Traditional Chinese Solaris System Administrator's Guide

Contents

Preface

Traditional Chinese Solaris System Administrator's Guide provides system administration information specific to Traditional Chinese Solaris™ operation in the Common Desktop Environment (CDE). This guide also includes some additional information that advanced users and developers can use to access and control the features of the Traditional Chinese Solaris operating environment.

Who Should Use This Book

You should read this guide if:

- You need specific instructions on how to set up features for users.
- You are a system administrator who has not used the Traditional Chinese Solaris operating environment, CDE.
- You are a developer who needs information on accessing and controlling the Traditional Chinese features of the Traditional Chinese Solaris operating environment.
- You are an advanced user who wants to use or customize the Traditional Chinese Solaris operating environment.
- You want information on a variety of details internal to the operation of the Traditional Chinese Solaris operating environment.

You should already be familiar with Sun's standard product documentation and the documentation of the window system that you are using. This guide adds only Traditional Chinese features.

Before You Read This Book

Before you read this book, please review the product overview and any last-minute changes that arrived too late to be included in this document:

- *Traditional Chinese Solaris Release Overview*

Make sure to install your system properly as described in the document appropriate to your hardware platform:

- *Solaris 9 Installation Guide*

Each chapter of this manual addresses a different aspect of administration of the Traditional Chinese Solaris operating environment. Some chapters give step-by-step instructions for using or customizing product features.

Chapter 1, "Starting the Traditional Chinese Solaris Software," introduces the Traditional Chinese Solaris operating environment, including CDE and the locales included in the product.

Chapter 2, "System Environment," describes advanced ways to use Traditional Chinese window system features.

Chapter 3, "Setting Up Traditional Chinese Solaris Printing Facilities," describes the set up for printers that can print Traditional Chinese output and the use of PostScript™ printers.

Chapter 4, "TTY Environment and Support," covers setting terminals to use the proper protocols for the input and display of Traditional Chinese characters.

Related Books

The following books are related to the topic of this book and may prove helpful for further reading.

For information on how to use the window system and associated applications:

- *Solaris Advanced User's Guide*

For information about how to develop applications for this Traditional Chinese Solaris release:

- *International Language Environments Guide*

What Typographic Changes Mean

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

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

Shell Prompts in Command Examples

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

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

Starting the Traditional Chinese Solaris Software

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

The Traditional Chinese Solaris operating environment provides the Common Desktop Environment (CDE) window environment. CDE is a fully internationalized environment. The Traditional Chinese Solaris product includes the following locales:

- `C` — ASCII English environment.
- `zh_TW` — Traditional Chinese environment in extended Unix Code (EUC) that supports CNS 11643–1992 standard.
- `zh_TW.BIG5` — Traditional Chinese environment in Big5 code.
- `zh_HK.BIG5HK` — Traditional Chinese environment in Big5-HKSCS code that supports the Hong Kong Supplementary Character Set (HKSCS), which is a supplementary character set of the Big-5 and ISO 10646 coding schemes.
- `zh_HK.UTF-8` — Traditional Chinese (Hong Kong) environment in Unicode 3.1.
- `zh_TW.EUC` — Symbolic links to `zh_TW` locale.
- `zh_TW.UTF-8` — Traditional Chinese environment in Unicode 3.1.

Applications Defaults Files

The Traditional Chinese CDE includes three directories for applications defaults. One is for system-wide defaults, and two are specific to locale features:

- The `/usr/dt/app-defaults/C` directory stores system wide application defaults. These values are for the `C` locale.

- The `/usr/dt/app-defaults/zh_TW` directory stores application defaults that are specific to the `zh_TW` locale.

- The `/usr/dt/app-defaults/zh_TW.BIG5` directory stores application default values specific to the `zh_TW.BIG5` locale.

- The `/usr/dt/app-defaults/zh_TW.UTF-8` directory stores application default values specific to the `zh_TW.UTF-8` locale.

- The `/usr/dt/app-defaults/zh_HK.BIG5HK` directory stores application default values specific to the `zh_HK.BIG5HK` locale.

- The `/usr/dt/app-defaults/zh_HK.UTF-8` directory stores application default values specific to the `zh_HK.UTF-8` locale.

System Environment

Users can change their locale settings with shell environment variables. Each category names an existing locale. The `setlocale()` function directly sets or queries the setting of these categories. Internationalized functions use these settings to access the appropriate tables for the desired locale.

