status area.

Each tool's first conversion mode is TsangChieh. You can then change to a different conversion mode by pressing its function key (for example, F2 or F3).

When you toggle input conversion off and ASCII input mode on (using Control-spacebar), the conversion manager program notes which conversion mode is on. Then when you toggle conversion mode back on again, the conversion manager program turns the last conversion mode on again.

Some conversion modes offer lookup choice areas. When more than one page of choices is available, press Control-N to display the next page of lookup choices or Control-P to display the previous page.

Pressing Delete, Backspace, or Control-H deletes the last character or symbol you typed while Chinese input conversion mode is on.

Array Input Mode

Array input mode works on the radicals that compose Chinese characters. It contains 260 radicals and allocates them on the keyboard in ten columns and three rows. To used this input mode effectively, you need to memorize only ten basic keystrokes. All radicals are naturally allocated on the keyboard by the basic keystrokes.

Typing Array Text

1. **Open a new Terminal (in ASCII mode) and type `echo`.**
2. **Type Control-spacebar to turn on TsangChieh mode.**

   The mode appears in the status area.
3. **Press the F2 key to turn on Array mode.**

   The mode appears in the status area.
4. **Type `x`.**

   The Array input converter finds several matching characters, so it automatically displays a lookup choice pop-up:
5. **Type `1` to select the appropriate character.**

   The radical is converted and the correct character appears.
6. **Type `eq`.**

   A lookup choice pop-up is displayed.

7. **Type** `2` **to select the appropriate character.**

   The radicals are converted and the correct character appears:

8. **Type** `l`**.**

   A lookup choice pop-up is displayed.

9. **Type** `0` **to select the appropriate character.**

   The radical is converted and the correct character appears:

10. **Type** `az` **and** `o`**.**

    The input converter finds multiple character choices. A lookup choice pop-up is displayed for each set of radicals.

11. **Type** `1` **to select the appropriate character.**

    The radicals are converted and the correct character appears.

12. **Type** `.tx` **and** `rkr`**.**

    The input converter finds multiple character choices. A lookup choice pop-up is displayed for each set of radicals.

13. **Type** `1` **to select the appropriate character.**

    The radicals are converted and the correct character appears.

BoShiaMy Input Mode

BoShiaMy input mode works on the radicals that compose Chinese characters. The radicals in this mode are easy to memorize by their shape, sound, or meaning. For effective use of this mode, memorize the rules: "Cut off long" and "Patch up short."

Typing BoShiaMy Text

1. **Open a new Terminal (ASCII mode) and type `echo`.**

2. **Type Control-spacebar to turn on TsangChieh mode.**
   The mode appears in the status area.

3. **Press the F3 key to turn on BoShiaMy mode.**
   The mode appears in the status area.

4. **Type the sets of radicals:** `uoo`, `ffl`, **and** `d`. **Press the spacebar after each set to convert it to its character.**

   The BoShiaMy input converter finds only one matching character for each set of radicals. The character is displayed as the next radical is typed:

5. **Type** `edn`.

   The input converter finds two matching characters and displays a lookup choice pop-up:

6. **Type** `a` **to select the appropriate character.**

   The radical is converted and the correct character appears:

7. **Type** `kxx` **and** `midk`. **Press the spacebar after each set to convert it to its character.**

   The input converter finds only one matching character for each set of radicals. The character is displayed as the next radical is typed:

8. **Type** `yryf`.

   The input converter finds four matching characters and displays a lookup choice pop-up:

9. **Type** `a` **to select the appropriate character.**

   The radical is converted and the correct character appears:

DaYi Input Mode

Similar to the TsangChieh input mode, DaYi conversion works on the radicals that compose Chinese characters. The keystroke sequence of the DaYi input mode completely follows the Chinese writing sequence, and only a few rules need to be memorized.

Typing DaYi Text

1. **Open a new Terminal (ASCII mode) and type** `echo`**.**
2. **Type Control-spacebar to turn on TsangChieh mode.**

   The mode appears in the status area.
3. **Press the F4 key to turn on DaYi mode.**

   The mode appears in the status area.

4. **Type the sets of radicals:** `;o`, `9ec`, `/db.` **and** `ev`**. Press the spacebar after each set to convert it to its character.**

   The DaYi input converter finds only one matching character for each set of radicals. The character is displayed as the next radical is typed:

5. **Type** `c8`**.**

   The input converter finds three matching characters and displays a lookup choice pop-up:

6. **Type** `b` **to select the appropriate character.**

   The radical is converted and the correct character appears:

7. **Type** `erk`**.**

   The input converter finds only one matching character. The character is displayed as the next radical is typed:

8. **Type** `f8fy`**.**

   The input converter finds two matching characters and displays a lookup choice pop-up:

9. **Type** `b` **to select the appropriate character.**

   The radical is converted and the correct character is displayed:

ChuYin Input Mode

ChuYin is another common input mode. This mode applies the Chinese phonetic symbols to the phonetics of Mandarin Chinese. When a typed sequence of phonetics uniquely matches a Chinese character, the phonetics are automatically converted and the character is displayed. However, if more than one character could represent the phonetics, a lookup selection area appears showing you the available choices. Several pages of lookup choices may be available for some phonetic sequences.

Typing ChuYin Text

The following example uses ChuYin mode to enter Chinese characters representing the statement "Taipei's weather is very hot."

1. **Open a new Terminal (ASCII mode) and type** `echo`**.**

2. **Type Control-spacebar to turn on TsangChieh mode.**

   The mode appears in the status area.

3. **Press the F5 key to turn on ChuYin mode.**

   The mode appears in the status area.

4. **Type** `w96`**.**

   The ChuYin input converter finds several matching characters and displays a lookup choice pop-up:

5. **Press** `a` **to select the appropriate character.**

   The radicals are converted and the correct character appears.

6. **Type** `1o3`**.**

   The input converter finds only one matching character for this set of radicals. The character is displayed as the final radical is typed

7. **Type** `2k7`**.**

   The input converter finds two matching characters, so a lookup choice pop-up is displayed:

8. **Press** `a` **to select the appropriate character.**

   The radicals are converted and the correct character appears.

9. **Type** `wu0`**.**

   These are not immediately converted and the lookup selection is not displayed because the last character is not the tone character.

10. **Press the spacebar to open the selection of lookup choices.**

11. **Type** `a` **to select the appropriate character.**

    The radicals are converted and the correct character appears.

12. **Type** `fu4`**.**

    The input converter finds several matching characters, so a lookup choice pop-up is displayed:

13. **Type** `f` **to select the appropriate character.**

    The radicals are converted and the correct character is displayed.

14. **Type** `cp3`**.**

    The input converter finds two matching characters, so a lookup choice pop-up is displayed:

15. **Type** `a` **to select the appropriate character.**

    The radicals are converted and the correct character appears.

