



# Simplified Chinese Solaris User's Guide

---

Sun Microsystems, Inc.  
901 San Antonio Road  
Palo Alto, CA 94303  
U.S.A. 650-960-1300

Part Number 806-3487-10  
March 2000

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

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

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

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

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

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

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

---

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 Etats-Unis. Tous droits réservés.

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

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

Sun, Sun Microsystems, le logo Sun, SunSoft, SunDocs, SunExpress, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

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



# Contents

---

	<b>Preface</b>	<b>9</b>
<b>1.</b>	<b>Introduction</b>	<b>13</b>
	Design of the Simplified Chinese Solaris Software	13
	Simplified Chinese Graphical User Interface	14
	Simplified Chinese Input/Output	14
	The Localized CDE Desktop	14
<b>2.</b>	<b>Starting the Simplified Chinese Solaris Software</b>	<b>17</b>
	Introduction	17
	Setting the Default Locale	18
	Using Aliases to Set Locales	18
	Changing Font Directories	19
<b>3.</b>	<b>Using the <code>htt</code> Input Method Server</b>	<b>21</b>
	Introduction	21
	Input Method Server Basic Properties	22
	Three <code>htt</code> Processes	22
	<code>htt</code> Application Display	22
	<code>htt</code> Property Manager	23
	Resetting and Terminating <code>htt</code>	23
	Customizing the on/off Key From General Dialog	24

	Customizing the Preedit/Status Window From the Preedit/Status Dialog	25
	Customizing the Lookup Window From the Lookup Table Dialog	26
	How and When <code>htt</code> Is Started	27
	Using the <code>iiim</code> Server in <code>zh.UTF-8</code> and <code>zh_CN.UTF-8</code> Locales	27
<b>4.</b>	<b>Entering Simplified Chinese Text</b>	<b>29</b>
	About This Chapter	29
	Input Window Areas	29
	Preedit Area	30
	Status Area	30
	Lookup Choice Area	30
	Input Methods and Conversion Modes for Entering Text	31
	Typing ASCII Text	32
	Typing Non-ASCII Text	32
	Location Code Mode	33
	Typing Location Code Text	33
	Pinyin Mode	35
	Typing Pinyin Text	36
	New QuanPin and New ShuangPin Input Methods	37
	▼ Defining Phrases for Later Use	38
	▼ Selecting Frequently-Used Candidates	39
	Other Features	39
	Keyboard Definition	41
	Dictionary Files	43
	New ShuangPin Features	43
	New ShuangPin Keyboard Mapping	44
	Quanpy Input Method	48
	Stroke Input Mode	50

Typing Stroke Mode Text	53
Golden Input Mode	55
Typing Single Simplified Chinese Text	56
Typing Phrase Text	60
Intelligent Pinyin Conversion Mode	62
Typing Intelligent Pinyin Text	62
Intelligent Pinyin Shortcuts	65
Simplified Chinese Symbol Input Mode	65
GBK Code Input Method	66
Typing GBK Code Text	66
Japanese Input Method	67
Typing Kanji Text	67
Hanja Input Method	69
Typing Hanja Text	69
Zhuyin Input Method	71
Unicode Input Method	74
Typing Unicode Text	74
Input Method Properties Editor	75
▼ Changing Input Method Properties	76
▼ Changing Lookup Choice Properties	77
<b>5. Code Table Input Method Interface</b>	<b>79</b>
Introduction	79
▼ Creating a Code Table	79
<b>6. Localized Applications</b>	<b>85</b>
About This Chapter	85
Using the <code>mailx</code> Utility With Simplified Chinese Characters	85
Using <code>talk</code> With Simplified Chinese Characters	86
The <code>xtobdf</code> Utility	86

	Convtool	86
	iconv	87
<b>7.</b>	<b>Fonts</b>	<b>91</b>
	About This Chapter	91
	Display PostScript System (DPS)	91
	Using Simplified Chinese PostScript Fonts and DPS Facilities	91
	Editing Fonts With <code>fontedit</code>	95
	Setting Up a Working Directory	95
	Starting the Font Editor	96
	Preparing a Font File	96
	Editing a BDF Font File	97
	Converting BDF to PCF Format	99
	Installing and Checking the Edited Font	99
	Creating Characters With <code>sdtudctool</code>	100
	▼ Using <code>sdtudctool</code>	100
	▼ Specifying Options	102
	▼ Viewing the Character Reference	103
	▼ Typing the New Characters	104
<b>8.</b>	<b>Simplified Chinese Printing Facilities</b>	<b>105</b>
	Introduction	105
	Printing Chinese Output From a Command Line	106
	Printing With a Line Printer	106
	Printing With the <code>xetops</code> Utility	106
	Printing with the <code>mp</code> Utility	108
<b>A.</b>	<b>Open Windows Information</b>	<b>111</b>
	Introduction to OpenWindows	111
	Design of Simplified Chinese Solaris Software	111
	Simplified Chinese OPEN LOOK DeskSet	112

Starting OpenWindows	113
Checking Your User Environment for the First Time	113
Starting the Simplified Chinese OpenWindows Environment	114
Using Localization on the Workspace Properties Worksheet	114
OpenWindows and <code>htt</code>	118
Changes to the <code>.openwin-init</code> File	119
Printing in OpenWindows	119
Customizing Your OpenWindows Workspace	120
Using Fonts	120
Simplified Chinese Font Sets	120
▼ Specifying a Font on the Command Line	121
Scaling Application Windows and Fonts	122
<b>B. Backward Compatibility</b>	<b>123</b>
Simplified Chinese Test Utilities	123
Simplified Chinese Conversion Utilities	125
Conversion Utilities	125
Conversion for Simplified Chinese Character Codes	127
Binary Compatibility Package (BCP)	128
Running Earlier Binary Code	128
<b>C. Running Networked Applications</b>	<b>129</b>
Introduction	129
Instructions for Running Networked Applications	129
Setting Required Environment Variables	130
Sample Command Sequences for Remote Operation	130
<b>Glossary</b>	<b>133</b>
<b>Index</b>	<b>137</b>





# Preface

---

---

## About This Book

*Simplified Chinese Solaris User's Guide* describes product behavior unique to the Simplified Chinese Solaris™ operating environment and answers many questions commonly asked about the software. This guide introduces you to the general appearance and properties of a variety of localized Desktop Tools™ and utilities offered with the Simplified 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 Simplified Chinese features in the Solaris operating environment to manage files, calendar, and e-mail, write or print Simplified Chinese files, and so forth. Tools for these and many other applications run under Simplified 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 Simplified Chinese Solaris software before
- If you need information on using product features
- If you need instructions for starting the Simplified Chinese Solaris operating environment (see especially Chapter 2)

---

# Before You Read This Book

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 11. This user’s guide focuses on using the Simplified Chinese features of the Desktop Tools and other features of Simplified Chinese Solaris software.

---

## How This Book Is Organized

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

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

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

Chapter 3 introduces the startup, appearance, and use of `http`.

Chapter 4 describes different Simplified 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 *Solaris Internationalization Guide for Developers* and *Simplified Chinese Solaris System Administrator’s Guide*.)

Chapter 5 describes the interface that allows the user to add new input.

Chapter 6 describes uses of two desktop tools localized for Chinese `mailx`, `talk`, and tools to convert file codes.

Chapter 7 explains how to customize fonts used in your Simplified Chinese Solaris applications.

Chapter 8 discusses Simplified Chinese Solaris support for line printers with built-in Chinese fonts or using the Simplified Chinese Solaris `xetops` filter package.

Appendix A describes the special requirements of the OpenWindows environment.

Appendix B discusses running compiled binary code of earlier SunOS™ 4.x/Solaris 1.x/Asian OpenWindows 2.x applications without recompilation.

Appendix C discusses running localized applications that reside on another machine across your network.

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

---

## Related Books

You should become familiar with the following basic documentation:

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

Advanced users may want to read *Solaris Advanced User's Guide*. Advanced users who want to customize their system environment or the operations of their Sun tools will find pertinent information in *International Language Environments Guide* and *Simplified Chinese Solaris System Administrator's Guide*. These books contain 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.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% You have mail.</code>
AaBbCc123	What you type, contrasted with on-screen computer output	<code>machine_name% su</code> <code>Password:</code>

TABLE P-1 Typographic Conventions (continued)

Typeface or Symbol	Meaning	Example
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> 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.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	<code>machine_name%</code>
C shell superuser prompt	<code>machine_name#</code>
Bourne shell and Korn shell prompt	<code>\$</code>
Bourne shell and Korn shell superuser prompt	<code>#</code>

# Introduction

---

---

## Design of the Simplified Chinese Solaris Software

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

Simplified 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 Simplified Chinese Solaris software. These products introduce Solaris input methods for the input and output of Simplified Chinese. Application programs and CDE use the features of Simplified Chinese CDE to communicate with users in Simplified Chinese.

This Simplified 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 (`dttwm`) that communicate with users in Simplified Chinese.
- A localized development environment, which programmers use to develop localized applications, with Xlib and Motif, which have been internationalized for this use. Programmer and developers should refer to *Solaris Internationalization Guide for Developers*.

---

# Simplified Chinese Graphical User Interface

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

## Simplified Chinese Input/Output

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

Simplified Chinese input from 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. Each tool can accept and display Simplified Chinese text. 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.
- App Manager – Contains the applications available on your system. You can launch these applications through the App 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.
- 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<sup>®</sup> commands at a system shell prompt and other terminal operations.





# Starting the Simplified Chinese Solaris Software

---

---

## Introduction

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

The Simplified Chinese Solaris operating environment provides two window environments: Common Desktop Environment (CDE) and OpenWindows. CDE is a fully internationalized environment; it does not require most of the administration tasks that the OpenWindows environment requires to handle Simplified Chinese. For information on starting OpenWindows, see Appendix A. The new multilingual Solaris product includes the following Simplified Chinese locales:

- `zh` – Simplified Chinese environment in extended UNIX code (EUC) that supports GB2312-80.
- `zh.GBK` – Simplified Chinese environment in GBK, an extension of GB2312-80 (that is, Guo Biao Kuo in Chinese PinYin). GBK contains all of the CJK characters that are in Unicode 2.1.
- `zh_CN.EUC` – Symbolic link to `zh` locale.
- `zh_CN.GBK` – Symbolic link to `zh.GBK` locale.
- `zh.UTF-8` – New in Solaris 8, Simplified Chinese environment in Unicode 3.0.
- `zh_CN.UTF-8` – Symbolic link to `zh.UTF-8` locale.

# Setting the Default Locale

You can change your default locale using the following procedure.



1. Choose the language button on the dtlogin window.

2. Select the C, zh, zh.GBK or zh.UTF-8 locale.

Your new locale is in effect.

## Using Aliases to Set Locales

You can use aliases to change a terminal-emulation window between the Simplified Chinese 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 chn "export LANG=zh; /bin/stty cs8 -istrip defeucw"
alias asc "export LANG=C; /bin/stty cs7 istrip defeucw"
```

## Setting Up Locale Aliases for the C Shell

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