Environment variables can indirectly set the categories: when `setlocale()` sets the categories to the default setting for that site, it uses the setting of each environment variable to set the associated categories. The `setlocale()` function does not change the settings of environment variables, it only reads their settings.

Changing the Default Locale

You can change the default locale system-wide with the following procedure.

1. **Edit the** `/etc/default/init` **file by adding or changing the line.**

   Substitute `C`, `zh_TW` `zh_TW.BIG5`, `zh_TW.UTF-8`, `zh_HK.BIG5HK` or `zh_HK.UTF-8` for *locale*.

   `LANG=`*locale*

2. **Have all users exit CDE.**

3. **Type the following commands:**

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

4. **Type the following commands:**

   ```
   % su
   # reboot
   ```

11

Locale and Category Terminology

The terms *locale* and *category* relate to each other as follows:

- A *locale* includes specification of a language, territory, code set, and other features. The Traditional Chinese Solaris operating environment includes the following locales:

  - `C`—For the ASCII English environment, the locale must be set to `C`.

  - `zh_TW`—For the Traditional Chinese environment in EUC, the locale must be set to `zh_TW`.

  - `zh_TW.BIG5`—For the Traditional Chinese environment in Big5, the locale must be set to `zh_TW.BIG5`.

  - `zh_TW.UTF-8`—For the Traditional Chinese environment in Unicode, the locale must be set to `zh_TW.UTF-8`.

  - `zh_HK.BIG5HK`—For the Traditional Chinese environment in Unicode, the locale must be set to `zh_HK.BIG5HK`.

  - `zh_HK.UTF-8`—For the Traditional Chinese environment in Unicode, the locale must be set to `zh_HK.UTF-8`.

- A *category* is a set of features that comprises a locale. For example, character displays or time/date representations, whose behavior depends on the *locale*. Traditional Chinese Solaris categories include the following:

  - `LC_CTYPE` sets the character-type for classification and conversion.

  - `LC_TIME` sets the locale for representation of date and time.

  - `LC_NUMERIC` sets the number representation locale (used also for I/O).

  - `LC_MONETARY` sets the currency representation locale.

  - `LC_MESSAGES` sets the language locale for messages to users.

  - `LC_COLLATE` sets the locale-dependent collation of strings.

The environmental variable `LC_ALL` explicitly sets the same locale for all categories; it has the highest priority. If categories or `LC_ALL` are not set, the `LANG` environmental variable will determine the category setting.

Interfacing With the Traditional Chinese Solaris Localization Facility

At the C shell level, each environment variable can be set to *locale* (`zh_TW`, `zh_TW.BIG5`, `zh_TW.UTF-8`, `zh_HK.BIG5HK` or `zh_HK.UTF-8` for Traditional Chinese, or C for ASCII) by a shell command as follows:

- **C shell users can enter a shell command as follows:**

  ```
  system% setenv LC_TIME locale
  ```

- **Bourne shell (`sh`) users can use `set` or `export`:**

  ```
  $ set -a LC_TIME
  $ LC_TIME=locale
  ```

  or

  ```
  $ LC_TIME=locale
  $ export LC_TIME
  ```

Making `zh_TW`, `zh_TW.BIG5`, `zh_TW.UTF-8`, `zh_HK.BIG5HK` or `zh_HK.UTF-8` the *locale* allows the user's environment to display time in Traditional Chinese format and text. A user can also define a mix of locales for the working environment. For example, characters can be typed and converted in Traditional Chinese, time can be displayed in French format, and messages can appear in English.

Many users work in a single cultural environment. The `LC_ALL` and `LANG` environment variables set the system default for all categories. For example, these C shell commands set the system default for all categories to *locale*.

```
system% setenv LC_ALL locale
system% setenv LANG locale
```

System administrators or users can set the default and the `setenv` syntax can be used in programming.

This setting is put into effect the next time a `setlocale()` function call in an application program line sets a category to the default setting:
`setlocale(LC_XXX"")`

Setting Up Traditional Chinese Solaris Printing Facilities

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

- Line printer containing built-in Traditional Chinese fonts
- PostScript-based printer containing built-in scalable fonts
- Any PostScript-based printer for bitmap printing

The system administrator installs printer(s) as described in the printer product documentation. Then users can print Traditional Chinese text using procedures described in this chapter.

Follow the printer documentation for physically connecting the printer. Then use the following instructions.