16. **Type** `bk4`**.**

    The input converter finds three matching characters, so a lookup choice pop-up is displayed:

17. **Type** `a` **to select the appropriate character.**

    The radicals are converted and the correct character appears.

I-Tien Input Mode

I-Tien mode maps Chinese phonetics to the keyboard according to the sound of each Roman keyboard character.

Type the command `itkbd` to display the I-Tien character mapping to the keyboard. This keyboard mapping and its icon are displayed as follows:

Typing I-Tien Input

1. **Open a new Terminal (in ASCII mode) and type `cat`.**
2. **Type Control-spacebar to turn on TsangChieh mode.**

   The mode appears in the status area.
3. **Press the F6 key to turn on I-Tien mode.**

   The mode appears in the status area.

4. **Type** `dx83`**.**

   The I-Tien input converter finds only one matching character and displays it:

5. **Type** `v`**,** `r`**, and a space.**

   The input converter finds several matching characters and displays a lookup choice pop-up:

6. **Press** `f` **to select the appropriate character.**

   The radicals are converted and the correct character appears.

7. **Type** `ce-2`**.**

   The input converter finds several matching characters and displays a lookup choice pop-up.

8. **Press** `b` **to select the appropriate character.**

   The radicals are converted and the correct character appears:

9. **If this Chinese-title file contained the following poem, pressing the Return key would run the** `cat` **command on this file and display:**

Telecode Input Mode

Telecode conversion uses the old four-digit codes used by the Taiwan Telephone & Telegraph Bureau. These codes uniquely identify 16,356 Chinese characters. They are organized into two planes: plane 1 numbered from 0000 to 8045, and plane 2 numbered from .0000 to .8489.

Typing Telecode Input

For an example of how Telecode conversion mode works, type two characters, one from each Telecode plane as follows.

1. **Open a new Terminal (ASCII mode) and type** `echo`**.**
2. **Type Control-spacebar to turn on TsangChieh mode.**

   The mode appears in the status area.

3. **Press the F7 key to turn on Telecode mode.**

   The mode appears in the status area.

4. **Type the first three digits of code `3456` (plane 1).**

   The preedit area shows the code:

5. **Type the fourth digit `6`.**

   The correct converted character appears:

6. **Type a period followed by the first three characters of character code `3456` to enter the character from plane 2.**

   The screen should look like this:

7. **Type `6`.**

   The correct converted character appears:

TsangChieh Input Mode

Each tool's first input conversion mode is TsangChieh, a commonly used entry mode.

TsangChieh is the default mode for typing Chinese (and other non-ASCII) characters when ASCII mode is toggled off.

Press F8 to switch to TsangChieh mode at any time. You can see that TsangChieh conversion mode is on by checking the status area, as shown in the following figure:

TsangChieh conversion works on the radicals that compose Chinese characters. Each Chinese character has a keystroke sequence made of the radicals that compose it. A TsangChieh keystroke sequence often uniquely defines a Chinese character, but sometimes selection from multiple choices is required.

TsangChieh conversion mode can also be used to type the characters in the Chinese National Standard 11643. To type characters in this mode, press `z` followed by the three-letter code assigned to the symbol by the standard.

Typing Characters in TsangChieh Mode

To type Chinese characters in this mode, type the Chinese radicals that make up the Chinese characters. If you do not have a Taiwan Chinese keyboard that shows the radicals on the keys, refer to a picture of the Chinese keyboard to see how the Chinese radicals map to your keyboard.

For an example, to enter a command that states in Chinese “Taipei’s weather is very hot,” perform the following steps:

1. **Open a new Terminal (in ASCII mode) and type `echo`.**
2. **Type Control-spacebar to turn on TsangChieh mode.**

   The mode appears in the status area.

3. **Type** `ir`**.**

   The TsangChieh input converter finds two matching characters and displays a lookup choice pop-up:

4. **Press** `a` **to select the appropriate character.**

   The radicals are converted and the correct character appears:

5. **Type** `lmp`**.**

   The TsangChieh input converter does not display a lookup choice pop-up because it does not find multiple character choices for these radicals. However, the converter does not automatically convert the preedit area radicals to a character.

6. **Press the spacebar to convert the radicals to their character.**

   The radicals are converted and the correct character appears.

7. **Type** `hapi`**,** `mk`**,** `onfd`**, and** `hoav`**. Press the spacebar after each set to convert it to its character.**

   The input converter finds only one matching character for each set of radicals. The character is displayed as the next radical is typed.

8. **Type `gif`.**

   The input converter finds two matching characters and displays a lookup choice pop-up:

9. **Press `a` to select the appropriate character.**

   The radicals are converted and the correct character appears: art/ct.8.im1

Typing Symbols in TsangChieh Mode

This conversion mode also allows you to enter the symbols in CNS 11643. Type the letter `Z` followed by the symbol's standard-assigned three-letter code. For example, to type "alpha," the first character in the Greek alphabet, in TsangChieh mode, do the following:

1. **Type `Z` and `gal` (code for the alpha symbol).**

   They appear in the preedit area:

2. **Press the spacebar to convert the code to its symbol.**

ChienI Input Mode

ChienI mode is a simplification of TsangChieh. Only the first and last keystrokes of the TsangChieh sequence for the target Chinese character make up the ChienI mode input.

This usually displays a lookup selection of Chinese characters, as shown in the following two-character input example.

Typing ChienI Mode Input

1. **Open a new Terminal (ASCII mode) and type** `echo`**.**
2. **Type Control-spacebar to turn on TsangChieh mode.**

   The mode appears in the status area.
3. **Press the F9 key to turn on ChienI mode.**

   The mode appears in the status area.
4. **Type** `h` **and** `a`**.**

   The ChienI input converter finds several matching characters and displays a lookup choice pop-up:
5. **Press** `k` **to select the appropriate character.**

   The radical is converted and the correct character appears.
6. **Type** `a` **and** `h`**.**

   The input converter finds several matching characters and displays a lookup choice pop-up:
7. **Press** `a` **to select the appropriate character.**

   The radical is converted and the correct character appears.

NeiMa Input Mode

NeiMa input is the hexadecimal representation of the target character’s EUC code. This requires four keystrokes for a two-byte EUC code or eight keystrokes for a four-byte EUC code. Such input results in a unique match (for a recognized code) or an error.

Typing NeiMa Text With Plane 1

In this example, to create the character with EUC code A2A1, type `A2A1`.

1. **Open a new Terminal (ASCII mode) and type** `echo`**.**

2. **Type Control-spacebar to turn on TsangChieh mode.**

   The mode appears in the status area.

3. **Press the F11 key to turn on NeiMa mode.**

   The mode appears in the status area.