```
alias chn "setenv LANG zh; /bin/stty cs8 -istrip defeucw"
alias asc "setenv LANG C; /bin/stty cs7 istrip defeucw"
```

---

## Changing Font Directories

The Xsession script (in `/usr/dt/bin`) that comes with the Simplified Chinese operating environment includes the following font path:  
`/usr/openwin/lib/locale/zh/X11/fonts`. To add a different font directory path dynamically, type:

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



## Using the `htt` Input Method Server

---

---

### Introduction

The `htt` input method server handles Simplified Chinese input for Simplified Chinese Solaris software. The `htt` server receives keyboard input and makes the input available as Simplified Chinese characters to Simplified 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 method server. Normally, you do not need to change any `htt` settings, as discussed in this chapter, to operate any Simplified Chinese Solaris application.

You need to be familiar with the input method terminology in the Chapter 4 before reading this chapter. However, to create Simplified Chinese text using Solaris applications, you do not need to read beyond the first section of this chapter.

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



# Input Method Server Basic Properties

The Simplified 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 27.

Simplified Chinese Solaris applications that use `htt` for Simplified Chinese character input typically find `htt` running when they are started. To get `htt`'s service for language input, `htt` should be running before an application is started. If `htt` is not running when an application is started, the application may not be able to get the input service, even if `htt` is started later.

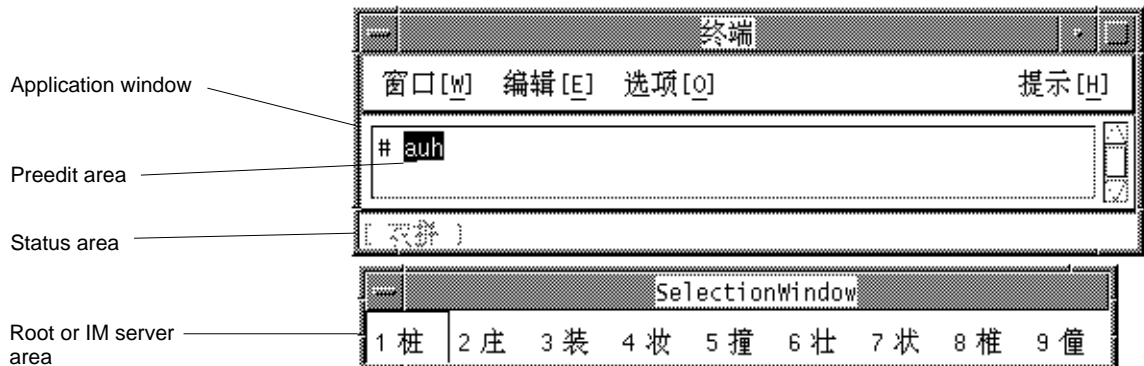
## Three `htt` Processes

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` start also.

## `htt` Application Display

The following figure shows a typical application display when using the `htt` input method.



---

# 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

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 pop-up.
- 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.**

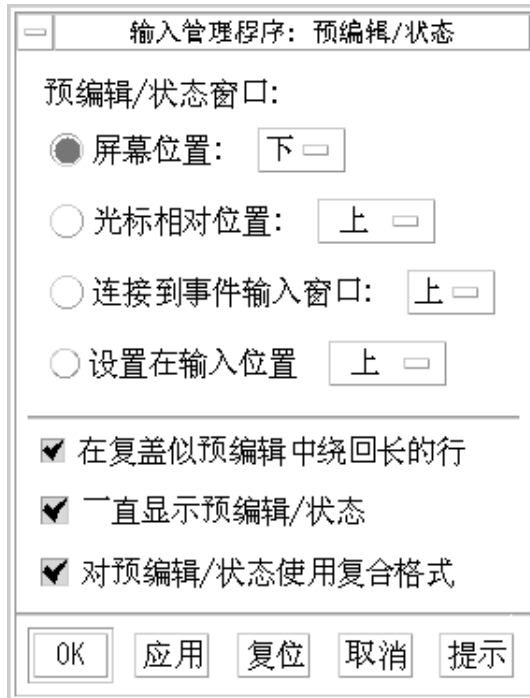
---

**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.

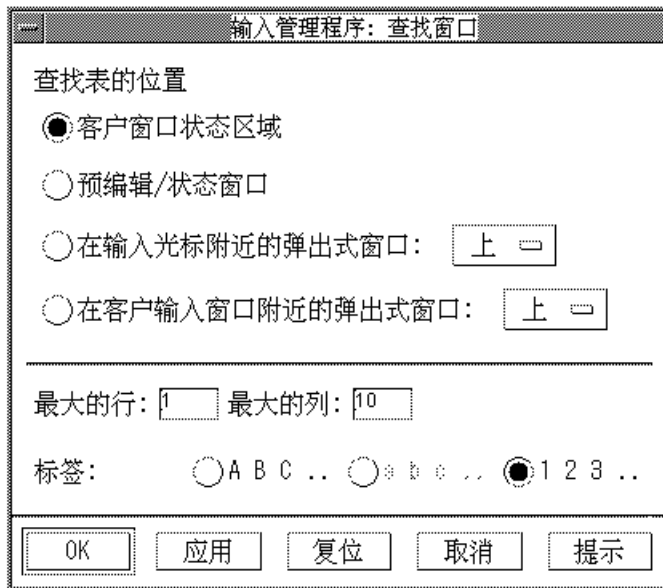


1. Selecting “Position on the screen” places the window in a fixed location of the workspace. This window is shared by multiple applications whose input style is root window style.
2. Selecting “Position relative to the cursor” places the window close to the mouse cursor.
3. Selecting “Attach to client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

4. Selecting “Use Composite Layout for Preedit/Status” uses the new layout in root window style (the preedit, status, and lookup choices are in the same window). Deselecting this option causes the lookup choices to appear in a separate 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 to the end of a line. In the 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
- Pop-up window near cursor – places the pop-up near to the current position of mouse

- Pop-up window near client input window – places the pop-up 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 pop-up. You can specify the labels used for lookup choices. The options are uppercase letters or numbers (except in New QuanPin and New ShuangPin, which support only numbered labels).

---

## How and When `htt` Is Started

Any application that uses the `htt` input method server to receive Simplified Chinese characters must find `htt` running before it can receive input. 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`, `htt` starts automatically when the Simplified Chinese windowing environment is started. 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 when a user logs in from `dtlogin`. This script ensures that `htt` is started – before other applications in CDE.

---

## Using the `iim` Server in `zh.UTF-8` and `zh_CN.UTF-8` Locales

The Internet Intranet Input Method Server (`iim`) 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:

- Cyrillic
- Greek

- Thai
- Arabic
- Hebrew
- Unicode Hex
- Unicode Octal
- Lookup
- Simplified Chinese

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

- Japanese — if `ja` locale is installed
- Korean— if `ko` locale is installed.
- Traditional Chinese — if `zh_TW` locale is installed.

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

- F2 — Unicode
- F8 — New Pinyin
- F11 — New Shuang Pin
- F7 — Zhuyin
- F4 — Japanese
- F6 — Korean Hanja

For more detailed information about Input Methods, see “Chapter 4, Entering Simplified Text”.

# Entering Simplified Chinese Text

---

---

## About This Chapter

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

- ASCII/English characters
- Simplified Chinese ideograms (characters or symbols representing ideas/actions/things)
- Special symbol characters

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

- In terminal emulation (TTY) windows such as Terminal
- In text entry subwindows such as used by Text Editor or Mailer
- In control panel subwindows such as used by File Manager for typing a file name
- In other special use (sometimes pop-up) subwindows

For information about creating your own input method, see Chapter 5.

---

## 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 (such as inverse 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 (for example, reverse video and underlined) preedit area displays characters as they are typed or converted. It holds formations of text before they are converted to Simplified Chinese characters or symbols and put in the text block being assembled for the application.

## Status Area

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

## Lookup Choice Area

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

---

# Input Methods and Conversion Modes for Entering Text

The following input methods and conversion modes are available for entering ASCII/English, Simplified Chinese, and other text:

In `zh`, `zh.GBK` and `zh.UTF-8`:

- New QuanPin input method, the default method (F8)
- New ShuangPin input method (F11)
- Quanpy input method (F5 or Control-Q with input conversion on)

In `zh`:

- Input conversion off (Control-spacebar toggle); for ASCII input
- Location input method (F2 or Control-A with input conversion on)
- Pinyin input method (F3 or Control-P with input conversion on)
- Stroke input method (F4 or Control-B with input conversion on)
- Golden input method (F6 or Control-J with input conversion on)
- Intelligent Pinyin input method (F7 or Control-S with conversion on)
- Simplified Chinese Symbol input method (F12 or Control-T)

In `zh.GBK`:

- GBK Code input method (F2)
- Japanese input method (F4)
- Hanja input method (F6)
- Zhuyin input method (F7)
- Unicode input method (F10)

In `zh.UTF-8`:

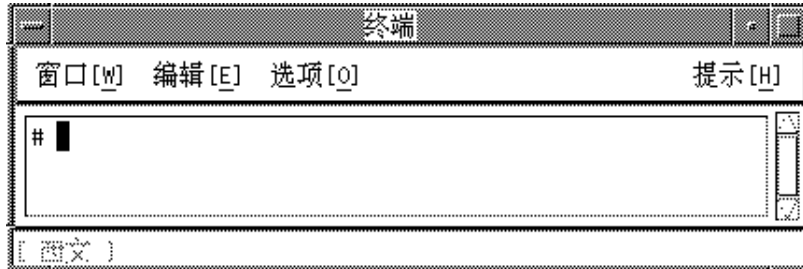
- Unicode input method (F2)
- Japanese input method (F4)
- Korean Hanja input method (F6)
- Zhuyin input method (F7)

Press Control-spacebar to toggle on or off the Simplified Chinese input conversion. The Function and Control keys listed (for example, F6, F7, and Control-A) turn on the corresponding input methods. The input conversion modes listed above are discussed in the following sections.

In the zh.GBK locale, you can modify input method properties, such as the Page Up and Page Down keys, candidate selection keys, and GBK switch keys, using the Properties Editor tool.

## Typing ASCII Text

Each tool first starts with all Simplified Chinese input modes off, the window's status area blank. This mode is for typing ASCII text:



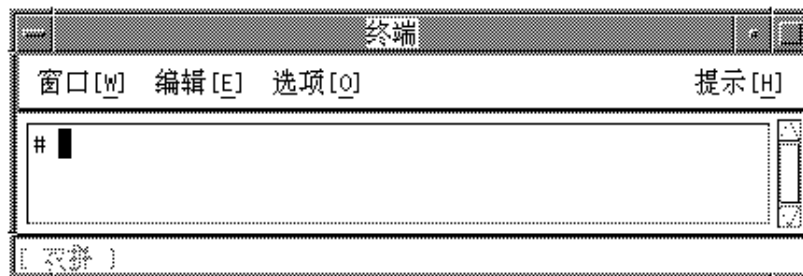
Simplified Chinese input conversion mode is toggled on and off by pressing Control-spacebar. After Simplified Chinese input conversion has been turned on once and input conversion is then turned off, the status area is no longer blank, but instead shows that conversion is off.

## Typing Non-ASCII Text

Each tool starts ready to receive only ASCII text, as shown by a blank status area.

- ◆ **To type Simplified Chinese or other non-ASCII symbols, press Control-spacebar to toggle on Simplified Chinese input conversion.**

The status area then shows conversion on:



After you turn on input conversion, you must turn on one of the input modes (such as Location or Stroke) by pressing the appropriate Function key or Control key. You



can then enter Simplified Chinese characters. To change from one input mode to another, press the appropriate Function key or Control key.

When you toggle off input conversion (by pressing Control-spacebar) the Conversion Manager notes which input mode is on. Then when you later toggle the conversion mode back on, the Conversion Manager automatically turns on the last input mode.

Some input methods offer a lookup choice page for symbol/character selection. Most input methods support pressing the period (.) or comma (,) keys to page among the available input method choices. Special keyboard definitions are documented with their input methods.

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

---

## Location Code Mode

Location mode uses the location code defined by national standard GB2312-80 of the People's Republic of China. This standard lists 7,445 Hanzi characters and other symbols organized in a 94 by 94 grid.

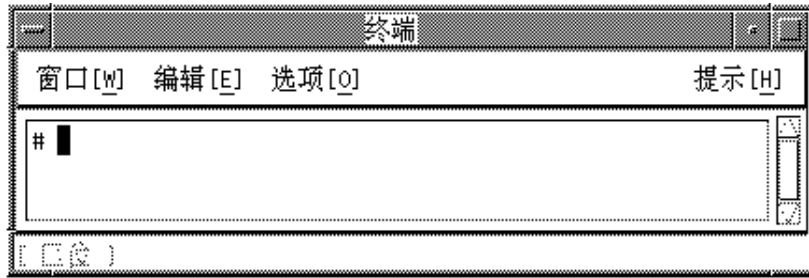
Of these characters, 6,763 are characters used often in Chinese daily life. Level 1 contains 3,755 characters and Level 2 contains 3,008 characters. The Level 2 Hanzi characters include Hanzi radicals, Roman characters, Greek and Cyrillic characters, Arabic and Roman numerals, and miscellaneous symbols.

Each Simplified Chinese character or symbol is identified by a four-decimal-digit location code composed of its row and column numbers. The first two digits are its row number, and the last two are its column number.

## Typing Location Code Text

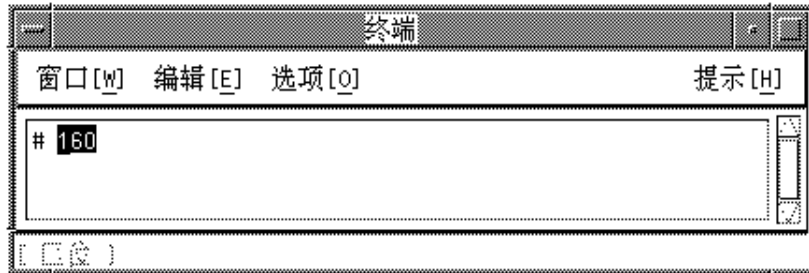
To use these location codes to type Simplified Chinese, proceed as shown in the following example:

- 1. In a new Terminal (for example), which first appears with a blank status area, turn Simplified Chinese input conversion on by pressing Control-spacebar.**
- 2. Turn on Location code input mode by pressing Control-A or F2.**  
The status area then shows Location mode is on.



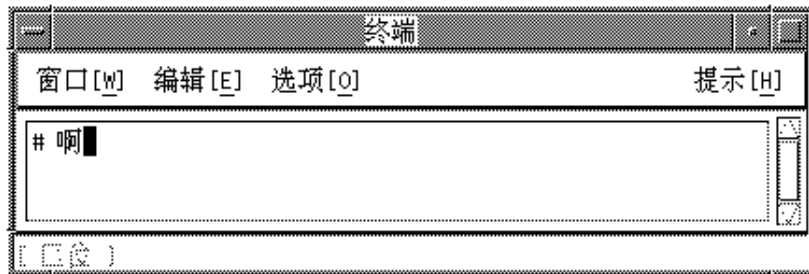
3. Type in the first three of the four digits that represent the character to display (in this example, 1601).

The digits remain visible in the preedit area.



4. Type the fourth digit.

The character automatically replaces the preedit area:



# Pinyin Mode

Pinyin mode uses keyboard mapping that reduces the number of keystrokes needed to type Pinyin input. Pinyin mode maps Pinyin phonetics to single lowercase Roman letters. This mapping includes combinations of sounds mapped to a single keyboard letter so more phonemes can be typed with fewer keystrokes.

The following keyboard figure and table explain the Pinyin mode map. The `clekbd` command displays this Pinyin keystroke map on the screen.

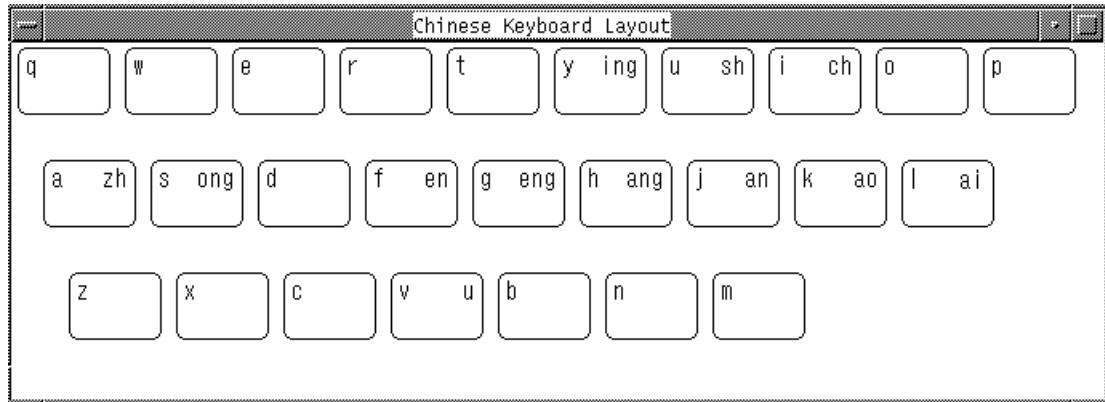


TABLE 4-1 Pinyin Phonetics and Appropriate Keystrokes

Pinyin Phonetics	Pinyin Mode Keystroke
ai	l
an	j
ang	h
ao	k
ch	i
en	f
eng	g
ing	y
ong	s

TABLE 4-1 Pinyin Phonetics and Appropriate Keystrokes (continued)

Pinyin Phonetics	Pinyin Mode Keystroke
sh	u
u	v
zh	a

## Typing Pinyin Text

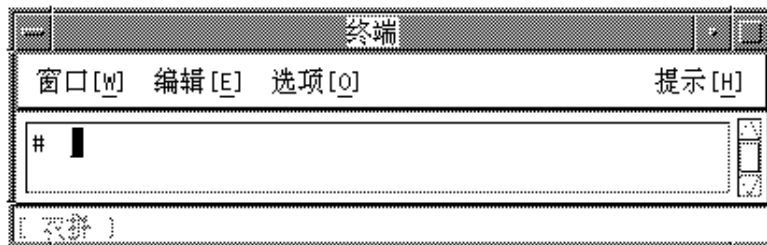
Pinyin input mode uses up to four keystrokes to type one Simplified Chinese character.

A lookup choice area of characters that match the Pinyin input is displayed with each keystroke. When more than one page of choices is available, typing . or > displays the next page moving forward through the lookup choices and typing , or < displays the next page moving backward.

You select the character you want by typing the upper case Roman letter corresponding to that character in the lookup choice area on the screen.

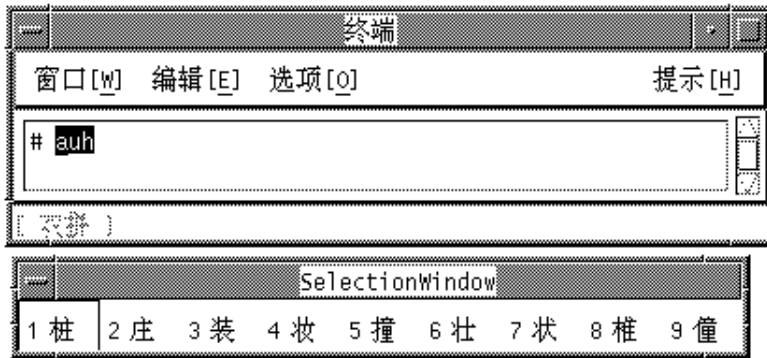
- ◆ **To use Pinyin input mode, make sure input conversion is on and press F3 or Control-P to turn on Pinyin mode.**

The status area shows Pinyin mode is on:



The following is an example of using Pinyin mode by creating the character representing the Pinyin “zhuang.” This Pinyin word simplifies to the three Pinyin mode keystrokes auh. Type and convert them as follows:

1. **Type the text** auh.



2. Type 3 to select the corresponding Simplified Chinese character in the lookup choice listing.

Your choice is substituted for the Pinyin string in the preedit area.




---

## New QuanPin and New ShuangPin Input Methods

This section describes the features in the New QuanPin and New ShuangPin input methods, and how to use some of the features in the zh and zh.GBK locales.

PinYin is a popular input method in PRC, and there are various PinYin-based input methods. Two of them, New QuanPin and New ShuangPin, contain the following features:

- Supports user-defined phrases for later use.
- Dynamically adjusts the frequency of candidates.
- Accepts PinYin strings up to 222 characters long.

- Supports input by ShengMu.
- Supports GBK Chinese.

These features are described in detail in the following sections.

## ▼ Defining Phrases for Later Use

The following example shows how to define the phrase “ke lin dun” and store it for later use.

### 1. Type the phrase `kelindun` without spaces.

The New QuanPin and New ShuangPin input methods will insert spaces for you automatically.



### 2. Type the number representing the first character you want to select.

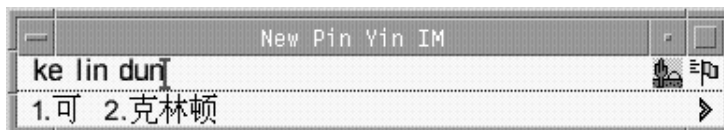
The following example shows the second character selected.



### 3. Select Chinese for the second and third parts of the phrase.



The new phrase is defined and added to the user dictionary file. The next time you type `ke lin dun`, you will see the phrase you defined in the candidate area.



## ▼ Selecting Frequently-Used Candidates

In these input methods, candidates that have been selected are moved to the start of the list to facilitate repeated use.

### 1. Type `sh yi`.

Notice the order of the five available candidates.

### 2. Select the fifth candidate.



### 3. Type `sh yi` again.



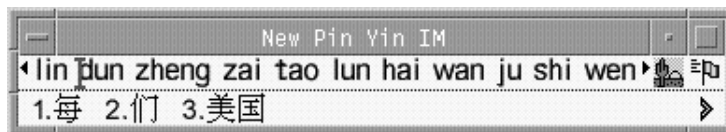
Notice that the fifth candidate has moved to the first position because you previously selected it. Frequently-used candidates are promoted for faster selection.

## Other Features

### Typing Long PinYin Strings

The New QuanPin input methods accepts PinYin strings up to 222 characters long. The following illustrations use the string below:

```
>>meiguozhongtongkelindunzhengzaitaolunhaiwanjushiwenti<<
```





The result is the following Chinese string:

美国总统克林顿正在讨论海湾局势问题

---

**Note** - The New ShuangPin input method supports up to 30-character strings.

---

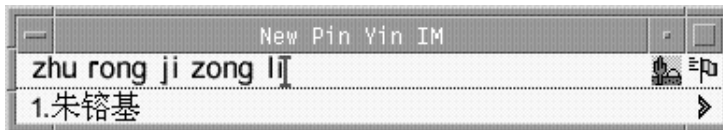
## Typing ShengMu

You can also type ShengMu only. Candidates are supplied for ShengMu, as shown in the following illustrations:



## GBK Support

The zh.GBK locale supports GBK by default, as shown in the following illustrations:



The second Chinese character in the following illustration is defined only in the GBK standard.

朱镕基总理



Single GBK candidates are placed at the end of the list of candidates. Press Return to scroll to the GBK area. For easier selection next time, you can define the GBK candidate as a phrase (for more information, see “Defining Phrases for Later Use” on page 38). Once a phrase is defined, you can insert it easily, as shown in the following illustrations:



Both New QuanPin and New ShuangPin support GBK Hanzi by default in the zh.GBK locale. However, because several Hanzi have the same ShengMu (the first part of Pinyin), New QuanPin and New ShuangPin do not display GBK candidates if you provide only the ShengMu.

For example, typing the string `rong` will display GBK candidates because it is a complete Pinyin string. However, typing `r` alone will not display any GBK candidates because it is only a ShengMu.

## Keyboard Definition

### Edit Keys

The following table shows the definitions of the edit keys.

---

**Note** - The preedit line is a normal X text field.

---

**TABLE 4-2** Edit Key Definitions

<b>Key</b>	<b>Definition</b>
[a-z]	PinYin character
Home	Moves to the start of the preedit line
End	Moves to the end of the preedit line
Left	Moves the caret in the preedit line to the left. If left is Hanzi, the original PinYin is recovered.
Right	Moves the caret in the preedit line to the right.
Delete	Deletes the PinYin character following the caret on the preedit line.
Backspace	Deletes the PinYin character preceding the caret on the preedit line.

## Page Scroll Keys

The candidates of a Pinyin string belong to the following groups:

- G1 – Highest frequency Hanzi + Long (3 or more) Cizu + Double Chinese Cizu
- G2 – GB Single Hanzi
- G3 – GBK Single Hanzi (in the zh.GBK locale)

Some Pinyin strings may have more candidates than can be displayed in the same window. In that case, use the keys described in the following table to scroll through the candidates.

**TABLE 4-3** Page Scroll Key Definitions

<b>Key</b>	<b>Definition</b>
- =	Scrolls to previous/next candidate(s)
[ ]	Scrolls to previous/next candidate(s)
, .	Scrolls to previous/next candidate(s)
Return	Quickly scrolls through all candidates

## Select Keys

New QuanPin and New ShuangPin do not support selection by typing letter keys. If you see candidates prefixed with letter keys (A, B, C, and so on), set `htt` again using the numeric selection keys.

## Separators

In accord with the national Pinyin standard, the separator (‘) is supported to avoid ambiguous interpretations of Pinyin strings. For example, the Pinyin string `[jiang]` can be interpreted as `[jiang]` or `[ji][ang]`; both are valid. In New QuanPin, however, `[jiang]` is interpreted only as `[jiang]`. You must use the separator and enter `[ji'ang]` for it to be interpreted as `[ji]` and `[ang]`. New ShuangPin does not require the use of separators.

## Dictionary Files

New QuanPin and New ShuangPin share two dictionary files: `PyCiku.dat` and `Ud.Ciku.dat`. In the `zh` locale, the default path names are `/usr/openwin/lib/locale/zh/xim/data/PyCiku.dat` and `/usr/openwin/lib/locale/zh/xim/data/UdCiku.dat`. In the `zh.GBK` locale, the default path names are `/usr/openwin/lib/locale/zh.GBK/xim/data/PyCiku.dat` and `/usr/openwin/lib/locale/zh.GBK/xim/data/UdCiku.dat`.

Users cannot normally write to these files. However, since users can affect the way New QuanPin and New ShuangPin work through features such as frequency adjustment and user-defined phrases, it is necessary to update the dictionary files frequently.

A user's dictionary is normally located in `~/.Xlocale/PyCiku.dat` or `~/.Xlocale/UdCiku.dat` (`~` indicates the home directory of the user who starts the `htt` command). When New QuanPin and New ShuangPin are started, they locate and read the dictionary files in the user's home directory. If a dictionary file is not found, the system default path is used (that is, `/usr/openwin/lib/locale/...`).

## New ShuangPin Features

ShuangPin is an abbreviated form of QuanPin. It is faster but more difficult to use than QuanPin. New ShuangPin supports all of the features, keyboard definitions, and dictionary files of New QuanPin.

There are various ShuangPin keyboard mapping designs in PRC. The most popular three are `ZiRanMa`, `Chinese Star`, and `Intelligent_ABC`. The New ShuangPin input

method supports all three of these keyboard mappings. The default keyboard mapping is located in one of the following files:

- /usr/openwin/lib/locale/zh.GBK/xim/IM.prop (for the zh.GBK locale)
- /usr/openwin/lib/locale/zh/xim/IM.prop (for the zh locale)
- ~/.Xlocale/IM.prop

## New ShuangPin Keyboard Mapping

The following tables contain keyboard mappings for the ZiRanMa, Chinese Star, and Intelligent\_ABC keyboards.

TABLE 4-4 ZiRanMa Keyboard Mapping

Key	Definition
i	ch
u	sh
v	zh
A	a
B	ou
C	iao
D	uang, iang
E	e
F	en
G	eng
H	ang
I	i
J	an
K	ao
L	ai

**TABLE 4-4** ZiRanMa Keyboard Mapping *(continued)*

<b>Key</b>	<b>Definition</b>
M	ian
N	in
O	o, uo
P	un
Q	iu
R	uan, er
S	iong, ong
T	ue
U	u
V	v, ui
W	ua, ia
X	ie
Y	uai, ing
Z	ei

**TABLE 4-5** CStar2.97 Keyboard Mapping

<b>Key</b>	<b>Definition</b>
u	ch
i	sh
v	zh
A	a
B	ia, ua

**TABLE 4-5** CStar2.97 Keyboard Mapping *(continued)*

<b>Key</b>	<b>Definition</b>
C	uan
D	ao
E	e
F	an
G	ang
H	iang, uang
I	i
J	ian
K	iao
L	in
M	ie
N	iu
O	o, uo
P	ou
Q	er, ing
R	en
S	ai
T	eng
U	u
V	v, ui
W	ei
X	uai, ue

**TABLE 4-5** CStar2.97 Keyboard Mapping *(continued)*

<b>Key</b>	<b>Definition</b>
Y	iong, ong
Z	un

**TABLE 4-6** Intelligent ABC Keyboard Mapping

<b>Key</b>	<b>Definition</b>
i	ch
u	sh
v	zh
A	a
B	ou
C	in, uai
D	ua, ia
E	e
F	en
G	eng
H	ang
I	i
J	an
K	ao
L	ai
M	ue, ui
N	un

TABLE 4-6 Intelligent ABC Keyboard Mapping (continued)

Key	Definition
O	o, uo
P	uan
Q	ei
R	iu, er
S	ong, iong
T	uang, iang
U	u
V	v
W	ian
X	ie
Y	ing
Z	iao

---

## Quanpy Input Method

Quanpy input method requires up to six keystrokes to type each Chinese Pinyin character. Quanpy maps Pinyin phonetics to single lowercase Roman letters. You can use the Quanpy input method to type individual Chinese characters in both zh and zh.GBK locales.

A lookup area showing the characters that match the Quanpy input is displayed with each keystroke. If more than one option is available, you can type a period (.) to display the next page moving forward through the lookup choices and typing a comma (,) to display the next page moving backward. You can select the character you want by typing the label letter corresponding to the character in the lookup area.

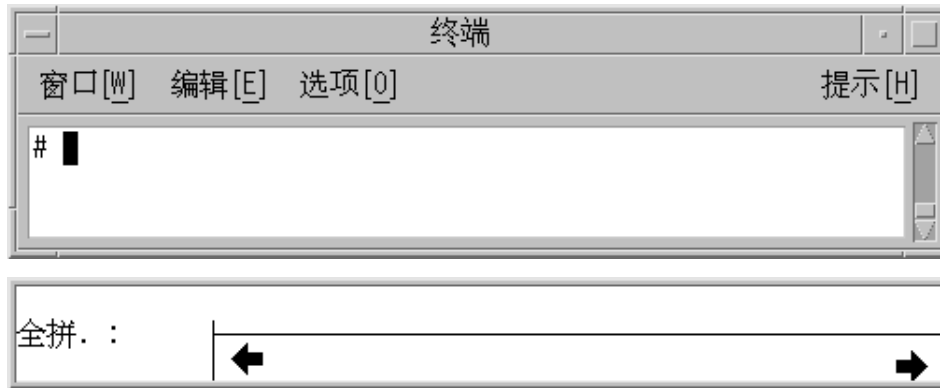
To use the Quanpy input method, make sure that input conversion is on and press F5. The status area, as shown in the following figure, shows that Quanpy mode is on.



---

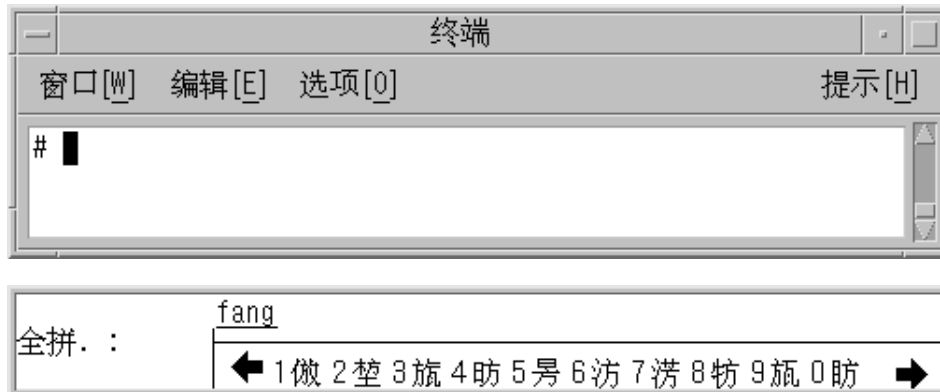
**Note** - This section shows how to use the Quanpy input method in the zh.GBK locale, though its behavior is very similar when run in the zh locale.

---

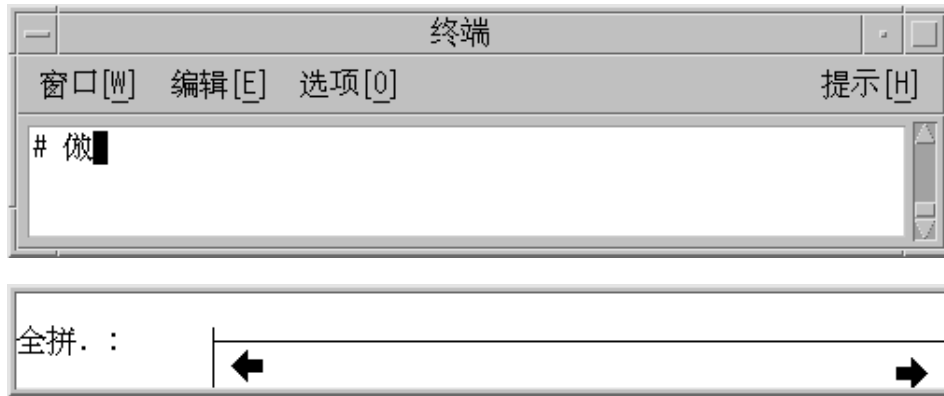


The following figure shows how to use this input method to type the character representing the Full Pinyin word *fang*. The word requires four keystrokes. Type them and select the text as follows:

1. Type the four keystrokes *fang*.



2. Type 1 to select the corresponding GBK Chinese character in the lookup choice listing. Your choice is substituted for the Full Pinyin string in the preedit area.



---

## Stroke Input Mode

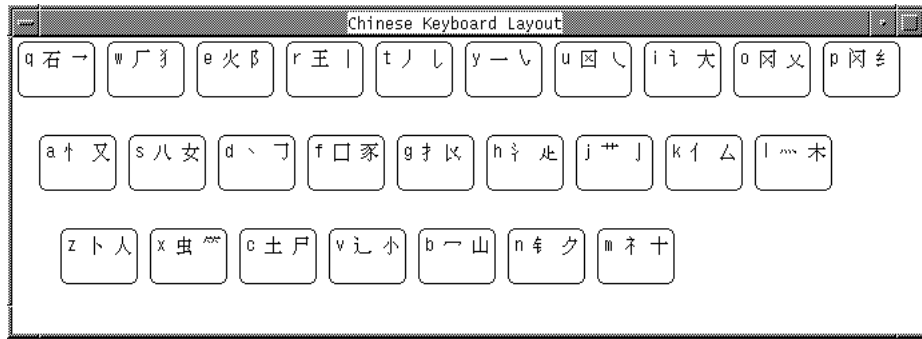
Stroke input conversion mode uses up to four keystrokes to type a Simplified Chinese character. The first two keystrokes stand for strokes in the character.

Each Simplified Chinese character consists of several elements called strokes. These strokes represent the strokes made with calligraphy brush and ink to create the character. Many characters comprise more than ten strokes each. Some combinations of strokes compose major recognizable elements, called radicals, that are common to many characters.

Simplified Chinese characters are commonly organized in a dictionary, grouped according to their strokes/radicals. Each Simplified Chinese dictionary normally contains such an index, called a radical index. However, hundreds of different stroke elements are used by different radical indexes, and the strokes used commonly differ from dictionary to dictionary. These common radical indexes are too large and inconsistent to use for computer character input.

The Stroke input mode however uses a refined and consistent radical index of fifty-two (52) carefully chosen radicals and stroke elements. These 52 are mapped to the twenty-six (26) lowercase alphabetic characters on the computer keyboard as shown in the following figure and tables.

The `clekbd -b` command displays this Stroke keystroke map on the screen:



汉字字母与西文键的对照表

西文键	笔形	西文键	笔形
a	丩, 又	b	一, 山
c	土, 尸	d	丩, 丩
e	火, 阝	f	口, 豕
g	扌, 扌	h	止, 止
i	大, 亼	j	止, 止
k	亼, 亼	l	木, 木
m	十, 十	n	夕, 夕
o	冈, 又	p	冈, 又
q	石, 一	r	王, 一
s	八, 女	t	丿, 丿
u	囟, 丩	v	小, 小
w	厂, 丩	x	虫, 虫
y	一, 丩	z	卜, 人

Some of these 52 stroke elements also sometimes represent the following additional more complex radicals:



of the character. These four keystrokes—head stroke, tail stroke, plus one or two Pinyin strokes or a lookup choice—select a single character.

## Typing Stroke Mode Text

For an example of using Stroke input conversion mode do the following steps:

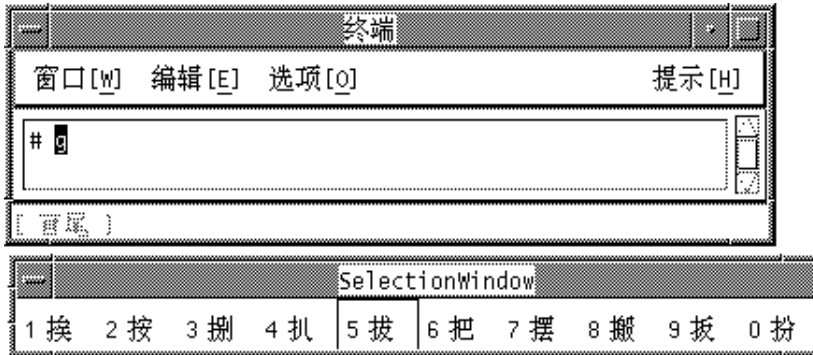
1. Turn on input conversion (press Control-spacebar) and then press Control-B or F4 to turn on Stroke input mode.

The status area shows Stroke mode is on:



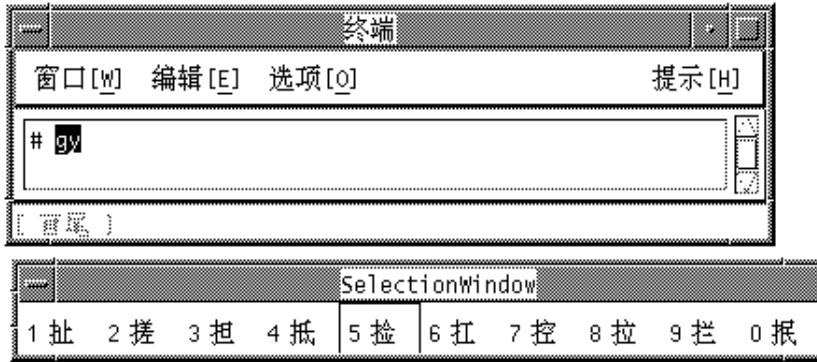
2. Type the lowercase letter *g*.

The lookup choice area displays the first page of many character choices with a head stroke that maps to *g*.



3. Type the lowercase letter *y*.

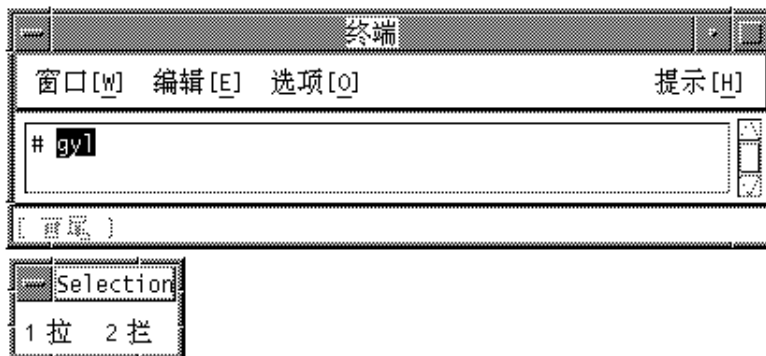
The lookup choice area displays the first page of character choices with a head stroke that maps to *g* and a tail stroke that maps to *y*.



4. Finish the input process using either the following single step a, or by doing steps b and c.
- a. Type 8 to select the corresponding character in the lookup choice area on the screen.  
The character replaces the letters gy in the preedit area:



- b. Alternatively, type a suitable Pinyin input, for example lowercase l.  
The lookup choice area displays the two characters containing those head and tail elements with the just-chosen Pinyin:



- c. Complete the character selection by typing either lowercase a for the second Pinyin or 1 to select the displayed character choice.

Both actions would have the following result, the same as the preceding alternative step a.



## Golden Input Mode

Golden input conversion mode uses its own mapping of Pinyin to single lowercase Roman letter inputs. You can use Golden mode for single character input and also access its phrase dictionary to simplify typing common phrases. Each Simplified Chinese character has only two Golden mode keystrokes for its Pinyin.

Golden mode, like Pinyin mode, maps multiple Pinyin phonetics to the 26 lowercase Roman keyboard letters. However the Golden mode map is different from the Pinyin mode map as can be seen in the following table:

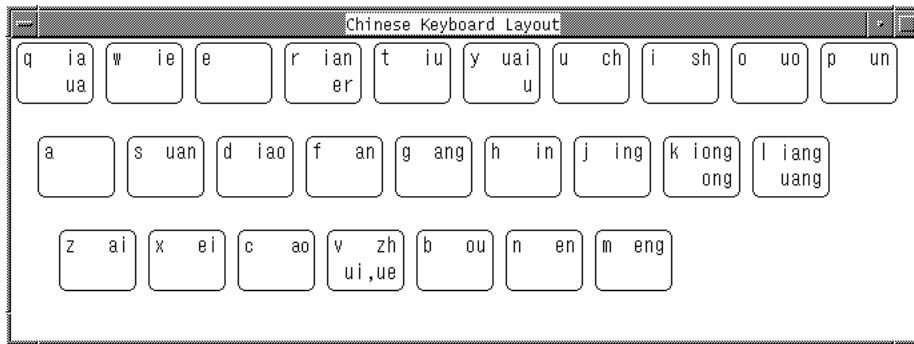
TABLE 4-7 Golden Input for Pinyin Phonetics

Pinyin Phonetics	Golden Keystroke	Pinyin Phonetics	Golden Keystroke
ai	z	ing	j
an	f	iong	k
ang	g	iu	t
ao	c	ong	k
ch	u	ou	b

TABLE 4-7 Golden Input for Pinyin Phonetics (continued)

Pinyin Phonetics	Golden Keystroke	Pinyin Phonetics	Golden Keystroke
ei	x	ua	q
en	n	u, uai	y
eng	m	uan	s
er	e	uang	l
ia	q	ue, ui	v
ian	r	un	p
iang	l	uo	o
iao	d	sh	i
ie	w	zh	v
in	h		

The `clekbd -j` command displays the Golden keystroke map on the screen:



## Typing Single Simplified Chinese Text

In Golden input mode up to three keystrokes type a Simplified Chinese character.

As each of the first two Golden keystrokes (Pinyin phonetics) is typed, a lookup choice area is displayed. Each character displayed in the lookup choice area includes the typed Pinyin(s). This display often consists of several pages of characters, up to



26 per page. Pressing the ] (close square bracket) key pages forward through this display, and pressing [ (open square bracket) pages backward.

A character can be selected in any lookup choice area by typing the uppercase Roman letter corresponding to the character displayed on the screen. This converts the preedit area to the chosen character.

## Example One

For example, to type the Simplified Chinese character for “zhong”:

1. After turning on input mode (press Control-spacebar), turn on Golden mode by typing F6 or Control-J.

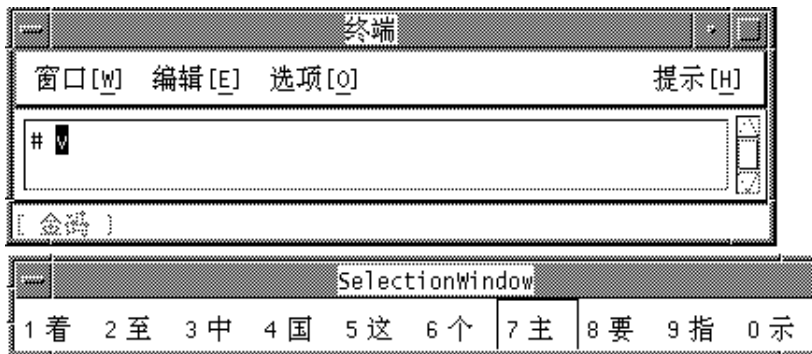
The status area shows Golden mode is on.



The Pinyin “zhong” maps to the two Golden keystrokes v and k, therefore:

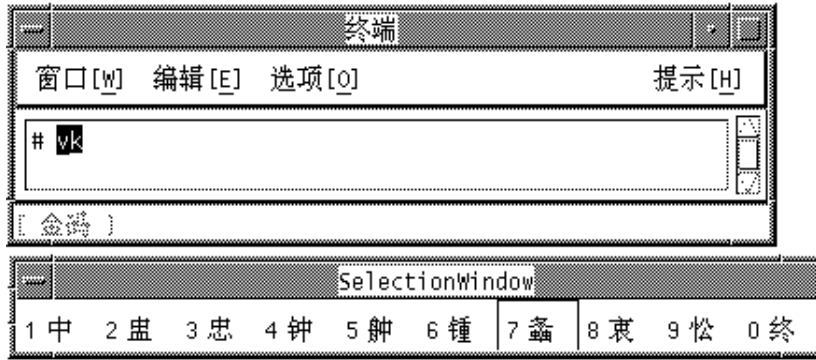
2. Type v.

The following lookup choice area is displayed:



3. Type k.

The following lookup choice area is displayed:



**4. Type 1 to choose the corresponding character from the lookup choice area.**

The preedit area is then converted to this character:

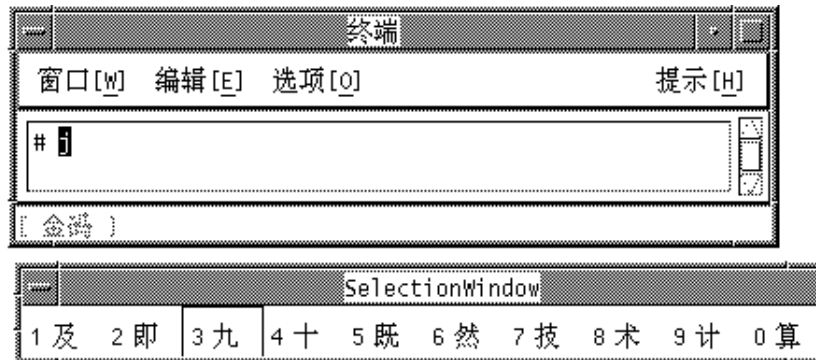


### Example Two

For a second example, type the character “jian,” which maps to the two Golden keystrokes j and r as follows:

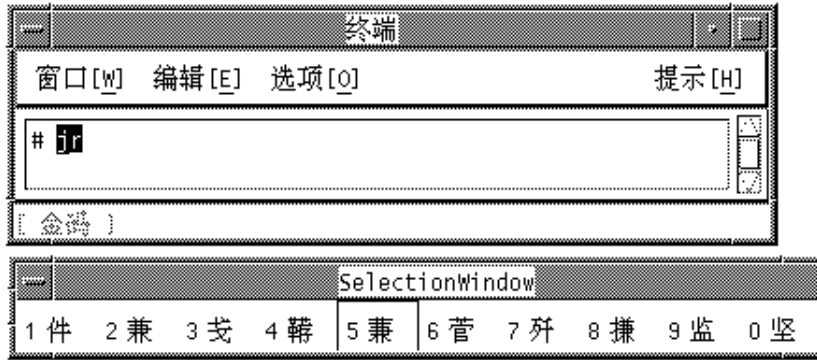
**1. With Golden mode on type j.**

Note the displayed lookup choice area:



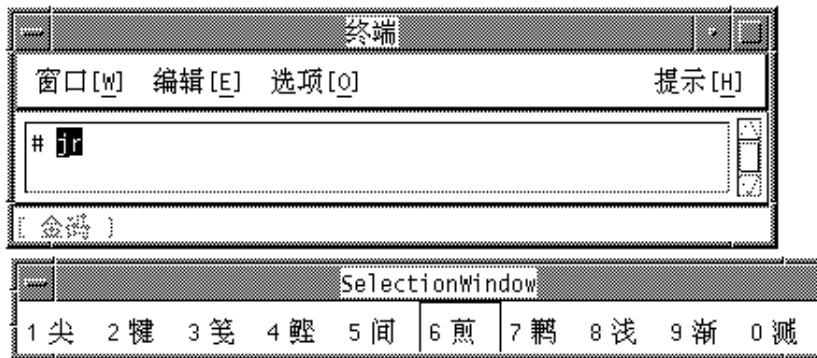
2. With Golden mode on, type r.

Note the displayed lookup choice area:



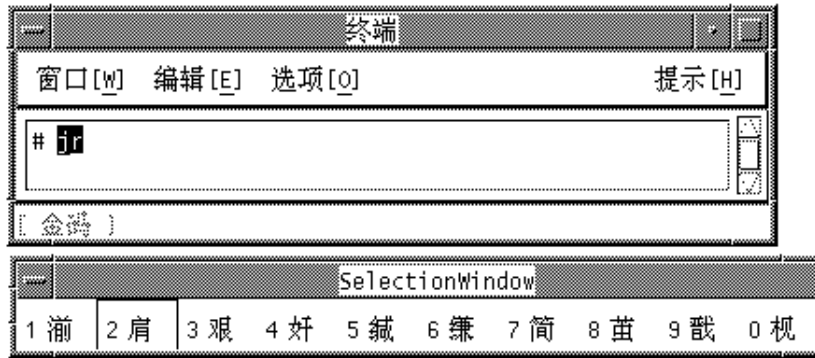
3. The correct character is not visible in the lookup choice area so page forward one page by pressing the ]key.

Note the displayed lookup choice area:



4. The correct character is still not visible in the lookup choice area, so page forward another page by pressing the ]key.

Note the displayed lookup choice area:



5. The correct character is now visible as choice 2, type 2 to select it.



## Typing Phrase Text

Typing four Golden mode keystrokes (lowercase letters) accesses the Golden phrase input dictionary. When you type the fourth Golden keystroke, the lookup choice area might display two or more phrases that contain the four phonetics you typed.

Input ends automatically if the fourth Golden keystroke uniquely defines a phrase in the dictionary. This phrase is then automatically committed. Otherwise you must type the uppercase Roman letter corresponding to the phrase displayed in the lookup choice area. Either event converts the preedit area to the chosen phrase.