Line Printer Support

For the Traditional Chinese Solaris operating environment to run a line printer, the printer must recognize at least one of the appropriate code sets:

- EUC
- UTF-8

Traditional Chinese Solaris Code Filters

EUC to Big5 Filter

A printer that does not support EUC needs filters that convert EUC files for printing. For example, the following command sequence tells LP, the print service, that printer `lp1` accepts only Big5 format files. This command line also installs printer `lp1` on port `ttya`. The *lpadmin(1)* man page explains this command more fully.

```
# lpadmin -p lp1 -v /dev/ttya -I Big5
# accept lp1
# enable lp1
```

An `lpfilter` command line like the following can be used in the process of printing files whose format is not supported by the printer:

```
# lpfilter -f filter-name -F pathname
```

The above command tells LP that a converter called *filter-name* (for example `euctobig5`) is available through the filter description file named *pathname*. This filter takes default type file input and converts it to Big5 format by using `euctobig5`. The content of *pathname* can be as follows:

```
Input types: simple
Output types: BIG5
Command: euctobig5
```

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

```
system% lp -d lp1 EUC-filename
```

To print a Big5 format file, use the following command:

```
system% lp -d lp1 -T BIG5 Big5-filename
```

Using Configuration Files for Printing With an Epson Dot Matrix Printer

Traditional Chinese Solaris software provides configurable files that allow users to configure the available built-in font ranges according to their own equipment. Traditional Chinese Solaris software provides a configuration file that defines the built-in fonts in the Epson dot matrix printers.

Modify the configuration file according to your printing equipment before using it:

- Modify `/usr/lib/lp/files/cns.epson.conf` if you are working with a `zh_TW` file.
- Modify `/usr/lib/lp/files/big5.epson.conf` if you are working with a `zh_TW.BIG5` file.

Printing a CNS File to an Epson Dot Matrix Printer

For instance, the following configuration file,
`/usr/lib/lp/files/cns.epson.conf`, is used for setting up an Epson
LQ-1070C+ Chinese dot matrix printer. In the example, the Epson LQ-1070C+ is
composed of built-in Traditional Chinese fonts for CNS 11643 plane 1 and plane 2. You
can specify available fonts in different planes and in different directories.

```
# @(#)cns.epson.conf    1.5 96/04/10 SMI; ALE
# CNS Print Filter Configuration File for Epson Dot Matrix Printers
#

ACCEPT CODE RANGE:
    PLANE 1:
        0x2121 - 0x234e
        0x2421 - 0x2570
        0x4421 - 0x7d4b
    PLANE 2:
        0x2121 - 0x7244

FONT FILES:
    PLANE 1:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/mingL1-24.pcf.Z
    PLANE 2:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/mingL2-24.pcf.Z
    PLANE 3:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/mingL3-24.pcf.Z
    PLANE 4:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/mingL4-24.pcf.Z
    PLANE 5:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/mingL5-24.pcf.Z
    PLANE 6:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/mingL6-24.pcf.Z
    PLANE 7:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/mingL7-24.pcf.Z
    PLANE 8:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/sungNotdef-24.pcf.Z
    PLANE 9:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/sungNotdef-24.pcf.Z
    PLANE 10:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/sungNotdef-24.pcf.Z
    PLANE 11:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/sungNotdef-24.pcf.Z
    PLANE 12:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/sungNotdef-24.pcf.Z
    PLANE 13:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/sungNotdef-24.pcf.Z
    PLANE 14:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/sungNotdef-24.pcf.Z
    PLANE 15:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/sungNotdef-24.pcf.Z
    PLANE 16:
        /usr/openwin/lib/locale/zh_TW/X11/fonts/75dpi/sungNotdef-24.pcf.Z
```

An `lpfilter` command line such as the following can be used to print files whose format is not supported by the printer:

```
# lpfilter -f filter-name -F filename
```

The above command tells LP that a converter called *filter-name* (for example `cns.epson.filter`) is available through the filter description file named *filename*.

If you are installing an Epson dot matrix printer, the content of the filter description file, `cns.epson.fd`, should be as follows:

```
Input types: simple
Output types: EUC
Command: cns.epson.filter
```

This filter takes the default type file input and converts it to a dot matrix data stream by using the `cns.epson.filter`.