4. **Type** `A2A` **(the first three keystrokes of the code):**

5. **Type** `1` **(the fourth keystroke).**
   The corresponding character appears:

6. **Press Return to enter the** `echo` **command:**

Typing NeiMa Text With Plane 2

1. **Open a new Terminal (ASCII mode) and type** `echo`**.**
2. **Type Control-spacebar to turn on TsangChieh mode.**
   The mode appears in the status area.
3. **Press the F11 key to turn on NeiMa mode.**
   The mode appears in the status area.
4. **Type** `8EA2F2C`**.**
   In the preedit area `8EA2F2C` appears in reverse video.

5. **Type** `4`**.**

   The corresponding character appears:

6. **Press Return to run the** `echo` **command:**

Typing NeiMa Input With Plane 3

1. **Open a new Terminal (ASCII mode) and type** `echo`**.**
2. **Type Control-spacebar to turn on TsangChieh conversion mode.**

   The mode appears in the status area.
3. **Press the F11 key to turn on NeiMa mode.**

   The mode appears in the status area.
4. **Type** `8EA3E2C`**.**

   In the preedit area `8EA3E2C` appears in reverse video.
5. **Type** `6`**.**

   The corresponding character appears:

6. **Press Return to enter the** `echo` **command:**

ChuanHsing Input Mode

Using this input mode makes the keyboard ASCII/English characters and punctuation two-columns wide in the input. No intermediate characters appear or are used. ChuanHsing input is committed directly to the application just as in ASCII.

Typing ChuanHsing Mode

1. **Open a new Terminal (ASCII mode) and type** `echo`**.**
2. **Type Control-spacebar to turn on TsangChieh mode.**

   The mode appears in the status area.
3. **Press the F12 key to turn on ChuanHsing mode.**

   The mode appears in the status area.
4. **Type any keys on the keyboard, for example, spacebar** `aAbBcC` **spacebar** `1234`**.The wide characters are displayed:**

CHAPTER **5**

Localized Applications

This chapter describes selected properties you need to use on two localized applications of the Traditional Chinese Solaris operating environment. This chapter also provides lists of code conversion utilities.

These tools (and the commands to invoke them) include:

- `mailx` (`/usr/SUNWale/bin/mailx`)
- `talk` (`/usr/SUNWale/bin/talk`)

Using the `mailx` Utility With Traditional Chinese Characters

The `encoding` variable in `.mailrc` does not affect Mailer behavior, but it does set `mailx` function. To send e-mail in formats other than 7-bit ASCII, such as (8-bit) Traditional Chinese 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. So Traditional Chinese 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 Traditional Chinese characters.

Using talk With Traditional Chinese Characters

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

The xtobdf Utility

Traditional Chinese Solaris software provides this BDF font generator:

xtobdf — Converts font in X server to font in BDF.

The runb5 Utility

Traditional Chinese Solaris software provides a utility that permits terminal-based Big5 applications to run under Sun's TTY window environment:

runb5 – Runs Big5 applications under Sun's TTY window environment.

Sdtconvtool

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

The following steps show how to convert a file encoded in UTF-8 to BIG5 encoding:

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

   Click on the arrow button to the right of the "Source Code Set" label to reveal a list of available code sets in the system. Scroll through the list and select the code set of the file to be converted. In this case, select "UTF-8."

2. **Type the path of the file to be converted.**

   The path to the file can either be entered manually in the “Source File Path” area, or chosen by selecting the “Browse...” button and selecting the file name from the file selection box. In this case, enter or select `/tmp/en_US.UTF-8`.

   The “Clear” button to the right of the “Browse...” button can be used to erase the entered source or path.

3. **Select the code set to which the file will be converted.**

   Select the target file code set information from the pulldown menu to the right of the “Target Code Set:” label, as in Step 1. In this case, select “BIG5.”

4. **Type the path for the converted file.**

   Enter the path to the file can in the “Target File Path” area. In this case, enter or select `/tmp/BIG5.all.txt`.

   The “Clear” button to the right of the “Browse...” button can be used to erase the entered source or path.

5. **Press the “Start Conversion” button.**

   The “Start Conversion” button will begin the conversion process using the given information. The “Clear All Fields” button will erase both the source and target file path names.

`iconv`

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. Traditional Chinese 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 codeset. 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 Traditional Chinese Solaris software.

TABLE 5–1

Source Code	Symbol	Target Code	Symbol
CNS 11643	zh_TW-euc	Big-5	zh_TW-big5
CNS 11643	zh_TW-euc	ISO 2022-7	zh_TW-iso2022-7
Big-5	zh_TW-big5	CNS 11643	zh_TW-euc
Big-5	zh_TW-big5	ISO 2022-7	zh_TW-iso2022-7
ISO 2022-7	zh_TW-iso2022-7	CNS 11643	zh_TW-euc
ISO 2022-7	zh_TW-iso2022-7	Big-5	zh_TW-big5
CNS 11643	zh_TW-euc	ISO 2022-CN-EXT	zh_TW-iso2022-CN-EXT
ISO 2022-CN-EXT	zh_TW-iso2022-CN-EXT	CNS 11643	zh_TW-euc
Big-5	zh_TW-big5	ISO 2022-CN	zh_TW-iso2022-CN
ISO 2022-CN	zh_TW-iso2022-CN	Big-5	zh_TW-big5
UTF-8	UTF-8	CNS 11643	zh_TW-euc
CNS 11643	CNS 11643	UTF-8	UTF-8
UTF-8	UTF-8	Big-5	zh_TW-big5
Big-5	zh_TW-big5	UTF-8	UTF-8
UTF-8	UTF-8	ISO 2022-7	zh_TW-iso2022-7
ISO 2022-7	zh_TW-iso2022-7	UTF-8	UTF-8
IBM Cp937	zh_TW-cp937	UTF-8	UTF-8
UTF-8	UTF-8	IBM Cp937	zh_TW-cp937
Big-5+	zh_TW-big5p	UTF-8	UTF-8
UTF-8	UTF-8	Big-5+	zh_TW-big5p

In the following example, an EUC mail file is converted to Big5:

```
system% iconv -f zh_TW-euc -t zh_TW-big5 mail.euc > mail.big5
```

For further information, see the *iconv(3)* and *iconv_zh_TW(5)* man pages. These utilities can be used for converting files for printing. See *Traditional Chinese Solaris System Administrator's Guide* for more information.

CHAPTER 6

Fonts

This chapter describes the PostScript fonts included in the Traditional Chinese Solaris operating environment, what you need to use them, and how to edit them.

This chapter also describes how to use the Font Editor for bitmap fonts.

Display PostScript System (DPS)

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

Using Traditional Chinese PostScript Fonts and DPS Facilities

The Traditional Chinese Solaris operating environment DPS provides the fonts listed in the following table.

TABLE 6–1

Font Name	Description
Kai-Medium	Alias of Kai-Medium-EUC-H.
Kai-Medium-EUC-H	Kai-Medium font, EUC encoding, horizontal display; can be used like a Roman font.