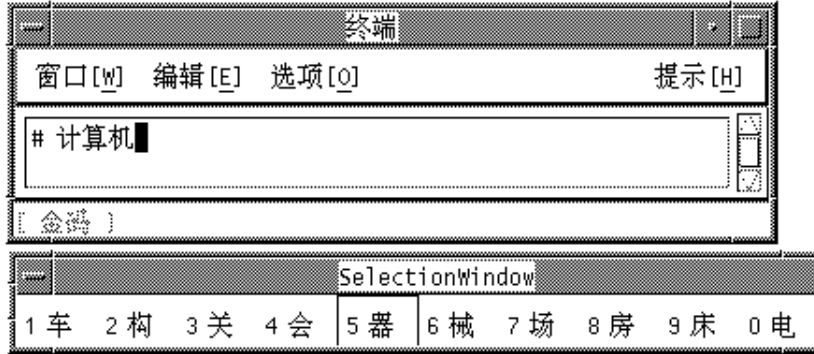
## Typing Two-Character Simplified Chinese Phrases

The Golden rule for phrases of two Simplified Chinese characters is: Type the first two Golden keystrokes of each Simplified Chinese character.

## Typing Three-Character Simplified Chinese Phrases

The Golden rule for phrases of three Simplified Chinese characters is: Type the first two Golden keystrokes of the first Simplified Chinese character and only the first keystroke of the second and third Simplified Chinese characters. For example, to type the common phrase “ji suan ji,” condense its Golden form, “ji ss ji,” further to “jisj” according to the rule.

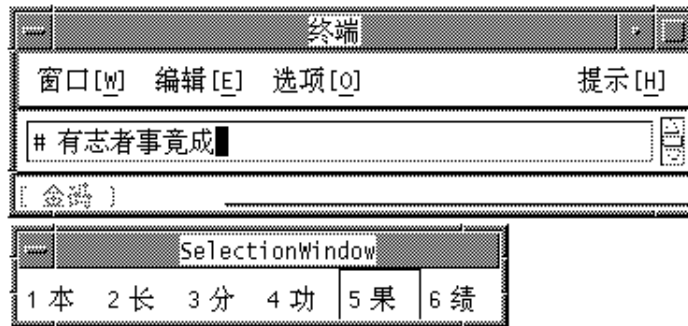
Typing these strokes displays the following:



## Typing Four-or-More-Character Simplified Chinese Phrases

The Golden rule for phrases of four or more Simplified Chinese characters is: Type the first Golden keystroke from each of the first three Simplified Chinese characters plus the first keystroke from the last Simplified Chinese character. For example, to type the common phrase “you zhi zhe shi jing cheng” condense its Golden form “yb vi ve ii jj um” further to “yvvu” according to the rule.

Typing these strokes displays the following:



---

## Intelligent Pinyin Conversion Mode

Intelligent Pinyin mode is based on phrases. It uses the same keyboard mapping as Pinyin mode. However, Intelligent Pinyin mode uses three keystrokes for every character rather than Pinyin mode's one, two, or three keystrokes. And Intelligent Pinyin handles phrases by accepting many keystrokes in its preedit area. It then performs intelligent searches of its phrase dictionary to convert these Pinyin keystrokes to character phrases.

Intelligent Pinyin mode uses the same map as Pinyin mode to map multiple Pinyin phonetics to the 26 lowercase Roman keyboard characters.

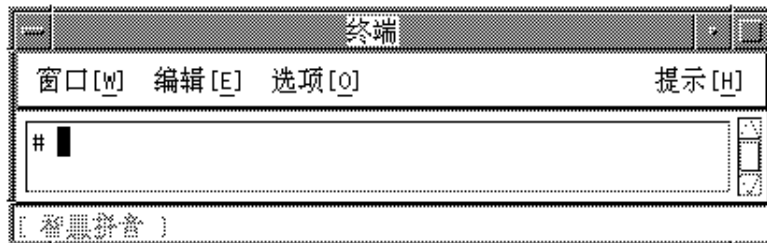
## Typing Intelligent Pinyin Text

See the following example and description of Intelligent Pinyin mode.

### Turn on Intelligent Pinyin Mode

- ◆ **With conversion mode on, press F7 or Control-S.**

The status area shows Intelligent Pinyin mode:



### Keystroke Use for Phrase Input

You can type phrases of many Pinyin keystrokes using three strokes per character. Type each character that requires fewer than three Pinyin-mode keystrokes by using the [ character as padding.

- ◆ **Type the phrase “dian zi gong ye bu zhong guo bei jing shi,” which requires the keystrokes di j zi [ g s [ y e [ b u [ a s [ g u o b e i j y [ u i [ in accord with the Pinyin-mode keystroke map.**

This entire phrase is reverse-video highlighted in the preedit area as you type:



## Converting Pinyin to Phrase Input

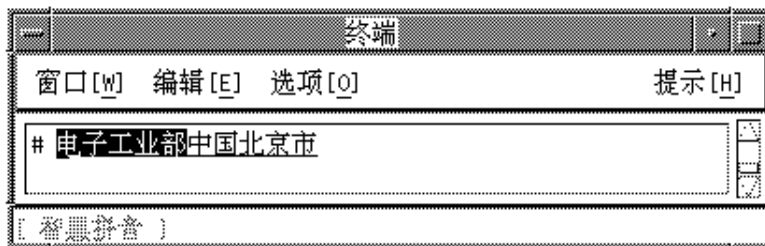
- ◆ To convert Pinyin keystrokes to Simplified Chinese characters, press Control-N.

When you press Control-N, the conversion process matches Simplified Chinese phrases in the dictionary with the Pinyin input in the preedit area as explained in the following sections.

### *Using Focus Areas for Phrase Input*

After Control-N is pressed, the conversion process works from left to right through the preedit area choosing its longest phrases. Given input of more than one phrase, this intelligent behavior segments the preedit area into *focus areas* corresponding to the identified phrases. The leftmost phrase is initially the *current focus area* and reverse-video highlighted. The other phrases are only underlined.

At this point the only part of the preedit area that is reverse-video highlighted is the current focus area, as shown by the example:



### *Controlling the Current Focus Area*

Some Pinyin input can represent more than one Simplified Chinese character phrase, although only one such phrase is displayed in the current focus area. You can look at the other choices as follows:

- Show choices one at a time in the current focus area by pressing Control-N to display the next choice forward through the choices and Control-P to display the preceding choice back through the choices.
- Alternatively have the other phrases displayed in a lookup choice area by pressing Control-W. If more than one page of choices is available then pressing Control-N displays the next page and Control-P to displays the previous page. Press the uppercase letter of a displayed choice to put it into the current focus area.

To lengthen the current focus area by one character press Control-I.

To shorten the current focus area by one character press Control-U.

To move the current focus area forward to the phrase to the right press Control-F.

Pressing Control-B moves the current focus area to the phrase to the left.

- ◆ **Move the current focus area forward to the phrase to the right by pressing Control-F.**

The window would then appear like this:



### *Committing the Phrases in the Predit Area*

Press Control-K when the preedit area contains the appropriate characters (converted from Pinyin input), which commits them. Otherwise commit them by resuming typing lowercase letters for Pinyin input. Either action commits the preedit area's contents.

- ◆ **Press Control-K:**

The example appears as follows:





## Intelligent Pinyin Shortcuts

The following table contains the keyboard shortcuts pertaining to Intelligent Pinyin.

TABLE 4-8 Keyboard Shortcuts for Intelligent Pinyin

Default Key	Function Description
F7	Turn on Intelligent Pinyin mode; the following control sequences apply:
Control-K	Commit and display Intelligent Pinyin conversion choice
Control-N	Display the next Intelligent Pinyin choice
Control-P	Display the previous Intelligent Pinyin choice
Control-N	Display the next group of Intelligent Pinyin choices
Control-P	Display the previous group of Intelligent Pinyin choices
Control-W	Display Intelligent Pinyin lookup choices

## Simplified Chinese Symbol Input Mode

Typing in this mode makes the characters, digits, punctuation, and other symbols on the ASCII/English keyboard two-columns (two-bytes) wide in the input. No intermediate characters appear or are used. Simplified Chinese Symbol input is committed directly to the application just as is normal ASCII. No Simplified Chinese characters can be typed in this mode.

# GBK Code Input Method

This method uses the GBK code defined by the Chinese Internal Code Specification. It includes all of the Chinese characters and symbols in GB2312-80, and other CJK Chinese characters in GB 13000-1. Each Chinese character or symbol is identified by a four hexadecimal digital internal code defined in the Chinese Internal Code Specification.

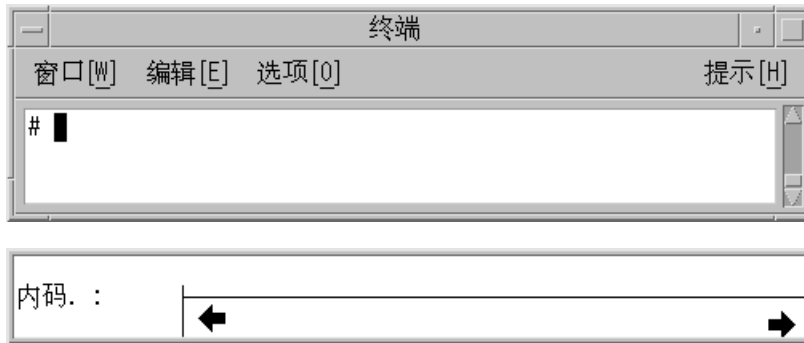
## Typing GBK Code Text

This section contains instructions on how to use the GBK codes to type Chinese characters and symbols.

1. In a new Terminal, turn Chinese input conversion on by pressing Control-spacebar.

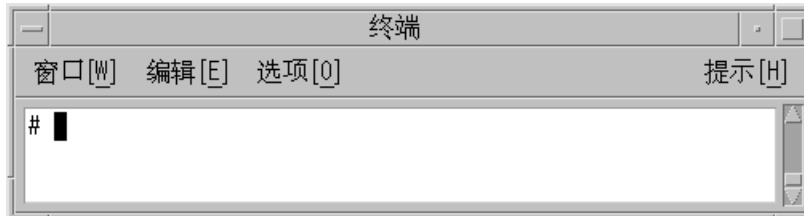
2. Press F2 to turn on the GBK code input method.

The status area shows that GBK code input mode is on.



3. Press the first three of the four keys that represent the character to display (in this example, b0a1).

The key remains visible in the preedit area.





#### 4. Type the fourth key.

The character automatically replaces the preedit area.



---

## Japanese Input Method

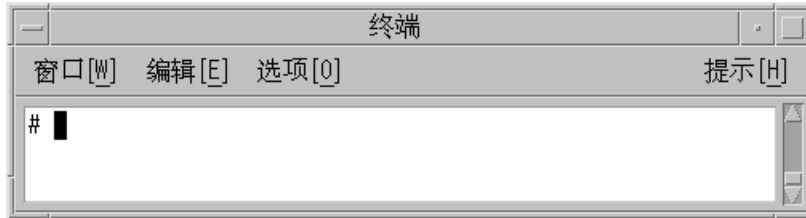
The Kanji input method provides a simple way to insert Japanese Kanji characters included in the GBK character set. If you do not know how to pronounce Japanese, you can press a key to get a lookup choice table and select what you want to insert. If you know Japanese, you can type Japanese Kanji phonetics to insert the Japanese text.

### Typing Kanji Text

Follow the steps in this section to use Kanji input mode.

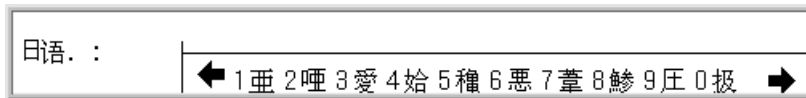
1. **Turn on input conversion (Control-spacebar) and then press F4 to turn on the Kanji input method.**

The status area shows that Kanji mode is on:



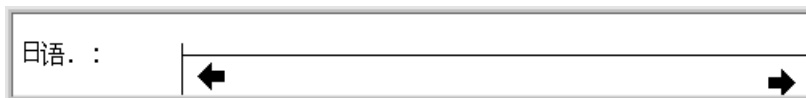
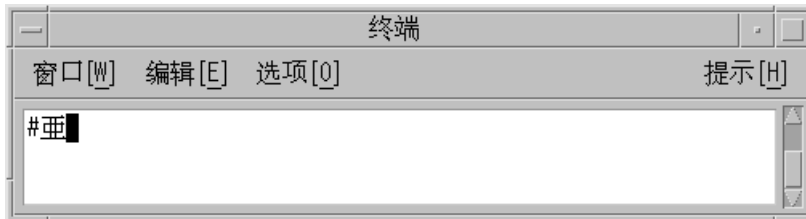
**2. Press Control-L.**

The lookup choice area displays the first page of Japanese kanji choices with this key event.



**3. Type a period (.) to display the next page of lookup choices, or type a comma (,) to display the previous page of lookup choices.**

**4. Press 1 to select the first character in the lookup choice area.**

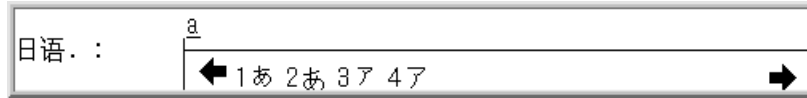


You can also type Japanese kanji phonetics to insert Kanji characters.

5. Type a.



The corresponding characters are displayed in the lookup choice window.



6. Press 1 to select the corresponding character in the lookup choice area.



---

## Hanja Input Method

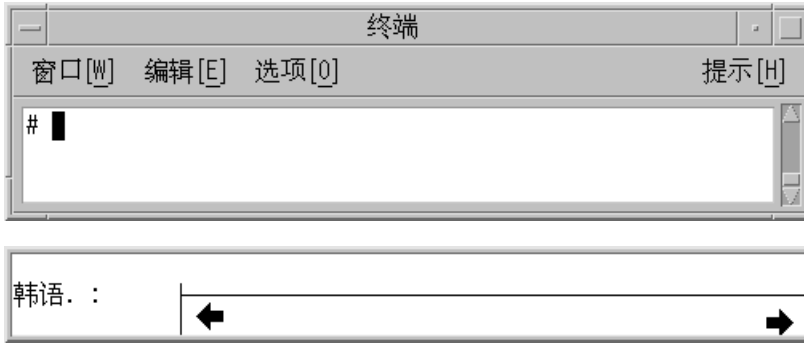
The Hanja input method provides a simple way to insert Korean Hanja characters included in the GBK character set.

### Typing Hanja Text

This section describes how to create Hanja text.

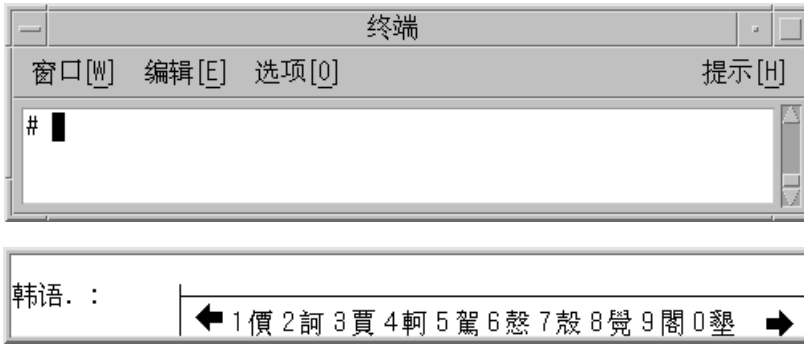
1. Turn on input conversion (Control-spacebar) and then press F6 to turn on Hanja input mode.

The status area shows Hanja mode is on.



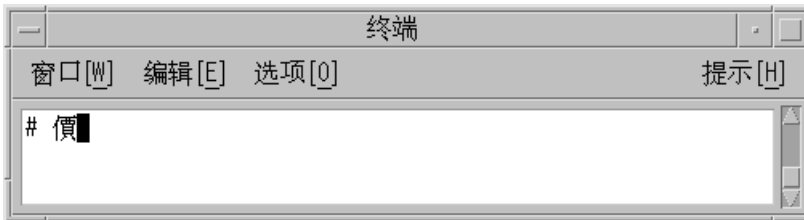
2. Press Control-L.

The lookup choice area displays the first page of all Korean Hanja characters that match this key event.



3. Type a period (.) to display the next page of lookup choices, or type a comma (,) to display the previous page of lookup choices.

4. Press 1 to select the first character in the lookup choice area.



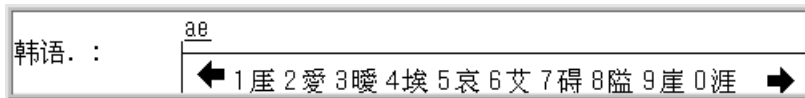


You can also type Korean phonetics to insert Hanja characters.

**5. Type ae to display the corresponding characters.**



The corresponding characters are displayed in the lookup choice window.



**6. Press 6 to select the corresponding character in the lookup choice area.**



---

## Zhuyin Input Method

The Zhuyin input method uses up to four keystrokes to type a Traditional Chinese character, which is in the GBK character set.

This input method can be used to type any Traditional Chinese character in the Zhuyin input method in the BIG5 and EUC\_TW locales; Taiwan Solaris users will find the zh.GBK locale very helpful.

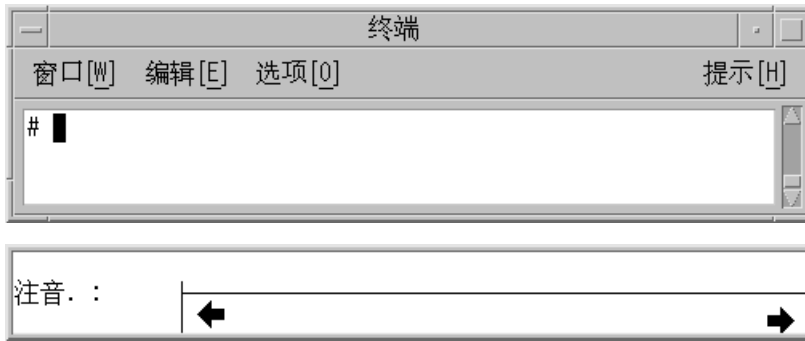
The following table lists the Zhuyin characters.

Zhuyin phonetics	Pinyin phonetics	Zhuyin Mode Keystroke	Zhuyin phonetics	Pinyin phonetics	Zhuyin Mode Keystroke
ㄅ	b	1	ㄩ	a	8
ㄆ	p	q	ㄛ	o	i
ㄇ	m	a	ㄝ	e	k
ㄉ	f	z	ㄟ	ie	,
ㄊ	d	2	ㄞ	ai	9
ㄋ	t	w	ㄠ	ei	o
ㄌ	n	s	ㄡ	ao	l
ㄍ	l	x	ㄣ	ou	.
ㄎ	g	e	ㄤ	an	0
ㄏ	k	d	ㄥ	en	p
ㄉ	h	c	ㄤ	ang	;
ㄊ	j	r	ㄤ	eng	/
ㄍ	q	f	ㄤ	er	-
ㄊ	x	v	ㄤ	i	u
ㄐ	zh	5	ㄤ	u	j
ㄑ	ch	t	ㄤ	ü	m
ㄒ	sh	g	tone 1		space
ㄓ	r	b	tone 2		6
ㄔ	z	y	tone 3		3
ㄕ	c	h	tone 4		4
ㄌ	s	n			

When using this input method, the last keystroke must be the tone of the Chinese character. A lookup window will appear if there are some Zhuyin phonetic candidates. Type a period (.) to display the next page of lookup choices, or type a comma (,) to display the previous page of lookup choices, and then select the character that you want.

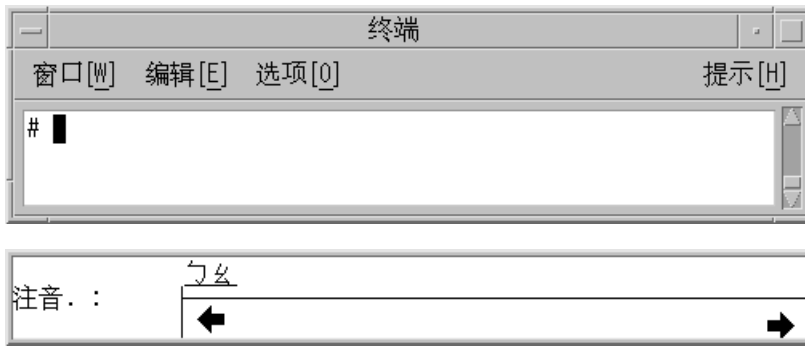


To use the Zhuyin input method, make sure that input conversion is on and press F7 to turn on Zhuyin mode. The status area, as shown in the following figure, shows that the Zhuyin mode is on.



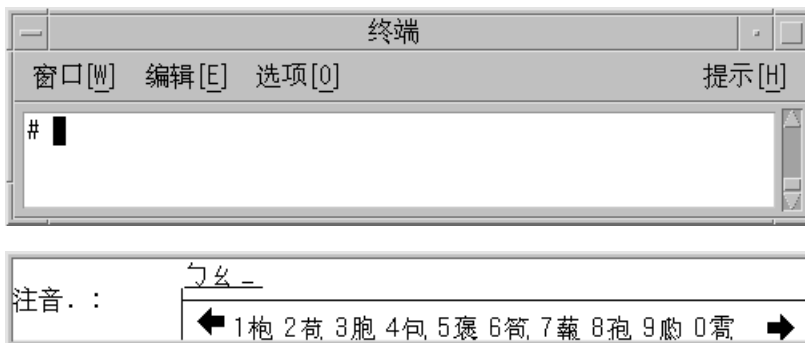
The following is an example of using Zhuyin input method to type a Zhuyin phonetic character.

1. Press the 1 key, followed by the lowercase letter l.

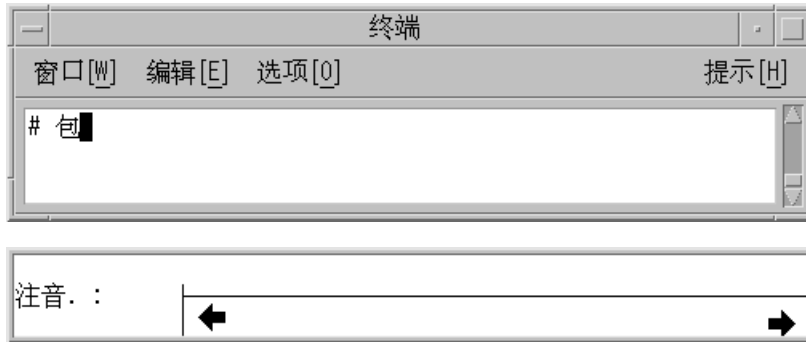


2. Type a space, representing tone 1.

The lookup choice area displays the first page of all Zhuyin characters that match this key event.



3. Press 4 to select the corresponding character in the lookup choice area on the screen.



---

## Unicode Input Method

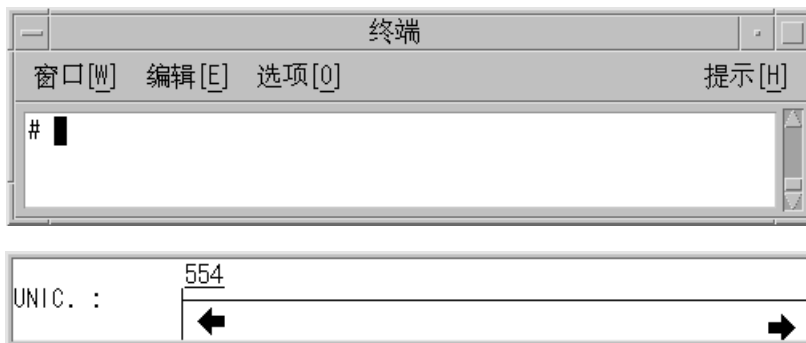
This input method uses Unicode to type Chinese characters and symbols defined by Chinese Internal Code Specification. Using the Unicode Input Method is as easy as gbk code mode if you know the unicodes of the Chinese characters.

### Typing Unicode Text

This section describes how to create Unicode text.

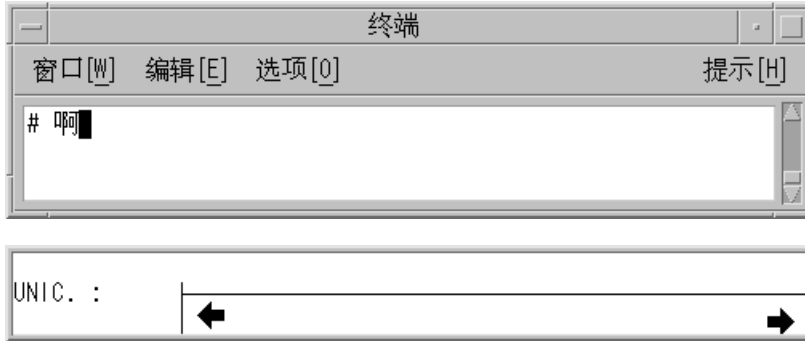
1. Turn on input conversion (Control-spacebar) and then press F10 to turn on the Unicode input method.

The status area shows that the Unicode input method is on.



2. Type the first three of the four keys that represent the character, in this case 554a.

The lookup choice area displays the first page of all Unicode characters that match this key event.

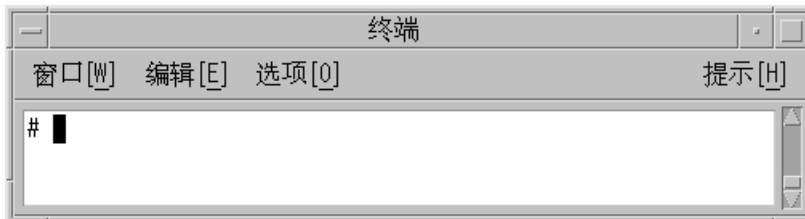


3. Press the fourth key, a.

---

## Input Method Properties Editor

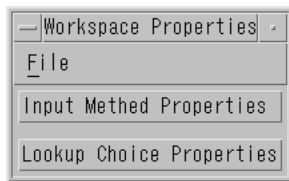
In the zh.GBK locale, you can use the Input Method Properties Editor to modify properties such as the page turn key, candidate select key, and GBK switch key. The F9 key is the switch key for the Input Method Properties Editor. You can open or close the editor by pressing F9 when input conversion is on and an editable area is displayed, as shown in the following illustration. To apply any properties changes, press Control-spacebar. The following procedure describes how to use the Input Method Properties Editor.



## ▼ Changing Input Method Properties

### 1. Press F9 to open the Input Methods Properties Editor.

The editor appears as follows. The pop-up window includes two buttons for modifying the current input method properties.



### 2. Click the Input Method Properties button. The following dialog box appears.



All preedit properties that can be changed are listed in this dialog box. If a property is unused for an input method, then the button is off.

### 3. Click Apply or OK to save the settings.

The properties settings are saved in the file `~/.Xlocale/IM.prop`. The default settings come from the file `/usr/openwin/lib/locale/zh.GBK/xim/IM.prop`.

## ▼ Changing Lookup Choice Properties

The lookup choice properties are used by all of the input methods. You can change the properties as follows:

1. Click the **Lookup Choice Properties** button in the **Workspace Properties** menu. The following dialog box appears.



2. Change the lookup choice properties. If a property is unused for an input method, then the button is off.
3. Click **Apply** or **OK** to save the settings.



# Code Table Input Method Interface

---

---

## Introduction

The Simplified Chinese Solaris software provides code table interface for adding new input methods. In a plain text file, you can create a list Chinese words or phrases along with the key strokes you have selected to create them.

### ▼ Creating a Code Table

#### 1. Create and edit code table source file:

Prepare the code table source file to present the new input method. You need to list characters, words, or phrases and the corresponding key strokes required to create them. Each row defines the mapping between one Chinese word and its input key stroke. The file format is:

```
#  
# Comment  
#  
<code_table>  
[ word   keystroke-sequence]
```

See the following example:

```

#
# 五笔字形码表
#
<codetable>
啊      kb
啊      kbs
啊      kbsk
阿      bs
阿      bsk
阿      bskg
埃      fct
埃      fctd
挨      rct
挨      rctd
哎      kaq
哎      kaqy
唉      ket
唉      kcdt
哀      yeu
皃      rmnn
经济    xciy
五笔字形 gtpg
...    ...

```

## 2. Convert code table:

The source code table needs to be converted to binary form for Chinese Solaris language engine. This can be done by the code table converter: `mkcodetab`. The converter can take several source code table files and converter into a binary code table file. If `-o` is not used, the output will be send to system standard out.

The command syntax is:

```
# mkcodetab source_code_table_files [-o binary_code_table_file]
```

See the following two examples:

```
# mkcodetab wubi.txt -o wubi.ct
# mkcodetab < wubi.txt > wubi.ct
```

Both forms do the same function: convert the source code table `wubi.txt` into the binary code table `wubi.ct`. The following command converts multisource code table files into one binary code table file:



```
# mkcodetab wubi1.txt wubi2.txt wubi3.txt > wubi.ct
```

### 3. Create a dictionary file (optional).

Specifying a dictionary file allows you to add a linking or associating function to an input method. Thus, when the user types a Chinese character, all the words associated with it in the dictionary will be shown for selecting.

The dictionary file is a plain text file that contains the words user wants to have associated. The words are separated by space. For example:

```
word1 word2 word3  
word4 word5  
word6 word7 word8 word9  
... ..
```

### 4. Add the code table into the input method specification file.

After code table is ready, the input method can be added to the system by updating input method specification file:

```
/usr/lib/mle/zh/data/keybind.dat
```

The `keybind.dat` file requires the following information:

- Input method name
- Function key and control key to turn on the input method
- Code table (binary) file name
- Key to show previous selection screen
- Key to show the next selection screen
- Wild char definition
- Key to commit the first candidate in the selection

Once `keybind.dat` is updated, the Simplified Chinese Solaris operating environment can interpret the input method. This information is added to `keybind.dat` in the following format:

```
% input_method_name: control_key:function_key:display_message
key_word = definition
```

See the following example:

```
%wubi:          ^W:F10:[ 五笔字型 ]
                CT = /usr/lib/mle/zh/data/wubi.ct
                PG = ,
                NG = .
                FC = ' '
                WC = 'z'
```

TABLE 5-1 Elements in keybind.dat File

<i>key_word</i>	<i>Definition</i>
CT:	Code Table file name
PG:	Key to display previous group of selections
NG:	Key to display next group of selections
FC:	Key to commit the first (matched) character
WC:	Wild Character to match any key in input keystroke. The matched selections will be shown.
DS:	Key to display all selections (not needed if WC is defined).
DICT:	Dictionary file name
PROPS:	If AUTO_POPUP is defined, the selections will be automatically shown while any keystroke is typed. The DS definition is not needed). One example to use this is Pinyin input method.

Edit the system keybind file, `/usr/lib/mle/zh/data/keybind.dat`, if you want to make the new input method available for all users on this system. If you

just want to do so for your own environment, you can set the KEYBIND environmental variable:

```
% setenv KEYBIND your_keybind_file
```

**5. Add a new line in /usr/openwin/lib/locale/zh/xim/hotkey.dat, as shown in the following example:**

```
**** hotkey.dat ****

@BEGIN_USER
#Hot Key      engine_name      method_table_name
#F8 for NewPY, the default IM
F8      cm      cm
F11     shuangpy      cm
@END_USER
@BEGIN_SYSTEM