For example, the following command sequence tells LP, the printer service, that printer `lp2` accepts only EUC format files. This command line also installs printer `lp2` on the bidirectional parallel port `/dev/bpp0`. The *lpadmin(1)* man page explains this command more fully.

```
# lpadmin -p lp2 -v /dev/bpp0 -I EUC
# accept lp2
# enable lp2
```

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

```
system% lp -d lp2 EUC-filename
```

Printing a Big5 File to an Epson Dot Matrix Printer

Traditional Chinese Solaris software provides a configuration file that defines for `zh_TW.BIG5` files the built-in fonts in the Epson dot matrix printers. Modify the configuration file according to your printing equipment before using it. The contents of the `/usr/lib/lp/files/big5.epson.conf` configuration file can be as follows:

```
# @(#)big5.epson.conf     1.6 96/05/06 SMI; ALE
#
# Big5 Print Filter Configuration File for Epson Dot Matrix Printers
#

ACCEPT CODE RANGE:
    PLANE 1:
        0xA140 - 0xA3E0
        0xA440 - 0xC6A1
        0xC940 - 0xF9D5

FONT FILES:
```

```
PLANE 1:
    /usr/openwin/lib/locale/zh_TW.BIG5/X11/fonts/75dpi/mingL1B5-24.pcf.Z
```

An `lpfilter` command line such as the following can be used to print files whose format is not supported by the printer:

lpfilter -f *filter-name* **-F** *filename*

The above command tells LP that a converter called *filter-name* (for example `big5.epson.filter`) is available through the filter description file named *filename*.

If you are installing an Epson dot matrix printer, the content of the filter description file, `big5.epson.fd`, should be as follows:

```
Input types: simple
Output types: BIG5
Command: big5.epson.filter
```

This filter takes the default type file input and converts it to a dot matrix data stream by using the `big5.epson.filter`.

For example, the following command sequence tells LP, the printer service, that printer `lp2` accepts only `zh_TW.BIG5` format files. This command line also installs printer `lp2` on the bidirectional parallel port `/dev/bpp0`. The *lpadmin(1)* man page explains this command more fully.

lpadmin -p lp2 -v /dev/bpp0 -I BIG5
accept lp2
enable lp2

To print a file that contains Chinese characters in `zh_TW.BIG5` format to an Epson dot-matrix printer, use the following command:

system% **lp -d lp2** *BIG5-filename*

Laser Printer Support

To print Traditional Chinese characters using a PostScript-based printer, a Traditional Chinese Solaris software application must have the Traditional Chinese Solaris `xetops`, `xutops` or `mp` utility to print EUC, BIG5, BIG5HK or UTF-8 files.

Using `xetops` and `xutops` Utilities

The `xetops` and `xutops` utilities produce bitmapped graphics as printed images. Traditional Chinese software includes the `xetops` and `xutops` utilities so any system can print Traditional Chinese text on a PostScript printer. The `xetops` utility in EUC locale and `xutops` Utility in `UTF−8` locale may no longer be supported in the future.

- `xetops` handles `EUC` or `BIG5`files in the `zh_TW` and `zh_TW.BIG5`locales respectively.
- `xutops` handles files in the `zh_TW.UTF-8` locale

Using `xetops` and `xutops` is described in *Traditional Chinese Solaris User's Guide*, in the chapter "Traditional Chinese Printing Facilities," and in the *xetops(1)* and *xutops(1)* man pages.

A typical command line for printing a file named *filename* containing Traditional Chinese characters with `xetops`, would be as follows:

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

The syntax for `xutops` is similar:

```
system% pr filename | xutops | lp
```

Make *filename* the name of the file to print. This file can contain ASCII/English characters as well as Traditional Chinese.

Using the `mp` Utility

The `mp` utility supports all Asian locales. As a printing filter, `mp` generates a pretitified version of contents in PostScript format. The Postscript output file contains glyph images from Solaris system-resident scalable or bitmap fonts, depending on each locale's system font configuration for `mp`. As a print filter, `mp(1)` is enhanced in the Solaris 9 environment to print either `EUC`file in the `zh_TW` locale, `BIG5` file in the `zh_TW.BIG5` locale, `BIG5HK` file in the `zh_HK.BIG5HK` locale or `UTF-8` files in both the `zh_TW.UTF-8` and `zh_HK.UTF-8` locales.

Using `mp` is described in *Traditional Chinese Solaris User's Guide*, in the chapter "Traditional Chinese Printing Facilities," and in the *mp(1)* man page.

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

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

Make *filename* the name of the file to print. This file can contain ASCII/English characters as well as Traditional Chinese.