TABLE 6–1 *(Continued)*

Font Name	Description
Kai-Medium-EUC-V	Kai-Medium font, EUC encoding, vertical display; can be used like a Roman font.
Kai-Medium-H	Kai-Medium font, horizontal display, for making a composite with a Roman font.
Kai-Medium-V	Kai-Medium font, vertical display; for making a composite with a Roman font.
Ming-Light	Alias of Ming-Light-EUC-H.
Ming-Light-EUC-H	Ming-Light font, EUC encoding, horizontal display; can be used like a Roman font.
Ming-Light-EUC-V	Ming-Light font, EUC encoding, vertical display; can be used like a Roman font.
Ming-Light-H	Ming-Light font, horizontal display, for making a composite with a Roman font.
Ming-Light-V	Ming-Light font, vertical display; for making a composite with a Roman font.

You can use the following Traditional Chinese fonts just as you would use Roman fonts:

- Kai-Medium
- Kai-Medium-EUC-H
- Kai-Medium-EUC-V
- Ming-Light
- Ming-Light-EUC-H
- Ming-Light-EUC-V

The following figure shows a sample of Kai-Medium and Ming-Light.

Creating Composite Roman and Traditional Chinese Fonts

You can create composite fonts using one Roman font and one of the following Traditional Chinese fonts:

- Ming-Light-H

- Ming-Light-V
- Kai-Medium-H
- Kai-Medium-V

For example, the following PostScript code defines a sample composite font, Times-Italic+Kai-Medium, which uses Times-Italic for ASCII characters and Kai-Medium horizontal font for Traditional Chinese characters:

```
/Times-Italic+Kai-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
                                /Kai-Medium-H findfont
                ] def
currentdict
end
definefont pop
```

Using Traditional Chinese Fonts in DPS Programming

You can use Traditional Chinese fonts just as you use Roman fonts in DPS wrap definitions. The following sample code creates the above display:

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

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

You can tell `PSWDisplayText`*(Chinese text)* in a C program to display the designated Chinese text; for example, as shown below:

Traditional Chinese Solaris software provides TrueType support in DPS.

Using Font Editor

This section describes how to edit, create, and install Portable Compiled Format (PCF) fonts used by the Traditional Chinese Solaris operating environment. You can save your changes and use the edited font(s) on your local system. This process involves the following steps:

1. Preparing a workspace for the font(s) you edit or create
2. Editing Bitmap Distribution Format (BDF) font file(s) with Font Editor
3. Converting BDF font file(s) to PCF format
4. Making the font(s) usable on your system

Note – `zh_TW.BIG5` messages are not supported by Font Editor. If you run Font Editor under the `zh_TW.BIG5` locale, the Font Editor interface will be in English.

Setting Up a Working Directory

1. **Create a new directory workspace for editing fonts, for example:**

```
system% mkdir /tmp/newfont
```

2. **Change your working directory to that directory:**

```
system% cd /tmp/newfont
```

Starting the Font Editor

The Traditional Chinese Solaris operating environment provides Font Editor for editing fonts.

- Start the Font Editor.

A Font Editor window is displayed.

```
system% fontedit
```

Font Editor does not operate on PCF format font files (file extension `.pcf`), which are used by the Solaris operating environment. This tool handles only fonts in BDF, a portable format defined by the MIT X Consortium.

Preparing a Font File

- Get the font file you want to edit in BDF format.

The font's encoding should start at 8481 (0x2121). For an example, assume that you have a BDF file such as the following `myfont14.bdf`:

```
STARTFONT 2.1
COMMENT Sample Font
FONT Myfont-Medium14
SIZE 14 75 75
...
STARTCHAR C101
ENCODING 8481
...
```

Editing a BDF Font File

1. **In the Font Editor tool window, choose FileLoad and load the font template file for the character(s) you will create.**

2. **For example, to edit or create `myfont14` font characters, load `myfont14.bdf`:**

3. **Click Select button to display the Fontedit: Select window.**

4. **To edit a character, type the hexadecimal high and low bytes of the font encoding code assigned to the character.**

   Click the right or left arrows to increase or decrease the numbers.

5. **Click Edit in the Fontedit: Select window.**

   This clears the field or displays the character for editing in the Font Editor main window:

6. **Edit the glyph by turning on or off the pixels on the main canvas:**

   - Click the SELECT mouse button to turn on a pixel.
   - Click the ADJUST mouse button to turn off a pixel.
   - Hold down the MENU mouse button for a menu of additional functions.

   Note – The character glyph in the upper left corner of the main window shows the actual appearance of the glyph as you turn its pixels on and off.

   ---

7. **In the Select window, type the TsangChieh, I-Tien, and Telecode input methods keystroke sequences for the character.**

8. **Save the character, including its glyph and input strokes, by clicking Store in the Select window.**

9. **Repeat the preceding steps as needed to create or modify font glyphs.**

10. **After you have edited the character(s), select Save... from the File menu and type the name of the font file.**

    In this example, you would save the font file in `/tmp/newfont/myfont14.bdf`.

11. **Select Save Dictionary from the File menu to save the input keystrokes in the dictionary.**

    In this exercise, save the dictionary in `/tmp/newfont/eucinput.usr`.

12. **Click Continue in the message that appears to save the dictionary.**

    The dictionary is saved when you click Continue. A message appears saying you must install the dictionary in the `/usr/lib/mle/zh_TW/eucinput.usr` file on your system before you can use the new input method keystrokes to type the new characters.

Converting From BDF to PCF Format

Before Solaris applications can use a modified BDF file, it must be converted to a file in PCF format. It must then be replaced in the `$OPENWINHOME/lib/locale/zh_TW/fonts` directory as follows:

- Convert the file to PCF format by typing:

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

The `-o` option enables the matrix encoding used for Asian PCF font files. For more information, see the *bdftopcf(1)* and *mkfontdir(1)* man pages.

Installing and Checking the Edited Font

1. **To add a new bitmap, move the `.pcf` font file into your font directory. You may compress the `.pcf` font file before moving it, as follows:**

```
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-cns11643-16
Myfont-Medium14
^D
system% mkfontdir
system% xset +fp ‘pwd‘
```

3. **You can view your font by entering:**

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

Installing the User Dictionary

You must install the dictionary file created during the Font Editor session in /usr/lib/mle/zh_TW/eucinput.usr to make the new characters accessible by Chinese input methods.

Normally you must be superuser to install such a feature. If you do not have superuser privileges on your system, contact your system administrator for help.

- To install the dictionary file, become superuser and type the following command:

```
system% su
# cp /tmp/newfont/eucinput.usr /usr/lib/mle/zh_TW
```

Restarting the htt Input Method Server

Before you can use your input method keystrokes to access the new characters, you must start a new htt to find the dictionary and new characters.

- Click the Reset Input Method button in the htt Properties window to start a new input method server.