##### All of the following hotkeys should be in the others.so.

#Hot Key      engine_name      method_table_name
#Hot Key for "location" method.
F2      others      cm
#Hot Key for "py" method.
F3      others      cm
#Hot Key for "Stroke" method.
F4      others      cm
#Hot Key for "full py" method.
F5      others      cm
#Hot Key for "Golden" method.
F6      others      cm
#Add a new IM by user, Hot key for "Wubi" method.
F10     others      cm
#Hot Key for "Chinese Symbol" method.
F12     others      cm
@END_SYSTEM
```

**6. Restart the input method server (htt) so that your new input method can be recognized.**

For the above example, the F10 function key will turn on the new input method. Your new input method is ready to use.



# Localized Applications

---

---

## About This Chapter

This chapter describes selected properties you need to use with two localized applications of the Simplified 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 Simplified 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 to send (8-bit) Simplified 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 Simplified 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 Simplified Chinese characters.

---

---

## Using `talk` With Simplified Chinese Characters

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

---

## The `xtobdf` Utility

Simplified Chinese Solaris software provides this BDF font generator:

`xtobdf` — Convert from font in X server to font in BDF

---

## Convtool

Convtool is a graphic user interface utility that enables file conversion between various code sets. Its functionality is the same as `iconv`.



- ◆ To get a list of the conversion options that are available on the system, select “Available Conversions” from the File pull-down menu.

After you select one conversion from the list, the fields for the source and destination code set will be filled automatically. The conversion options are the same as those available through `iconv`.

Convtool can be launched through the App Manager interface.

---

## `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. Simplified 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 Simplified Chinese Solaris software.

**TABLE 6-1** Simplified Chinese `iconv` Code Conversion Modules (`zh` locale)

<b>Code</b>	<b>Symbol</b>	<b>Target Code</b>	<b>Symbol</b>
GB2312-80	zh_CN.euc	ISO 2022-7	zh_CN.iso2022-7
ISO 2022-7	zh_CN.iso2022-7	GB2312-80	zh_CN.euc
GB2312-80	zh_CN.euc	ISO 2022-CN	zh_CN.iso2022-CN
ISO-2022-CN	zh_CN.iso2022-CN	GB2312-80	zh_CN.euc
UTF-8	UTF-8	GB2312-80	zh_CN.euc
GB2312-80	zh_CN.euc	UTF-8	UTF-8
GB2312-80	zh_CN.euc	BIG5	zh_TW-big5
BIG5	zh_TW.big5	GB2312-80	zh_CN.euc
HZ-GB-2312	HZ-GB-2312	GB2312-80	zh_CN.euc
GB2312-80	zh_CN.euc	HZ-GB-2312	zh_CN.euc

**TABLE 6-2** Simplified Chinese `iconv` Code Conversion Modules (`zh.GBK` locale)

<b>Code</b>	<b>Symbol</b>	<b>Target Code</b>	<b>Symbol</b>
UTF-8	UTF-8	GBK	zh_CN.gbk
GBK	zh_CN.gbk	UTF-8	UTF-8
GBK	zh_CN.gbk	ISO-2022-CN	zh_CN.iso2022-CN
ISO-2022-CN	zh_CN.iso2022-CN	GBK	zh_CN.gbk
GBK	zh_CN.gbk	BIG5	zh_TW-big5
BIG5	zh_TW-big5	GBK	zh_CN.gbk



**TABLE 6-2** Simplified Chinese `iconv` Code Conversion Modules (`zh.GBK` locale) (continued)

Code	Symbol	Target Code	Symbol
HZ-GB-2312	HZ-GB-2312	GBK	zh_CN.gbk
GBK	zh_CN.gbk	HZ-GB-2312	zh_CN.gbk
HZ-GB-2312	HZ-GB-2312	UTF-8	UTF-8
UTF-8	UTF-8	HZ-GB-2312	HZ-GB-2312

The following `iconv` code conversion modules are located in `/usr/lib/iconv`:

For the `zh` locale:

- `zh_CN.euc%zh_TW-big5.so`
- `zh_TW-big5%zh_CN.euc.so`

For the `zh.GBK` locale:

- `UTF-8%zh_CN.gbk.so`
- `zh_CN.gbk%UTF-8.so`
- `zh_CN.gbk%zh_CN.iso2022-CN.so`
- `zh_CN.iso2022-CN%zh_CN.gbk.so`
- `zh_CN.gbk%zh_TW-big5.so`
- `zh_TW-big5%zh_CN.gbk.so`

In the following example, an EUC mail file is converted to ISO 2022-CN:

```
system% iconv -f zh_CN.euc -t zh_CN.iso2022-CN mail.euc > mail.iso2022-CN
```

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



# Fonts

---

---

## About This Chapter

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

---

## Display PostScript System (DPS)

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

## Using Simplified Chinese PostScript Fonts and DPS Facilities

The Simplified Chinese Solaris operating environment includes the fonts listed in the following table. You can use any of the Simplified Chinese fonts just as you would use Roman fonts.

**TABLE 7-1** Simplified Chinese Solaris Operating Environment PostScript Fonts

<b>Font Name</b>	<b>Description</b>
Song-Medium	Alias of Song-Medium-EUC; can be used like a Roman font.
Song-Medium-EUC	Song-Medium font, EUC encoding, horizontal display; can be used like a Roman font.
Kai-Medium	Alias of Kai-Medium-EUC; can be used like a Roman font.
Kai-Medium-EUC	Kai-Medium font, EUC encoding, horizontal display; can be used like a Roman font.
FangSong-Medium	Alias of FangSong-Medium-EUC; can be used like a Roman font.
FangSong-Medium-EUC	FangSong-Medium font, EUC encoding, horizontal display; can be used like a Roman font.
Hei-Medium	Alias of Hei-Medium-EUC; can be used like a Roman font.
Hei-Medium-EUC	Hei-Medium font, EUC encoding, horizontal display; can be used like a Roman font.

The following figure shows a sample of Song-Medium.



Figure 7-1 Sample Simplified Chinese Text Display PostScript Output

## Creating Composite Roman and Simplified Chinese Fonts

You can create composite fonts using any one Roman font and the Simplified Chinese fonts. For example, the following PostScript code defines a composite font, Times-Italic+Kai-Medium, which uses Times-Italic for ASCII characters and Kai-Medium font for Simplified Chinese characters:

```

/makeEUCfont {
  /AsianFont exch def
  /WestFont exch def
  /NewFont exch def

```

```

/CIDInit /ProcSet findresource
begin
  NewFont [ AsianFont WestFont ]
  beginrearrangedfont
    1 usefont
    1 beginbfrange
      <00> <7e> <00>
    endbfrange
  endrearrangedfont
end
} bind def

/LC_Times-Roman /Times-Roman /Kai-Medium makeEUCfont

```

## Using Simplified Chinese Fonts in DPS Programming

You can use Simplified Chinese fonts just as you use Roman fonts in DPS wrap definitions. The following code sample creates the display in the next figure.

```

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
  /Song-Medium cpSize selectfont
  (text) stringwidth pop 2 div neg pointSize neg moveto
  (text) show
endps

```

You can call `PSWDisplayText(Chinese text)` in a C program to display the designated Chinese text. For an example see the following figure.



*Figure 7-2* Using Simplified Chinese Solaris Operating Environment DPS  
Simplified Chinese Solaris software provides TrueType support in DPS.

---

## Editing Fonts With `fontedit`

This section shows you how to edit or create and install new characters in Portable Compiled Format (PCF) format fonts used by the Simplified Chinese Solaris operating environment. You can then 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.

### 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 Simplified Chinese Solaris operating environment provides the Font Editor for editing bitmap fonts.

### ◆ Start the Font Editor.

A Font Editor window is displayed.

```
``system% fontedit ````
```

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

## Preparing a Font File

### ◆ Locate the Simplified Chinese font file you want to edit in BDF format.

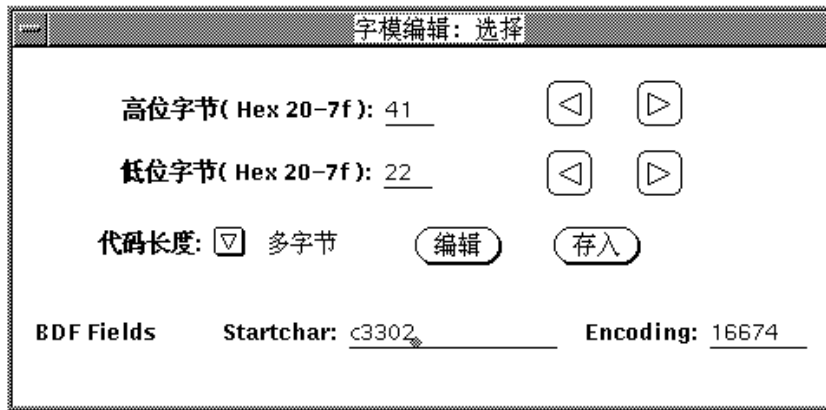
Its encoding should start at 8481 (0x2121). For example, say you have a BDF file such as myfont14.bdf:



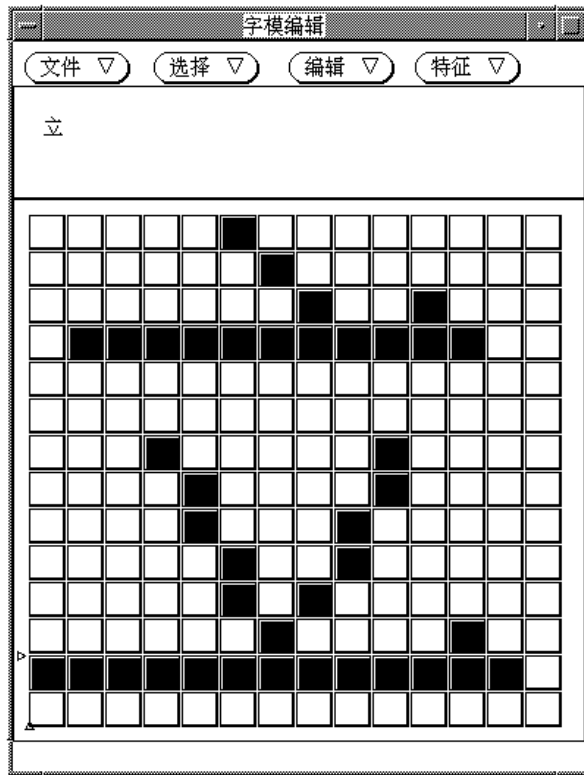
```
``STARTFONT 2.1``COMMENT Sample Font``FONT Myfont-  
Medium14``SIZE 14 75 75``...``STARTCHAR c101``ENCODING 8481``...``
```

## Editing a BDF Font File

1. Hold down the MENU mouse button on the Font Editor window's File button and select Load... on the pull down menu.
2. Type the name of the BDF file you want to edit, for example myfont14.bdf, in the Font Name field of the Load pop-up window.
3. Click SELECT on the Load button in the pop-up.  
This loads the file.
4. Click SELECT on the Font Editor main window Select button.  
This displays the Select window, where you do the following steps.
5. Type the high and low bytes of the font encoding code for the character you want to edit, for example 0x3302.  
You can use the Next (right arrow) button or Previous (left arrow) button to increase or decrease the high or low byte.



6. Set Code Length to MultiByte for this Simplified Chinese font.  
You would set Code Length to SingleByte for an ASCII/English font.
7. Click on the Edit button (in the Fontedit: Select window) to display the glyph on the main canvas.  
The specified glyph appears in the Font Editor window.



**8. Edit the glyph on the main canvas by turning pixels on/off:**

- 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.

---

**9. When you finish editing the glyph, click SELECT on the Store button.**

**10. Repeat the preceding steps Step 5 on page 97 through Step 9 on page 98 to edit other glyphs in this font.**

**11. After you finish editing, hold down MENU on the File button and select Save... to save the edited glyph(s) in the font file.**

---

## Converting BDF to PCF Format

Before applications can use the modified BDF file, it must be converted to a PCF format file.

- ◆ **Make the edited font usable by the Simplified Chinese Solaris operating environment by converting to X11 PCF format as follows:**

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

For more information, see the `bdf2pcf(1)` man page.

---

## 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 to your font directory:**

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

2. **Run the following commands in your font directory.**

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

```
system% cat >> fonts.alias '-new-myfont-medium-r-normal--16-140-75-75-c-140-gb2312.1980-0 'Myfont-Medium14'^D' system% mkfontdir system% xset +fp `pwd`'
```

3. **You can view your font using the X Font Displayer (`xfd`) by typing:**

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

All Simplified Chinese fonts have XLFD font names, and you can use the `xlsfonts` utility to display these names as follows:

```
system% xlsfonts | grep gb2312
```

---

## Creating Characters With `sdtudctool`

You can create new characters using the drawing tools in the `sdtudctool` utility. (For information on modifying a font, see “Editing Fonts With `fontedit`” on page 95.) The new characters you create with `sdtudctool` can be saved in the five empty code positions between level-1 and level-2 Chinese characters (from 0xD7FA to 0xD7FE). The `sdtudctool` utility supports bitmap, Type 1, and CID fonts. You can also specify a font size for the new characters. This section contains procedures for using `sdtudctool`.

---

**Note** - Currently `sdtudctool` supports only EUC locales, such as `zh`. This utility is therefore not available in the `zh_GBK` locale.

---

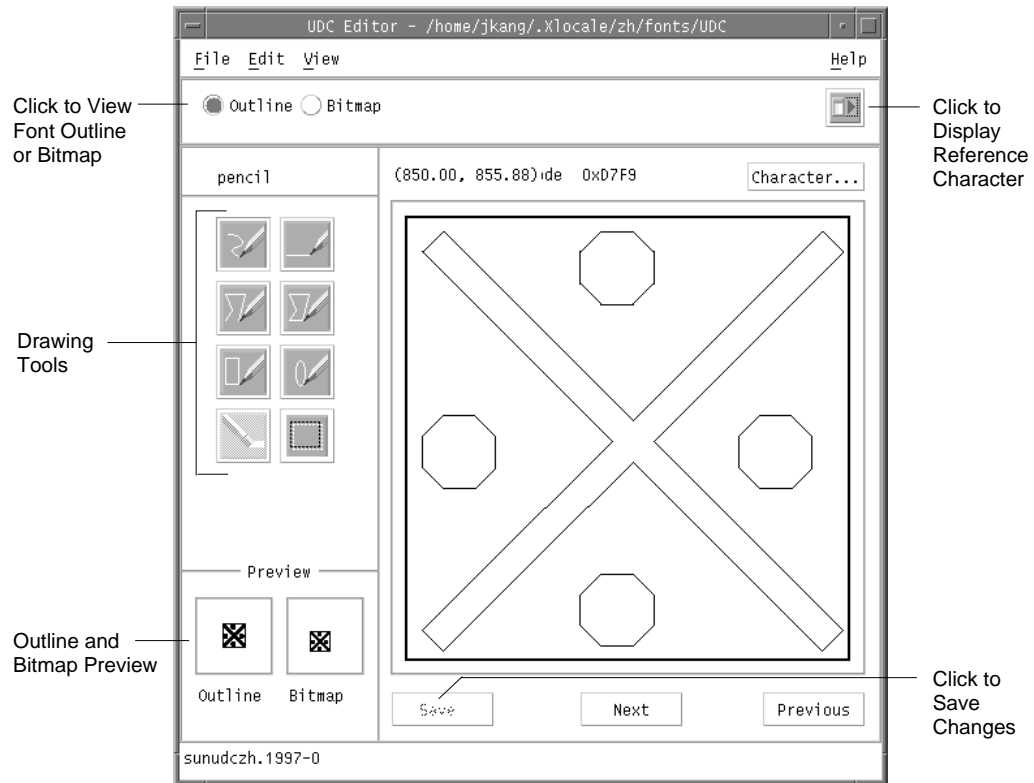
### ▼ Using `sdtudctool`

This section contains instructions for creating new characters with the `sdtudctool` utility.

1. **To start the utility, type the following command:**

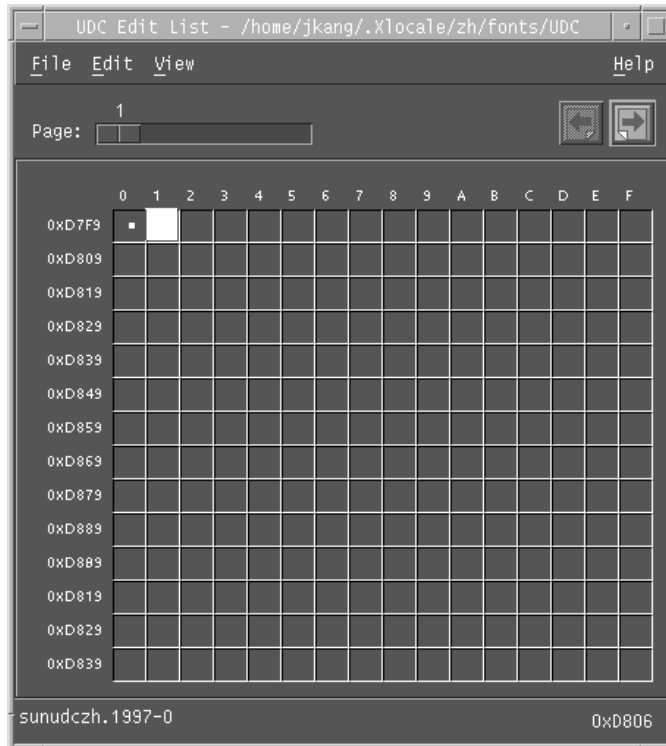
```
system% sdtudctool
```

The following window appears.



**2. Click Character.**

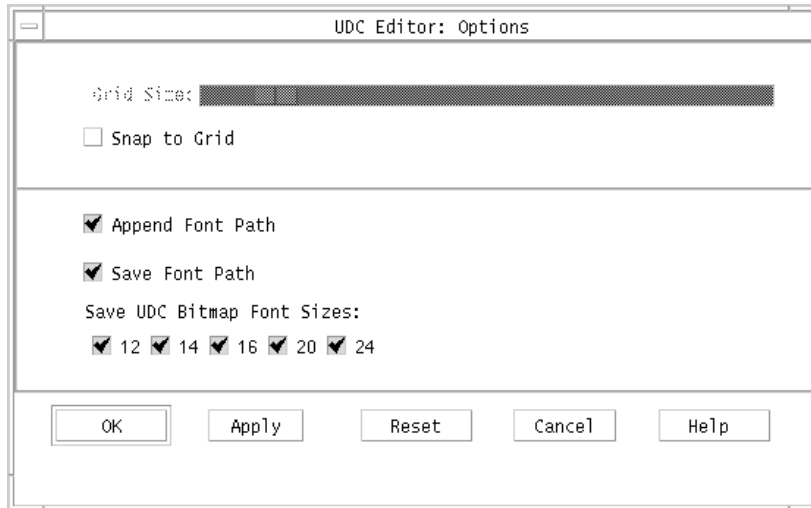
The following dialog box appears:



3. **Specify the code position of the new character.**
4. **Create the outline or bitmap for the character using the drawing tools.**
5. **Click Save to save the character.**
6. **Continue creating new characters.**
7. **When you are finished, choose Save from the File menu to save the changes to your environment.**  
The new character(s) are saved in `$HOME/.Xlocale/zh/fonts/UDC`.

## ▼ Specifying Options

1. **Choose Options from the File menu.**  
The Options dialog box appears.



2. **Turn on Append Font Path to append the UDC font aliases to your font path.**
3. **Turn on Save Font Path to add the font files to the system font path.**

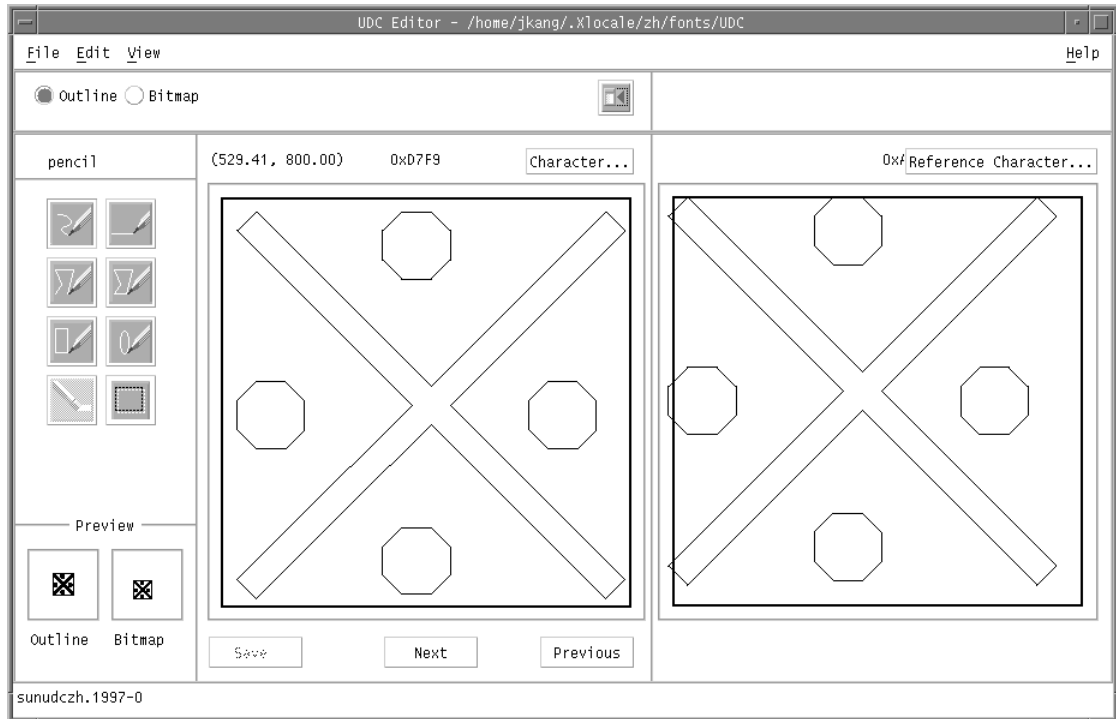
The new font files (located in `$HOME/.Xlocale/zh/fonts/UDC`) will be listed in the `$HOME/.OWfontpath` file and made available to the X server the next time you log in.

If Save Font Path is off, the font files will still be saved in `$HOME/.Xlocale/zh/fonts/UDC`, but will not be added to the system fontpath the next time you log in.
4. **Choose one or more of the following font sizes (in points) for the bitmap fonts: 12, 14, 16, 20, 24.**

## ▼ Viewing the Character Reference

1. **To view the character reference, click the icon below the Help menu.**

The following window appears.



2. To close the window, click the icon again.

## ▼ Typing the New Characters

1. To type the characters you created, use the Location code input mode (for more information, see “Location Code Mode ” on page 33).



# Simplified Chinese Printing Facilities

---

---

## Introduction

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

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

---

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

---

You can use the Simplified Chinese Solaris `xetops` utility to print files containing Chinese text on a PostScript printer. These printing facilities can be used directly from a command line or from within Simplified Chinese Solaris applications as discussed in the following sections.

---

# Printing Chinese Output From a Command Line

## Printing With a Line Printer

The Simplified 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 command in the following example:**