For more information on this operation, see Chapter 3, "Using the htt Input Method Server."

Accessing Edited Fonts

The edited font characters are then accessible to all applications using htt. You can test this by starting a new application, such as a Terminal, turning on input conversion (Control-spacebar), and typing the keystrokes you assigned to the new character.

CHAPTER 7

Traditional Chinese Printing Facilities

The Traditional Chinese Solaris operating environment supports printing Traditional Chinese output through the following types of printing facilities:

- Line printer containing built-in Traditional Chinese fonts
- PostScript-based printer

Note – Before you can print Traditional Chinese text, a system administrator must set up your printing support as described in *Traditional Chinese Solaris System Administrator's Guide*.

You can use the Asian Solaris `xetops` utility to print files containing Traditional Chinese text on a PostScript printer, regardless of other printing support.

These printing facilities can be used directly from a command line or from within Traditional Chinese Solaris applications as discussed in the following sections.

Printing From a Command Line

From a command line, you can print one of two ways:

- Directly to a line printer.
- Using the `xetops` utility to convert text to bitmapped graphics.

Printing With a Line Printer

The Traditional Chinese Solaris operating environment uses EUC code sets. Its printing applications (such as desktop tools) generate PostScript code that uses EUC. If you use different PostScript printing, make sure it has the same capabilities.

- To print an EUC file, use the following command:

  ```
  system% lp EUC-filename
  ```

- To print a Big5 format file on a printer that supports this format, use the following command:

  ```
  system% lp -d Big5_printer Big5-filename
  ```

  For more information on setting up the Big5 filter, see *Traditional Chinese Solaris System Administrator's Guide*.

- To print a file that contains Traditional Chinese characters in EUC format on an Epson® dot matrix printer, use the following command:

  ```
  system% lp -d Epson_printer  EUC-filename
  ```

  For more information on setting up the filter for dot matrix printers, see *Traditional Chinese Solaris System Administrator's Guide*.

Printing With the `xetops` Utility

The `xetops` utility allows you to print Traditional Chinese characters using a PostScript-based printer. The utility converts Traditional Chinese text into a bitmapped graphics printed image.

A typical command line for printing a file containing Traditional Chinese characters, with or without ASCII/English characters, would be as follows:

```
system% pr filename | xetops | lp
```

The file may contain ASCII/English characters as well as Traditional Chinese. Refer to the *xetops(1*) man page for more detailed information.

Printing With the `xutops` Utility

The `xutops` utility allows you to print Traditional Chinese characters in zh_TW.UTF-8 locale using a PostScript-based printer. The utility converts Traditional Chinese text into a bitmapped graphics printed image.

A typical command line for printing a file containing Traditional Chinese characters, with or without ASCII/English characters, would be as follows:

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

The file may contain ASCII/English characters as well as Traditional Chinese. Refer to the *xutops(1*) man page for more detailed information.

Printing with the mp Utility

A new and enhanced `mp(1)` print filter is available in the Solaris 8 environment to print in all three Traditional locals with a PostScript-based printer.

The output from the utility is standard PostScript, and can be sent to any PostScript printer.

Note – Starting with the next release of the Solaris environment, `xutops (10)` will be obsolete.

A typical command line for printing a file containing Traditional Chinese characters, with or without ASCII/English characters, would be as follows:

```
system% mp -L $LANGfilename | lp -d  printer
```

The file may contain ASCII/English characters as well as Traditional Chinese. Refer to the mp(1) man page for more detailed information.

You can also use the utility as a filter, as the utility accepts `stdin` stream:

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

You can set the utility as a printing filter for a line printer. For example, the following command sequence tells the printer service `LP` that the printer `lp1` accepts only `mp` format files. This command line 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 add the `lpfilter` utility for a filter by using the `lpfilter(1M)` command as follows:

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

The `lpfilter` command tells LP that a converter (in this case, xutops) is available through the filter description file named `pathname`. The pathname can be determined as follows:

```
Input types: simple
Output types: MP
```

```
Command: /usr/bin/mp
```

The filter converts the default type file input to PostScript output using `/usr/bin/mp`.

To print a UTF-8 text file, use the following command:

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

For more details on the mp(1) command, refer to the `mp(1)` man page.

APPENDIX **A**

Open Windows Information

This appendix contains information specific to the OpenWindows environment.

Introduction to OpenWindows

Traditional Chinese Solaris Software Design

This Traditional Chinese localization of Sun's internationalized OpenWindows environment includes enhancements for handling appropriate linguistic and cultural conventions, which it provides to two broad working environments:

- A localized user environment, which includes localized DeskSet tools and window manager (`olwm`) that communicate with users in Traditional Chinese.
- A localized development environment, which programmers use to develop localized applications, with Xlib and the XView™ Toolkit, which have been internationalized for this use. Programmers/developers should refer to *Solaris Internationalization Guide for Developers*.

Expanded Workspace Properties Worksheet

The Localization category in the Workspace Properties worksheet lets you set the locale in which applications will start. With this page you can set the Basic Setting, Display Language, Input Language, Numeric Format, and Time Format for new application windows from inside the Traditional Chinese OpenWindows environment.

These settings take effect each time a local application is started. Application windows are displayed in the locale that is currently set. If you change the locale, new application windows are displayed in the new locale, but existing application windows continue to be displayed in the original locale.

Traditional Chinese OPEN LOOK DeskSet

The following DeskSet tools are provided in this Solaris release. Each tool can handle Traditional Chinese language input and output. A man page is provided for each.

Audio Tool – Tool for recording, playing, editing, and controlling workstation audio parameters.

Binder – Tool for defining which actions are associated with which file types. This association can be set graphically.

Calculator – Visual calculator for use with the mouse or keyboard.

Calendar Manager – Manages business and social appointments; can use electronic mail to send automatic reminders.

Clock – Displays time in analog or digital format.

Command Tool – Standard OPEN LOOK scrolling window terminal emulator.

File Manager – Graphical tool for accessing files and directories. Represents file types with various colors and icons. Navigates through the file system with the mouse.

Font Editor – Visual tool for editing font appearance and creating new characters and fonts.

Icon Editor – Visual tool for editing icon appearance and creating new icons.

Image Tool – Interactive image viewer. Image Tool can be used to view the contents of file types such as GIF, TIFF, JPEG, PostScript, and others.

Mail Tool – Tool for handling electronic mail.

Performance Meter – Real-time system performance meter that can display a variety of data.

Print Tool – Graphical front-end to the print command. It supports OPEN LOOK *drag-and-drop* file transfer operations.

Shell Tool – Standard OPEN LOOK non-scrolling window terminal emulator. The window behaves like an ASCII character terminal for entry of UNIX commands at a system shell prompt and other terminal operations.

Snapshot – Tool to *snap* or capture a picture of a window or region of a screen in bitmap (raster file) format. Used for capturing screen image displays in this user's guide.

Tape Tool – Graphical tool for controlling the tape driver.

Text Editor – Visual text editor used in OpenWindows tools such as the Mail Tool composition window.

Starting OpenWindows

Checking Your User Environment

Before you log in, your system administrator should set your required user environment variables and corresponding entries in the .cshrc file in your home directory. These system environment variables are essential to using Traditional Chinese features.

.cshrc File

System environment variables need to be set by your system administrator in your .cshrc file. Therefore, the first time you log in, before you start OpenWindows for the first time, check to make sure there are lines such as the following in the .cshrc file in your home directory:

```
setenv LANG zh_TW
setenv OPENWINHOME /usr/openwin

set path=( /usr/SUNWale/bin $OPENWINHOME/bin $path )

...

if ($?USER != 0 && $?prompt != 0) then
   /bin/stty cs8 -istrip defeucw
endif
```

Make sure the LANG variable is set to C (ASCII) or zh_TW (Chinese) *before* the if...endif statement and that *no* prompt is set before the if...endif.

If these lines are not present or are different, contact your system administrator. If you are your own system administrator or an advanced user, refer to *Traditional Chinese Solaris System Administrator's Guide* for further information on setting up your system.

`.openwin-init` File

If you have a `.openwin-init` file in your home directory and might use the Traditional Chinese character input facilities, make sure this file contains an `htt` command, as described in "How and When htt Is Started" on page 15.

`.xinitrc` File

If your system has a `.xinitrc` file in your home directory, make sure it contains at least the lines provided in Traditional Chinese Solaris `$OPENWINHOME/lib/Xinitrc` file.

Starting the Traditional Chinese OpenWindows Environment

After verifying that your system administrator has set your user environment correctly for Traditional Chinese language operation, you are ready to start your Chinese OpenWindows environment as follows:

- Type the following command at the system prompt:

```
system% openwin
```

Changing the Language Setting on the Workspace Properties Worksheet

The Traditional Chinese OpenWindows Workspace Properties worksheet contains a localization field. To change the language setting for the next OpenWindows tools you start:

1. **Choose Properties... on the main window Workspace menu.**
2. **Choose the Locale category on the Category pull down menu.**

   The following screenshots show the English and Traditional Chinese versions of the Workspace Properties worksheet:

You can set the display and input mechanisms of the Traditional Chinese OpenWindows environment by using the Locale field in the Workspace Properties worksheet. You can switch between the U.S. and the Chinese setting.

The Locale setting determines which characters (ASCII, Traditional Chinese) appear in new tool windows when they are started. Changing the locale does not affect the appearance or operation of tool windows that were started up before the change.

Changing Your System Locale Setting

To change the Locale setting on the Workspace Properties worksheet (and in the `.OWdefaults` file as explained in *Traditional Chinese Solaris System Administrator's Guide*):

OpenWindows and `htt`

OpenWindows uses the `.openwin-init` file in the user's home directory if the file exists. If the `.openwin-init` file lacks a line to start `htt`, `htt` will not start when the OpenWindows environment is started.

For the Traditional Chinese input functions in applications to operate as intended, the `.openwin-init` script must start `htt` before the script starts an application that uses `htt` for Traditional Chinese character input. If `htt` is started after the application, only a root-window style input method server window can be used.

Verifying Changes to the `.openwin-init` File

The OpenWindows WorkspaceUtilitiesSave Workspace command writes or rewrites a user's `.openwin-init` file when it saves the current configuration of the workspace screen. Therefore, each time you use the Save Workspace pull-down menu selection, or edit `.openwin-init`, check your `.openwin-init` file and make sure the `htt` command precedes any line that starts an application that takes Traditional Chinese character input.

Placing the `htt` command this way ensures correct connection to `htt` if the OpenWindows system is restarted later. Manual edits to `.openwin-init` will be overwritten the next time you use the Save Workspace command. You can save and use your edits by exiting and then restarting the OpenWindows environment.

Printing in OpenWindows

You can use the Workspace Properties menu to set up the `xetops` print filter.

- Type `cat $FILE | xetops | lp` in the Properties worksheet, as shown in the following figure:

Customizing Your OpenWindows Workspace

Using Fonts

The localized language functions of Traditional Chinese Solaris applications use font sets, or groups of fonts, including both ASCII character fonts and non-ASCII Traditional Chinese character fonts. These font sets are required for Traditional Chinese display. They can be used, as font names are, in customizing your workspace as described in *Traditional Chinese Solaris User's Guide*.

Traditional Chinese (zh_TW) Font Lists

A Traditional Chinese (zh_TW) font list is composed of one English font, representing ASCII characters in CNS11643-0 or ISO8859-1, and a number of Traditional Chinese fonts representing characters such as CNS11643-1, CNS1643-1, CNS11643-2, and CNS11643-3.

Traditional Chinese Solaris provides some default font lists defined in application defaults files in /usr/dt/app-defaults/zh_TW/*. The following is an excerpt from one of these files, Dtwm:

```
Dtwm*icon*fontList: \
-dt-interface system-medium-r-normal-s*-*-*-*-*-*-*-*-*:
```

This portion of the file refers to a font list that contains the following fonts, which are defined in /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/fonts.alias:

```
"-dt-interface system-medium-r-normal-s serif-16-140-75-75-p-70-cns11643-0"
"-dt-interface system-medium-r-normal-s serif-16-140-75-75-p-140-cns11643-1"
"-dt-interface system-medium-r-normal-s serif-16-140-75-75-p-140-cns11643-2"
"-dt-interface system-medium-r-normal-s serif-16-140-75-75-p-140-cns11643-3"
```

The first is the English font for codeset 0 (ASCII) character font display. The rest are Traditional Chinese fonts for codeset 1 (CNS11643) plane 1 character font display, and codeset 2 (CNS11643) plane 2 and plane 3 character font display.

Traditional Chinese (zh_TW.BIG5) Font Lists

A Traditional Chinese zh_TW.BIG5 font list is composed of one English font, representing ASCII characters, and one Traditional Chinese font representing Chinese characters in Big 5.

Traditional Chinese Solaris provides some default font lists defined in an application defaults file in /usr/dt/app-defaults/zh_TW.BIG5/*. Below is a part of one of the files, Dtwm:

```
Dtwm*icon*fontList: \
-dt-interface system-medium-r-normal-s*-*-*-*-*-*-*-*-*
```

This font list contains the following fonts, defined in /usr/openwin/lib/locale/zh_TW.BIG5/X11/fonts/75dpi/fonts.alias:

```
"-dt-interface system-medium-r-normal-s serif-16-140-75-75-p-70-big5-0"
"-dt-interface system-medium-r-normal-s serif-16-140-75-75-p-140-big5-1"
```

The first is an English font for ASCII character font display. The second is a Traditional Chinese Big 5 font.