```
system% lp EUC_filename
```

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

```
system% lp -T GB GB_filename
```

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

## Printing With the `xetops` Utility

The `xetops` utility converts Chinese text into a bitmapped graphics printed image. A typical command line for printing a file named `filename` containing Simplified Chinese characters, with or without ASCII/English characters, would be as follows:

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

Make `filename` the name of the file to print. This file may contain ASCII/English characters as well as Chinese. Refer to the `xetops(1)` man page for more detailed information.

The `xetops` utility allows you to print Simplified Chinese characters using a PostScript-based printer.

In the `zh.GBK` locale, the `xetops` utility allows you to print Simplified Chinese characters using TrueType or Bitmap fonts. For TrueType fonts, you can select four

typefaces, including FangSongTi, SongTi, HeiTi, and KaiTi. For usage information on `xetops`, type

```
system% xetops -h
```

The following syntax information is returned:

```
xetops [-r] [-B|[-T [-t S|F|K|H] [-s [i|m|c]
        [-p A3|A4|A5|B5|USLtr|USLg1]
        [-o length=i|c| [,width=i|c]]
        [filename]
```

The options for `xetops` are as following:

- `-r`: prints the file in landscape mode.
- `-B`: prints the file using bitmap fonts. (The default is TrueType SongTi font.)
- `-T`: prints the file using TrueType fonts.
- `-t`: use this option to specify up to four typefaces. *S* indicates SongTi, *F* indicates FangSongTi, *K* indicates KaiTi, and *H* indicates HeiTi.
- `-s`: designates the size of each character. *i* stands for inches, *m* stands for millimeters, and *c* stands for centimeters. (The default is inches.)
- `-p`: specifies the paper size (A3, A4, A5, B5, US Letter, or US Legal)..
- `-o`: specifies the length and width of the paper. *i* stands for inches, and *c* stands for centimeters. (The default is the number of the lines.)

Here are some examples of `pr` and `xetops`. The following command:

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

will print the file using the SongTi TrueType font. The printed characters will be 4mm in size.

The following command:

```
system% pr filename | xetops -t K -s 8m | lp
```

will print the file using the KaiTi TrueType font. The printed characters will be 8mm in size.

The following command:

```
system% xetops -t H -s 1.2 filename | lp
```

will print the file using the HeiTi TrueType font. The printed characters will be 1.2 inches in size.

The following command:

```
system% xetops -B filename | lp
```

will print the file using a bitmap font.

## Printing with the mp Utility

A new and enhanced `mp(1)` print filter is available in the Solaris 8 environment that can print various input file formats including flat text files written in UTF-8. It uses TrueType and Type 1 scalable fonts and X11 bitmap fonts available on the Solaris system.

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.

---

To print with the `mp` utility, type the following:

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

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.



# Open Windows Information

---

This appendix includes information specific to the OpenWindows environment.

---

## Introduction to OpenWindows

This section describes the Simplified Chinese localization.

### Design of Simplified Chinese Solaris Software

The Simplified 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 (`o1wm`) that communicate with users in Simplified 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. Programmer/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 for applications to start in. 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 Simplified Chinese OpenWindows environment.

These settings take effect each time a local application is started. Application windows are displayed in the currently set locale. After you change the locale, new application windows are displayed in the new locale, but existing application windows stay in the locale in which they were originally started.

## Simplified Chinese OPEN LOOK DeskSet

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

- Audio Tool – Tool for recording, playing, editing, and controlling workstation audio parameters.
- Binder – Tool for defining which actions associate with which file type. This association can be set graphically.
- 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.
- Command Tool – Standard OPEN LOOK scrolling window terminal emulator.
- 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.
- Font Editor – Visual tool for editing font appearance and creating new 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 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.
- 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

This section describes the OpenWindows files required for using the Simplified Chinese operating environment.

## Checking Your User Environment for the First Time

Before you log in, your system administrator should have 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 Simplified Chinese features.

### `.cshrc` File

These system environment variables might not have been set. So the first time you log in, before you start OpenWindows for the first time, check to make sure lines like the following are in the `.cshrc` file in your home directory:

```
setenv LANG zh
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` (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 *Simplified Chinese Solaris System Administrator's Guide* for further information on setting up your Simplified Chinese Solaris system.

## .openwin-init File

If you have a `.openwin-init` file in your home directory and might use the Simplified Chinese character input facilities, check this file for the presence of an `htt` command, as described in “How and When `htt` Is Started” on page 27.

## .xinitrc File

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

## Starting the Simplified Chinese OpenWindows Environment

After you verify that your `.cshrc` file has set your user environment correctly for Simplified Chinese language operation, you are ready to start the Chinese OpenWindows environment as follows:

- ◆ **Type the following command at the system prompt to start the Simplified Chinese OpenWindows environment:**

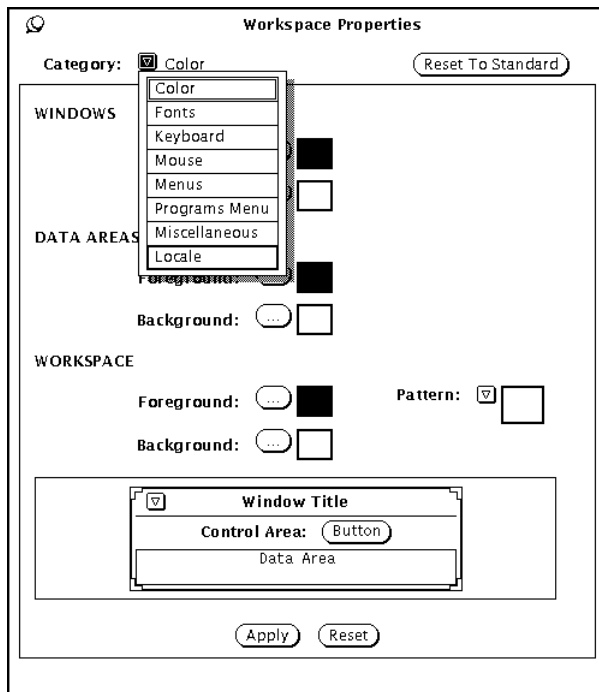
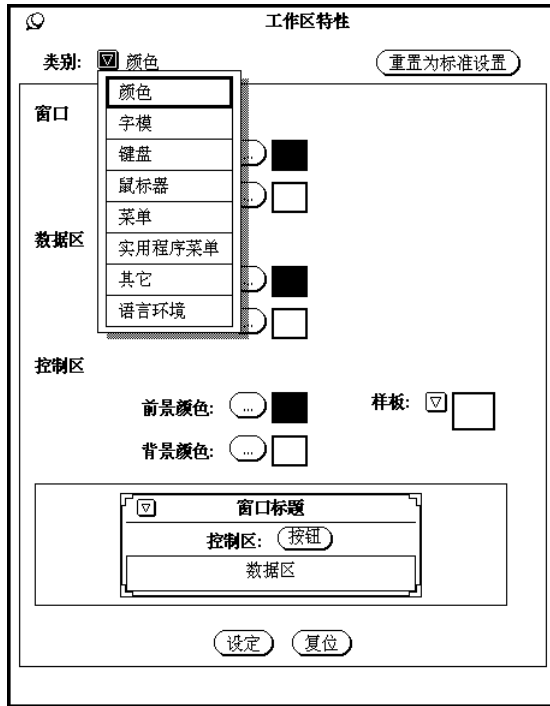
```
system% openwin
```

## Using Localization on the Workspace Properties Worksheet

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

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

The English and Simplified Chinese versions of the Workspace Properties worksheet appear as follows:



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

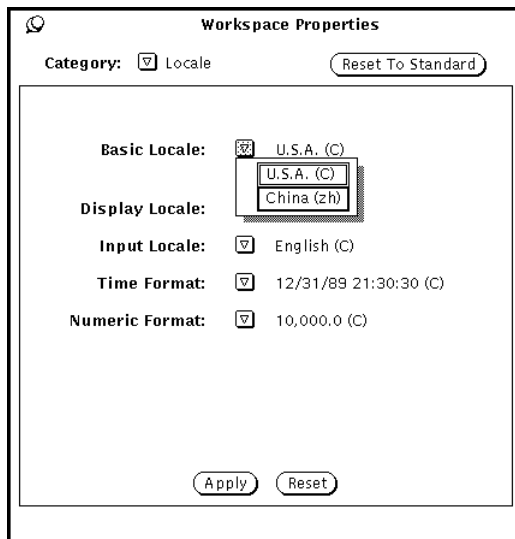
The Locale setting determines which characters (ASCII, Simplified Chinese) appear in new tool windows when they are started. Changing the locale does not affect the appearance or operation of tools windows that are started before the change.

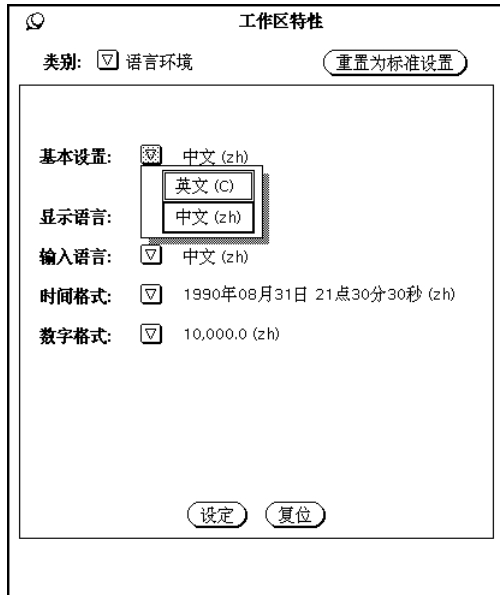
## Changing Your System Locale Setting

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

1. Access the Workspace Properties Locale worksheet.
2. Change the Locale setting by choosing U.S.A. or China on the Basic Locale menu.

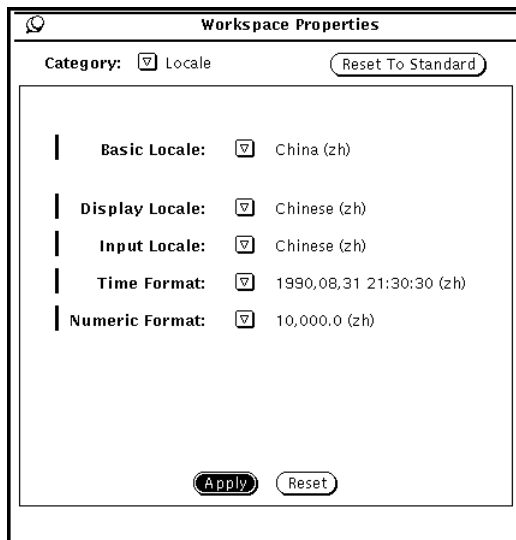
The English and the Simplified Chinese versions of this worksheet appear as follows. In this example, China is selected as the locale.

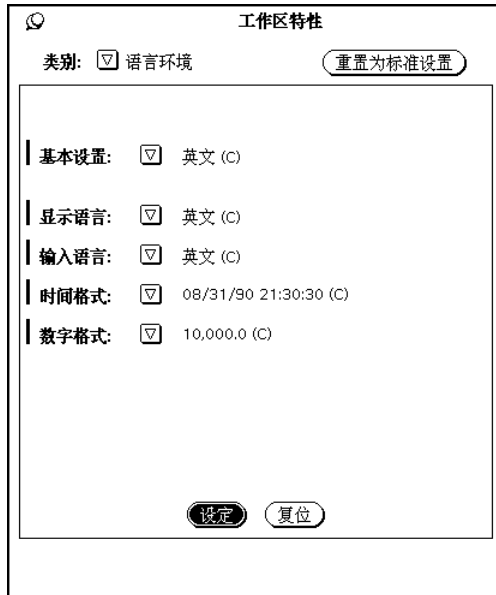




**3. Click the Apply button to set the locale on the system.**

The English version of this menu is on the left and the Simplified Chinese equivalent is on the right.





Clicking Apply then applies these settings, overwriting your current `.OWdefaults` file. This file contains the start-up settings OpenWindows uses to start your applications. For explanations of the contents and functions of the `.OWdefaults` file, refer to *Solaris Internationalization Guide for Developers* and *Simplified Chinese Solaris System Administrator's Guide*.

---

## OpenWindows and `htt`

In the OpenWindows environment, the `.openwin-init` file in the user's home directory is referenced if it exists. Therefore, if the user's `.openwin-init` file lacks a line to start `htt`, `htt` is not started when the OpenWindows environment is started up. (Refer to "Checking Your User Environment for the First Time" on page 113.) For your applications' Simplified Chinese input functions to operate as intended, the `.openwin-init` script must start `htt` before the script starts an application that uses `htt` for Simplified Chinese character input. If `htt` is started after the application, only a root-window style input method server window can be used, as shown in "htt Application Display" on page 22.

## Changes to the .openwin-init File

The OpenWindows WorkspaceUtilitiesSave Workspace function writes or rewrites a user's .openwin-init file when it saves the current configuration of the workspace screen. So after 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 (see above) precedes any line that starts an application that takes Simplified Chinese character input.

Such placement of this command ensures future correct connection to `htt` if the OpenWindows system is restarted later. A manual edit to .openwin-init will be overwritten the next time you use the Save Workspace function. You can save and use your edits by first 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 it in the Properties worksheet as follows:**

The screenshot shows a dialog box titled "文件管理: 特征" (File Management: Features). The "项目:" (Item) dropdown is set to "进一步设置" (Further Settings). The settings are as follows:

- 打印方式:** `cat $FILE | mp -lo | lp -s`
- 文件系统更新周期:** 5 秒
- 显示过滤器模式:** (empty)
- 跟踪符号链:**  是  否
- 缺省编辑器:**  文本编辑  其他:
- 在插入下列设备时打开文件管理窗口:**
  - CDROM:  是  否
- 使用通用图符加快显示:**
  - 软盘:  否  是
  - CDROM:  否  是

Buttons at the bottom:

[ 无输入变换 ]

---

# Customizing Your OpenWindows Workspace

This section describes how to customize your workspace.

## Using Fonts

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

## Simplified Chinese Font Sets

A Simplified Chinese font set is composed of one English font, representing codeset 0 (ASCII) characters in GB1988.1989-0 or ISO8859-1, and one Simplified Chinese font representing gb2312.1980-0 characters.

Simplified Chinese Solaris 8 provides some default font sets defined in application defaults files in `/usr/dt/app-defaults/zh/*`. The following is an excerpt from one of the files, `Dtwm`:

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

This portion of the file refers to a font set containing two fonts previously mentioned that are included in the `zh` locale.

```
"-dt-interface system-medium-r-normal-s serif-14-120-75-75-p-60  
-gb1988.1989-0"-dt-interface system-medium-r-normal-s serif-14-120-75-75-p-120  
-gb2312.1980-0"
```

The first is an English font for codeset 0 (ASCII) character font display. The second is a Simplified Chinese font for codeset 1 (GB2312.1980) character font display.

Note that these fonts are defined in the file  
`/usr/openwin/lib/locale/zh/X11/fonts/75dpi/fonts.alias`.



## Starting Applications With a Specific Simplified Chinese Font Set

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 Simplified Chinese Windows terminal with a specified font set:

```
system% dtterm -fn "-dt-interface system-medium-r-normal-s \
serif-14-120-75-75-p-60-gb1988.1989-0; \
-dt-interface system-medium-r-normal-s \
serif-14-120-75-75-p-120-gb2312.1980-0:"
```

Note the two delimiters used in the font set. 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 Simplified Chinese font name.) Since there are spaces in the long font names, the font list is enclosed in quotation marks.