Starting Applications With a Specific Traditional Chinese Font List

When you start an Asian Solaris tool at the command line, you can also specify its fonts. Below is an example of a command line argument used to start a new Traditional Chinese Windows terminal with a specified font list:

```
system% dtterm -fn  "-dt-interface system-medium-r-normal-s \
serif-16-140-75-75-p-70-cns11643-0; \      -dt-interface
system-medium-r-normal-s \      serif-16-140-75-75-p-140-cns11643-1:"
```

Note the two delimiters used in the font list. The `;` delimiter is used to separate the font names except for the last font name, which ends with the `;` delimiter. (In the example above, `;` follows the English font name, and the `:` delimiter follows the Traditional Chinese font name.) Since there are spaces in the long font names, the font list is enclosed in quotation marks.

▼ Specifying a Font at the Command Line

You can specify which font a Traditional Chinese OpenWindows application will use on a command line. When the current locale is `zh_TW` or `zh_TW.BIG5`, the command uses one of the defined font-set aliases instead (explained in the following section), for example:

```
system% cmdtool -font fontset_name &
```

However, when the current locale is `C`, the command uses a font name and cannot use a font-set alias. The following shows a command using the long name of an ASCII character font:

```
system% cmdtool -font
-misc-fixed-medium-r-normal--9-80-100-100-c-60-iso8859-1 &
```

Font Set Names

The Traditional Chinese OpenWindows environment provides several font sets that combine two or more fonts so that both English and Chinese characters can be used together in one window. Five of the font sets each comprise one Roman font (ASCII characters) in the ISO8859 standard plus a Chinese font specified in CNS 11643-92. A few examples are:

- sung12
- sung14
- sung16
- sung20
- sung24

Each of these font sets is made up of several font files. The `$OPENWINHOME/lib/locale/zh_TW/OW_FONT_SETS/OpenWindows.fs` file defines the full Chinese Solaris font set.

In addition, the Traditional Chinese Solaris operating environment provides TrueType scalable fonts in the following type faces:

- Hei
- Kai
- Ming

These fonts are located in `$OPENWINHOME/lib/locale/zh_TW/X11/fonts/TrueType`

Scaling Applications Windows and Fonts

The `$OPENWINHOME/lib/locale/zh_TW/OW_FONT_SETS/OpenWindows.fs` file also sets the following font size definitions for use in command lines:

- `small`=12 points
- `medium`=14 points (default size)
- `large`=16 points
- `extra_large`=24 points

For example, the following command line shows how to start a Command Tool window that uses 16-point type and is scaled proportionally larger than the default:

```
system% cmdtool -scale large &
```

APPENDIX **B**

Binary Compatibility Package

Applications compiled under Chinese OpenWindows 2.*x* or Solaris 1.*x* or SunOS 4.*x* systems have different binary formats than the current Chinese Solaris release. Older applications can nevertheless be run under the current Chinese 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* /Chinese OpenWindows 2.*x* applications without recompilation, although OpenWindows V2 Chinese 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_basiclocale oldlocale -lc_inputlang oldlocale \
    -lc_displaylang oldlocale
```

The following example shows the command for running the compiled binary code of an earlier version of the `textedit` application on a current Traditional Chinese Solaris release system:

```
system% textedit -lc_displaylang tchinese -lc_basiclocale tchinese \
    -lc_inputlang tchinese
```

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

Traditional Chinese 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 *Solaris Internationalization Guide for Developers*.

The utilities in the following table test various aspects of the Traditional Chinese (CNS 11643) national standard character set. 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 *hctype(3x)* man page.

TABLE B–1

Utility	Description
`ishalpha`	Returns true if it is a Roman character in the CNS 11643 character set.
`ishupper`	Returns true if it is an uppercase Roman character as defined by the CNS 11643 character set.
`ishlower`	Returns true if it is a lowercase Roman character in the CNS 11643 character set.
`ishdigit`	Returns true if it is a number in the CNS 11643 character set.
`ishspace`	Returns true if it is the space character in the CNS 11643 character set.
`ishpunct`	Returns true if it is a punctuation character in the CNS 11643 character set.
`ishparen`	Returns true if it is a left or right parenthesis in the CNS 11643 character set.
`ishphontone`	Returns true if it is a Mandarin phonetic tone.
`ishradical`	Returns true if it is a Chinese character radical.
`ishline`	Returns true if it is a ruled line symbol in the CNS 11643 character set.
`ishunit`	Returns true if it is a unit character in the CNS 11643 character set.
`ishsci`	Returns true if it is a scientific symbol in the CNS 11643 character set.
`ishgen`	Returns true if it is a general symbol in the CNS 11643 character set.
`ishgreek`	Returns true if it is a Greek character in the CNS 11643 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 or Chinese character sets. Asian Solaris software implements a wide character library for handling Korean or Chinese character codes according to industry standards.

Routines that have Korean or Chinese language-specific dependency are in their own language-specific library, which is linked with the corresponding C compiler option. In Traditional Chinese Solaris, `libhle` is linked with `-lhle`. 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 Solaris software defines in `wchar.h` as follows:

```
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, for example:

- Characters within a codeset, such as converting uppercase ASCII to lowercase.
- Between different conventions for national standard character sets, such as:
  - Between Combination and Completion code, both KS C 5601-1987 and KS C 5601-1992.
  - Between GB and EUC.
  - Between CNS 11643 code and Big5.
- Between code formats (such as EUC and WC).

Programs using the general multibyte conversion utilities should include the header files `widec.h` and `wctype.h`. Traditional Chinese Solaris specific routines (such as `ish`*xxx*) are declared in `zh_TW/xctype.h`.

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

As with classification functions described in the previous section, the use of these utilities can be controlled by the `setlocale` function. Locale-specific routines are contained in a locale-specific library. For Traditional Chinese, that library is `libhle`. The library can be linked during compilation using the C compiler option `-lhle`.

Conversion Within a Codeset

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 *hconv(3)* for Traditional Chinese.

The following routines are in the regular Chinese C library:

TABLE B–2

Function	Description
`tohupper()`	Converts codeset 1 Roman lowercase to uppercase.
`tohlower()`	Converts codeset 1 Roman uppercase to lowercase.

Conversion for Traditional Chinese Character Codes

The following routines perform character-based code conversion on the CNS-11643 character set. They convert characters in the set between CNS-11643, EUC, and Big5 formats. To use these routines, the library `hle` must be linked using the C compiler option `-lhle`. For more information, see the *hconv(3x*) man page.

TABLE B–3

Function	Description
`cbig5toeuc()`	Converts Big5 character to EUC.
`ccnstoeuc()`	Converts CNS character to EUC.
`ceuctobig5()`	Converts EUC character to Big5.
`ceuctocns()`	Converts EUC character to CNS.

TABLE B–4

Function	Description
`big5toeuc()`	Converts Big5 string to EUC.