### ▼ Specifying a Font on the Command Line

A command line that starts a Simplified Chinese OpenWindows application can specify the application’s font. When the current locale is Simplified Chinese, the command uses one of the defined font-set aliases instead (explained in the following section), for example:

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

But 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

Simplified Chinese Solaris software provides several Song-style font sets that combine two or more fonts so both English and Chinese characters can be used together in one window. Each font set comprises one Roman font (ASCII characters) in the ISO8859 standard plus a bitmap Chinese font specified in GB2312-80. A few examples are:

- song12
- song14
- songb14
- song16
- song20

- **song24**

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

## Scaling Application Windows and Fonts

The `$OPENWINHOME/lib/locale/zh/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 starts a Command Tool window that uses 16-point type and is scaled proportionally larger than the default:

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

## Backward Compatibility

---

This appendix contains information for making programs backward-compatible with earlier versions of Asian Solaris Software. Every utility described 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*.

---

## Simplified Chinese Test Utilities

These utilities test various aspects of the Simplified Chinese (GB-2312-80) 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 these tables must be a character in WC, `wchar_t`. For more information, see the `cctype(3x)` man page.

TABLE B-1 Simplified Chinese Character Classification Functions

<b>Routine</b>	<b>Description</b>
<code>ischanzi</code>	Returns true if it is a Hanzi ideogram in GB-2312-80.
<code>iscaccent</code>	Returns true if it is an accent notation in GB-2312-80.
<code>iscphonetic</code>	Returns true if it is a phonetic symbol in GB-2312-80.
<code>iscpinyin</code>	Returns true if it is a Pinyin symbol in GB-2312-80.

**TABLE B-1** Simplified Chinese Character Classification Functions *(continued)*

<b>Routine</b>	<b>Description</b>
<code>iscalpha</code>	Returns true if it is a Roman alphabetic in GB-2312-80.
<code>iscdigit</code>	Returns true if it is a Roman digit in GB-2312-80.
<code>iscnumber</code>	Returns true if it is a number in GB-2312-80.
<code>isclower</code>	Returns true if it is a Roman lowercase in GB-2312-80.
<code>iscupper</code>	Returns true if it is a Roman uppercase in GB-2312-80.
<code>isblank</code>	Returns true if it is a white space character from GB-2312-80.
<code>iscspace</code>	Returns true if it is a space character from GB-2312-80.
<code>iscgen</code>	Returns true if it is a graphic or general symbol in GB-2312-80.
<code>iscsci</code>	Returns true if it is a scientific symbol in GB-2312-80.
<code>iscline</code>	Returns true if it is a ruled line symbol in GB-2312-80.
<code>iscunit</code>	Returns true if it is a unit character in GB-2312-80.
<code>iscparen</code>	Returns true if it is a right or left parenthesis in GB-2312-80.
<code>iscpunct</code>	Returns true if it is a punctuation character in GB-2312-80.
<code>iscgreek</code>	Returns true if it is a Greek character in GB-2312-80.
<code>iscrussian</code>	Returns true if it is a Russian character in GB-2312-80.
<code>iscspecial</code>	Returns true if it is a Greek or Russian character in GB-2312-80.
<code>ischira</code>	Returns true if it is a Japanese Hiragana character in GB-2312-80.
<code>iskata</code>	Returns true if it is a Japanese Katakana character in GB-2312-80.

Two additional routines for Simplified Chinese, `iscgb` and `isceuc`, test for characters from the GB-2312-80 character set. The `iscgb` routine expects a wide character, and `isceuc` expects a GB-2312-80 character in EUC format. For more information, see the `cctype(3x)` man page.

**TABLE B-2** General Simplified Chinese General Character Classification Functions

<b>Routine</b>	<b>Description</b>
<code>iscgb</code>	Returns true if it is in GB-2312-80.
<code>isceuc</code>	Returns true if it is a GB-2312-80 character in EUC format.

---

## Simplified Chinese Conversion Utilities

This section describes functions for wide character and string input and output, character classification, and conversion functions for the Simplified Chinese character sets. Solaris 2.7 software implements a wide character library for handling Simplified Chinese character codes according to industry standards.

Routines that have Chinese language-specific dependency are in their own language-specific library, which is linked with the corresponding C compiler option. Simplified Chinese Solaris `libcle` is linked with `-lcle`

Refer to the appropriate man pages 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 functions described in this section are available, but you should use `iconv()` as a standard function.

Simplified Chinese 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 GB and EUC.
- Between code formats (such as converting between EUC and WC).

Programs using the general multibyte conversion utilities should include the header files `widec.h` and `wctype.h`. Simplified Chinese Solaris specific routines (such as `iscxxx`) are declared in `zh/xctype.h`.

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

The `locale/xctype.h` file declares the Chinese locale-specific routines, which have names of the form `iscxxxx`:

As with the classification functions described in the previous section, the use of these functions can be controlled by the `setlocale` function (described elsewhere in this and other chapters).

Locale-specific conversion routines (such as Chinese `cgbtoeuc`) are contained in the `libc1e` library:

This library can be linked during compilation using the C compiler option `-lc1e`.

## 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)` and `cconv(3)`.

The following routines are in the regular Chinese C library:

**TABLE B-3** Simplified Chinese Case Conversion Functions (declared in `zh/xctype.h`)

Function	Description
<code>tocupper</code>	Converts codeset 1 Roman lowercase to uppercase
<code>tolower</code>	Converts codeset 1 Roman uppercase to lowercase

## Conversion Between Simplified Chinese Codesets

In the Simplified Chinese character sets, the Roman characters and numbers in codeset 0 are repeated in codeset 1. The following functions test wide characters.

**TABLE B-4** Simplified Chinese Codeset Conversion Functions

Function	Description
atocgb	Converts alphabetic or numeric characters in ASCII (codeset 0) to the corresponding characters in GB-2312-80 (codeset 1).
cgbtoa	Converts alphabetic or numeric characters in GB-2312-80 (codeset 1) to the corresponding characters in ASCII (codeset 0).

For further information on these functions, see the `man` page for `ccconv( )` (3x).

## Conversion for Simplified Chinese Character Codes

The following routines do character-based code conversion on the GB-2312-80 character set. They convert characters and strings between EUC format and GB-2312-80 format. To use these routines, the library `libc1e` must be linked using the C compiler option `-lc1e`. For further information, see the `ccconv(3x)` man page.

**TABLE B-5** Simplified Chinese Character-Based Functions

Function	Description
cgbtoeuc	Converts a character in GB-2312-80 format (7 bit) to EUC format
scgbtoeuc	Converts a string in GB-2312-80 format (7 bit) to EUC format
sncgbtoeuc	Converts part of a string in GB-2312-80 format (7 bit) to EUC format
euctocgb	Converts a character in EUC format to GB-2312-80 format (7 bit)
seuctocgb	Converts a string in EUC format in GB-2312-80 format (7 bit)
sneuctocgb	Converts a part of a string in EUC to GB-2312-80 format (7 bit)

---

## Binary Compatibility Package (BCP)

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.

---

---

## Running Earlier Binary Code

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
```

- ◆ **For a current Simplified Chinese Solaris release system to run the compiled binary code of an earlier version of the `textedit` application enter, for example:**

```
system% textedit -lc_displaylang chinese -lc_basiclocale chinese -lc_inputlang  
chinese
```

Incompatibilities between Simplified Chinese Solaris 2.x and 1.x applications do not permit cutting and pasting of Chinese characters between them.



# Running Networked Applications

---

---

## Introduction

You can run Simplified Chinese localized applications on a remote machine as explained in Appendix C, “Networked Applications,” in *Solaris Advanced User’s Guide*. That guide describes the advanced features of the OpenWindows environment that enable you to run applications that reside on another machine on your network.

---

**Note** - Most users do not need to read this appendix. If you want to explore running networked applications, ask your system administrator if there are any special applications available on your network.

---

Applications running on Simplified Chinese OpenWindows 2.x can be displayed remotely on an Simplified Chinese OpenWindows 3.x system. Simplified Chinese OpenWindows 3.x applications may not be able to be displayed remotely in an Simplified Chinese OpenWindows 2.x environment because not all 3.x system fonts are available in 2.x environments.

---

## Instructions for Running Networked Applications

You must follow the information and directions in Appendix C, “Networked Applications,” in *Solaris Advanced User’s Guide*, which you should read thoroughly.

You must additionally make the following adjustments to enable operation of the Simplified 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`, `zh.GBK` or `zh.UTF-8`.
- 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`)

Typically, an application uses the value of the `LANG` environment variable to set each category that has not been set explicitly using `LC_XXX`.

## Sample Command Sequences for Remote Operation

### How to Display Remote OpenWindows V2.x on a Local OpenWindows V3.x System

The following sequence of commands is an example of starting a Shell Tool on a remote Simplified Chinese Solaris 1.x (including Simplified Chinese OpenWindows 2.x) machine. In this example, your local machine is running Solaris 2.x (including Simplified Chinese OpenWindows 3.x), and the remote machine (where you start the Shell Tool) has Simplified Chinese Solaris 1.x, including Simplified Chinese OpenWindows 2.x:

1. **Type the following command.**

```
local_machine% xhost +remote_machine
```

2. **Log in on 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 chinese
```

**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 an OpenWindows application, for example, a Shell Tool:**

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

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

# Glossary

---

<b>ANSI</b>	American National Standards Institute. ANSI proposes standard definitions for different computing languages. The most recent standard for the C language, prepared by the ANSI C X3J11 Committee, includes library functions for computing with multibyte characters for international usage, as well as a new data type, <code>wchar_t</code> , for dealing with four-byte characters. This standard is not completed, so it is referred to as the “proposed ANSI C standard,” or ANSI C-X3J11.
<b>ASCII</b>	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.
<b>Category</b>	In the Simplified 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 <i>Month, Day, Year</i> ; while in another country it might be <i>Day, Month, Year</i> . The date and time can be thought of as one category of a local language. Categories also refer to the program categories, the environment variables that are related to categories, and the ANSI localization tables for each category.
<b>Character Set</b>	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.

<b>Code set</b>	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.
<b>EUC</b>	Extended UNIX Code. Describes four code sets modeled 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.
<b>ISO</b>	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.
<b>Locale</b>	A locale describes a language or cultural environment. Its setting affects the display or manipulation of language-dependent features. Simplified Chinese Solaris software provides <code>c</code> for U.S.A and <code>zh</code> for Simplified Chinese.
<b>POSIX</b>	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.
<b>Unicode</b>	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 the X/Open System Interface Portability Guide.





# Index

---

## A

- Address Manager 14
- aliases
  - setting locales with 18
- App Manager 15
- application windows
  - scaling 122
- applications
  - interaction with htt 22
  - running remotely 129
  - starting with specific font set 121
- ASCII
  - typing 32
- ASCII keyboard 14
- Audio Tool 15

## B

- Background option, in File menu 24
- Backspace key 33
- backward compatibility 123
- BDF, xtobdf 86
- binary compatibility package (BCP) 128
- bitmap fonts, printing with 107

## C

- Calculator 15
- Calendar Manager 15
- calligraphy strokes 50

- candidate select key, modifying 75
- candidates 14
- characters 17
  - Hanzi 33
- Chinese characters 17
- Chinese Star keyboard mapping 44
- CID fonts 100
- clekbd command 50, 56
- Clock 15
- code tables 79, 83
- codeset conversion 126
- comma, for toggling input method choices 33
- command line options, for htt 24
- Common Desktop Environment (CDE) 17
- compatibility, backward 123
- Console 15
- Control keys
  - for input modes 31
  - listing 65
- Control-H 33
- Control-spacebar toggle 31
- conversion
  - in codesets 126
- conversion manager 14
- Conversion Manager 33
- conversion modes 31
- conversion utilities 125
- converting
  - GB-2312-80 character codes 127

- converting files 87
- Convtool 86
- .cshrc file 19, 113
- customizing
  - htt 25

## D

- default locale, setting 18
- Delete key 33
- DeskSet tools 112
- desktop tools 14
- dictionary files 43
- DISPLAY environment variable 130
- Display PostScript System (DPS) 91
- .dtlogin file 18
- dtwm 13

## E

- editing fonts 95
- environment variables 113, 130
  - DISPLAY 130
  - LANG 130
- Exit option, in File menu 24

## F

- FangSongTi font 107
- file conversion 86
- File Finder 15
- File Manager 15
- File menu 24
- files, conversion 87
- focus area, Intelligent Pinyin input 63
- font directories, changing 21
- font sets 121
  - Simplified Chinese 120
- fontedit command 96
- fonts 92
  - font names and font sets 121
    - GB2312-80 121
    - ISO8859 121
    - Portable Compiled Format (PCF) 95
    - scaling 122
    - sdtudctool utility 100
    - specifying in xetops 107
    - specifying at command line 121
    - UDC directory 102

- Function keys
  - for input modes 31
  - listing 65

## G

- GB-2312-80 code conversion 127
- GB13000-1 17
- GB2312-80
  - fonts 121
  - in Location mode 33
- GBK Code input method 31, 66
- GBK switch key, modifying 75
- Golden input conversion mode 31, 55, 61
  - phrase input 60, 61
  - single character input 56, 60
- Graphical User Interface 14
- GUI 14

## H

- Hanja input method 31
- Hanja text 69
- Hanzi characters 33
- HeiTi font 107
- Help 15
- Help menu 24
- htt 119
  - command line options 24
  - customizing 25
  - display 23
  - Property Manager 23
  - startup 119
  - startup script 27
  - startup sequence 27
- htt server 21
- htt\_props 22
- htt\_server 22

## I

- Icon Editor 15
- icon, htt 21
- iconv 86
- iconv command 87
- ideograms 29
- Image Tool 15
- input

- Intelligent Pinyin mode 62
- input code tables 79
- input method
  - creating new 79
  - GBK Code 31
  - Golden 31
  - Hanja 31
  - Intelligent Pinyin 31
  - Japanese 31
  - Location 31
  - New QuanPin 31, 37
  - New ShuangPin 31, 37
  - Pinyin 31
  - Quanpy 31, 48
  - ShuangPin 43
  - Simplified Chinese Symbol 31
  - Stroke 31
  - Unicode 31
  - Zhuyin 31
- Input Method Properties Editor 75
- input method server (htt) 21
- input methods 31
  - Location 105
  - toggling among 33
- Intelligent Pinyin input method 31
- Intelligent Pinyin input mode 62
- Intelligent\_ABC keyboard mapping 44
- intermediate representation 14
- ISO/IEC10646-1 characters 17
- ISO8859 standard, fonts 121

## J

- Japanese input method 31
- Japanese text 67

## K

- KaiTi font 107
- Kanji input method 67
- keybind.dat file 82
- keyboard
  - Chinese 56
  - mapping Chinese 56
  - Pinyin keyboard map 36
- keyboard mappings 44
- keyboards 14
- keys, input conversion control 31

- keystroke map 51
- keystrokes, Pinyin 36
- Korean text 69

## L

- LANG
  - environment variable 130
- LANG environment variable 113
- Level 1 characters 33
- Level 2 characters 33
- locale
  - Workspace Properties 116
- locale input
  - turning on/off 24
- locale setting
  - in OpenWindows 116
- locales
  - overview 17
  - setting 18
  - setting default 18
  - zh.GBK 17
- localization
  - Localization Category menu 114
  - OpenWindows 111
  - OpenWindows menu 116
- Location code input mode 33, 105
- Location input method 31
- login 113
- lookup choice area 30
- Lookup Table 24
- Lookup Window
  - customizing 26

## M

- Mailer 15
- mailx utility 85
- Motif 14

## N

- networked applications 129
- New QuanPin 31
- New QuanPin input method 37
- New ShuangPin 31
- New ShuangPin input method 37

## O

- olwm 111
- on/off key, customizing 24
- OPEN LOOK DeskSet tools 112
- .openwin-init file 114, 118, 119
- OpenWindows 17
  - scaling application windows and fonts 122
  - starting 113, 114
  - V2 applications 128
- .OWdefaults file 116

## P

- page turn key, modifying 75
- paper size, with xetops utility 107
- Performance Meter 15
- period, for toggling input method choices 33
- phonetic input, Golden 55, 61
- phonetic keyboard, Chinese 56
- phonetics, Pinyin 36
- phrase input, Chinese 61
- phrase text, typing 60
- Pinyin
  - input mode 35
  - phonetics and keystrokes 36
  - phonetics with Golden 55, 61
- Pinyin input method 31
- Portable Compiled Format (PCF) fonts 95
- PostScript fonts 91
- preedit area 14, 30
- Preedit Status 24
- Preedit/Status window
  - customizing 25
- Print Manager 15
- printing 105
  - in OpenWindows 119
  - using xetops print filter 106
  - with xetops utility 107
- Process Manager 15
- .profile file 19
- Properties Editor tool 32
- Property Manager
  - htt 23
- PSWDisplayText() 94
- PyCiku.dat dictionary file 43

## Q

- Quanpy 31
- Quanpy input method 48
- QuWei input method, *see* Location code input mode

## R

- radicals, Chinese 50
- remote applications 129
- remote environment variables 130
- reverse video 14

## S

- scaling windows and fonts 122
- scripts
  - htt startup script 27
  - sdtudctool, using 100
- separators 43
- setting locale
  - in OpenWindows 117
- ShuangPin input method 43
- Simplified Chinese
  - font sets 120
- Simplified Chinese keyboard 14
- Simplified Chinese locale in OpenWindows 113
- Simplified Chinese Symbol input method 31
- Simplified Chinese text, input modes 29
- Snapshot 15
- Song fonts 92
- SongTi font 107
- status area 14, 30
- Stroke input method 31
- Stroke input mode 50
  - characters 51
- strokes 50
- Style Manager 15
- Symbol input mode 65

## T

- talk command 86
- TCA\_CNS 11643-92 17
- terminal window 17
- text
  - Chinese 29

- intermediate representation of 14
- Text Editor 15
- tools
  - DeskSet 112
  - desktop 14
  - Properties Editor 32
- Traditional Chinese text 71
- TrueType fonts, printing with 107

## U

- Ud.Ciku.dat dictionary file 43
- UDC directory 102
- Unicode input method 31, 74
- utilities
  - conversion 125

## W

- window environments 17
- Window Manager 13, 111
- windows
  - Simplified Chinese characters allowed in 29

- status area 14
- Workspace Properties 114
  - settings 116

## X

- X Windows and htt 21
- .Xdefaults, *see* .OWdefaults file
- xetops print filter 106, 119
  - examples 107
- XIM 21
- .xinitrc file 114
- Xlib 14
- xtobdf command 86
- XView 111

## Z

- zh locale 17
- zh.GBK locale 17
- Zhuyin input method 31, 71
- ZiRanMa keyboard mapping 44