TABLE B–4 *(Continued)*

Function	Description
`cnstoeuc()`	Converts CNS string to EUC.
`euctobig5()`	Converts EUC string to Big5.
`euctocns()`	Converts EUC string to CNS.

APPENDIX **C**

Running Networked Applications

You can run Traditional Chinese localized applications on a remote machine as explained in *Solaris Advanced User's Guide*. That guide also describes the advanced features of the OpenWindows environment that enable you to run applications that reside on another machine on your network.

Applications running on Traditional Chinese OpenWindows 2.*x* can be displayed remotely on a Traditional Chinese OpenWindows 3.*x* system. However, Traditional Chinese OpenWindows 3.*x* applications may not display remotely in a Traditional Chinese OpenWindows 2.*x* environment because not all 3.*x* system fonts are available in 2.*x* environments.

Note – The `zh_TW.BIG5` locale does not support running networked applications as described in this appendix.

Note – Most users do not need to read this appendix. If you want to explore running networked applications, ask your system administrator about special applications possibly available on your network.

Instructions for Running Networked Applications

To run networked applications, you must follow the information and directions in *Solaris Advanced User's Guide*. You must additionally make the following adjustments to enable operation of the Traditional Chinese features of the Chinese OpenWindows environment.

Setting Required Environment Variables

To run a networked application on a remote machine, you must set your environment variables correctly, as follows:

- The `DISPLAY` environment variable in your shell on the remote machine must be set to your local screen.
- The `LANG` environment variable in your shell on the remote machine must be set to `zh_TW`.
- If the OpenWindows libraries have not been installed in the standard `/usr/lib` or `/usr/local` shared library directories, you must set the `LD_LIBRARY_PATH` environment variable to the appropriate directory (`$OPENWINHOME/lib`).

Sample Command Sequences for Remote Operation

▼ Displaying Remote OpenWindows 2.*x* on a Local OpenWindows 3.*x* System

The following sequence of commands shows how to start a Shell Tool on a remote machine running Traditional Chinese Solaris 1.*x* (including Traditional Chinese OpenWindows 2.*x*). In this example, the local machine is running Solaris 2.*x*, including

Traditional Chinese OpenWindows 3.*x*, and the remote machine, used to run the Shell Tool, has Traditional Chinese Solaris 1.*x*, including Traditional Chinese OpenWindows 2.*x*:

1. **Type the following command.**

   ```
   local_machine% xhost +remote_machine
   ```

2. **Log in to the remote machine.**

   ```
   local_machine% rlogin remote_machine
   ```

   or

   ```
   local_machine% telnet remote_machine
   ```

3. **Set the language locale on the remote machine.**

   ```
   remote_machine% setenv LANG tchinese
   ```

4. **Set the remote machine to display on your local machine.**

   ```
   remote_machine% setenv DISPLAY local_machine:0.0
   ```

5. **Set the OpenWindows environment home directory.**

   ```
   remote_machine% setenv OPENWINHOME /usr/openwin
   ```

6. **Set the path to the OpenWindows LD library.**

   ```
   remote_machine% setenv LD_LIBRARY_PATH $OPENWINHOME/lib
   ```

7. **Start up the OpenWindows application, for example Shell Tool:**

   ```
   remote_machine% $OPENWINHOME/bin/xview/shelltool -lc_basiclocale \oldlocale
    -lc_inputlang oldlocale
    -lc_displaylang oldlocale
   ```

Note – The status region does not show which Chinese input method is used.

To run a different application, use that application's command in place of `shelltool`.

APPENDIX D

Mapping Traditional Chinese Keyboard Functions

This appendix shows how to configure a Sun Chinese keyboard to make selected key functions when you need them.

The Chinese Type-4 keyboard does not have a Compose key. You can use the `xmodmap` command to make any key function as Compose. In the following example, the right Meta key () is set to function as the Compose key:

```
system% xmodmap -e "remove mod1 = Meta_R"
system% xmodmap -e "keysym Meta_R = Multi_key"
```

See the `$OPENWINHOME/share/etc/keytables/Taiwan4.kt` file and the *xmodmap(1)* man page for more information.

The Chinese Type-5 keyboard does not have an AltGraph key. You can use the `xmodmap` command to make any key function as AltGraph. In the following example, the right Meta key () is set to function as the AltGraph key:

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

See the `$OPENWINHOME/share/etc/keytables/Taiwan5.kt` file and the *xmodmap(1)* man page for 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, `wchar_t`, 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.

BIG5 A commonly used code set in Taiwan.

Category In the Traditional Chinese 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 *month*, *day*, *year*; while in another country it might be *day*, *month*, *year*. 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 CNS 11643 character set contains English, Greek, and Chinese character sets in addition to Chinese radicals and many other characters.

CNS

Taiwan's Chinese National Standard. This is the Taiwan analogue to ASCII. In this document set, it refers to the code set defined by CNS 11643. It contains the Chinese characters, phonetic symbols and radicals, control codes, punctuation, and western alphabets, including Roman and Greek characters. Each character is two bytes long, with the highest or most significant bit of each byte set to zero. In other words, it uses the lower seven bits of each byte. Due to the size of the Taiwan Chinese character set, they are divided into multiple codeplanes, with the default plane containing the most commonly used characters. ISO 2022 provides mechanisms for shifting from one codeplane to another.

After its revision in 1992, CNS 11643 defines 48,000 characters, which are divided among codeplanes 1-7, Codeplanes 8-16 are undefined, but are included in the code set architecture. Codeplanes 1 and 2 (common and rarely used characters) are unaffected by the revision. Characters that were in codeplane 14, a provisional user-defined plane, have been standardized into codeplane 3, with the overflow in codeplane 4.

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.

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 codesets 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. Codeset 0 is also called the primary code set, and codesets 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.

EUC-CNS

The EUC representation of CNS 11643. For Codeset 1, this is the normal CNS code with a one in the most significant bit of each byte. In other words, EUC-CNS equals CNS plus 0x8080. For example, the CNS character 0x212A becomes the EUC-CNS character 0xA1AA. Or in binary, 00100001 00101010 becomes 10100001 10101010. For Codesets 2 and 3, characters are also prefixed by single shift bytes SS2 and SS3. In addition, codeset 2 requires a codeplane byte. The code of a codeset-2 character is SS2 followed by codeplane byte followed by EUC-CNS.

The codeplane byte is plane number added to 0xA0; for example plane 2 has codeplane bye 0xA2.

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.

Locale A locale describes a language or cultural environment. Its setting affects the display or manipulation of language-dependent features. Traditional Chinese Solaris software provides `C` for U.S.A, `zh_TW` for Traditional Chinese extended UNIX code, and `zh_TW.BIG5` for the Traditional Chinese Big5 locale.

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.

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 (codeset 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 *X/Open System Interface Portability Guide*.

Index