TTY Environment and Support

This chapter assumes you are familiar with:

- How the Solaris operating environment communicates with external devices using STREAMS and `ioctl`.
- How different terminal types are supported by `termcap` and `terminfo`.

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

TTY Streams

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

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

Traditional STREAMS

The traditional STREAMS TTY environment contains a raw device driver, a line discipline module, and a stream head. The raw device driver provides an I/O interface between the kernel and the hardware. Because it is closest to the physical

hardware, it provides basic communication protocols, baud rate switching, and other low level services. The line discipline module is a set of instructions or disciplines that transforms the raw data to processed data. This includes handling the delete character, line kill character, and others. The stream head provides an interface between the user's process and the stream.

Traditional Chinese Solaris STREAMS

The Traditional Chinese Solaris operating environment uses the modular nature of STREAMS to support Traditional Chinese. In addition to the traditional TTY modules, this product implements code conversion in STREAMS. Chinese input is typically supplied by many existing Traditional Chinese TTYs and is not available in the Traditional Chinese Solaris TTY environment.

The Traditional Chinese Solaris operating environment enhances the traditional modules. Its line discipline handles proper cursor movement for wide characters as well as normal protocols. The Traditional Chinese Solaris software code conversion modules convert between two different character code formats such as between Big5 and EUC.

Code conversion depends on the appropriate flags or parameters being set. For example, if a Big5 code terminal is being used, the input from the terminal is converted to EUC and the output to the terminal is converted to Big5 code.

The major modules that can be pushed onto the stream are `ldterm` and `big5euc`:

- `ldterm(7)` is a generic EUC line discipline module. It processes all normal line discipline functions and also handles proper cursor movement and backspacing for wide characters (EUC)
- `big5euc` controls code conversion between Big5 and EUC

TTY Utilities

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

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

EUC `ioctl` Features

The Traditional Chinese Solaris operating environment uses ioctl(2) STREAMS commands for general EUC handling. The following is a summary of these `ioctl` calls and their effects:

TABLE 4–1 `ioctl` Requests and Descriptions

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

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

termcap

termcap and terminfo are the databases used to tailor the terminal characteristics for an application. The following are extensions to the termcap database:

TABLE 4–2 `termcap` Variables and Descriptions

Variable	**Description**
dv	Device type: language and codeset
ci	Init sequence for multiple codesets
s0	Shift into codeset 0
s1	Shift into codeset 1
s2	Shift into codeset 2
s3	Shift into codeset 3

terminfo

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

TABLE 4–3 `terminfo` Variables and Descriptions

Variable	Capname	Tc	Description
device_type	devt	dv	Device type: language and codeset
code_set_init	csin	ci	Init sequence for multiple codesets
set0_des_set	s0ds	s0	Shift into codeset 0
set1_des_set	s1ds	s1	Shift into codeset 1
set2_des_set	s2ds	s2	Shift into codeset 2
set3_des_set	s3ds	s3	Shift into codeset 3

TTY Commands

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

`setterm` Command

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

`setterm` uses a terminal device name that reflects the `devt` (device type) field in the `terminfo` database for configuring STREAMS modules for a TTY port. This device name is matched with an entry of the same name in the setterm configuration file, `/usr/share/lib/setterm/zh_TW/conf.file`. This entry contains detailed instructions on which modules to pop and push in order to properly configure the STREAMS environment.

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

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

The Traditional Chinese Solaris operating environment provides special purpose modules to enable/disable code conversion and properly handle multibyte and wide characters. Pushing a module onto the stack, enables the corresponding conversion. The default state for conversion is enabled.

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

/bin/stty Command

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

The following command queries the user's environment for information on EUC code-set width and sets that information in the line discipline:

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

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

TTY Setup Examples

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

Configuring STREAMS for Traditional Chinese Solaris Software

To explicitly configure the STREAMS module for a Big5 terminal use:

```
setterm -x big5
```

This usage is independent of terminfo.

Further consider using a VT-100 terminal (which is Big5 compatible) on a system with an entry like the following (which is appropriate for such a terminal) in the `terminfo` database:

```
vt100-b|VT-100-compatible with Big-5 chars,
lines#40, .csin=E(BE)IE[?1h, csin=E(BE)I, devt=PACKBig5,
s0ds=E(H, s1ds=E$@, s2ds=E(H, use=cit600,
```

A configuring command that references this entry would be:

```
setterm -t vt100-b
```

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

```
#
big5throw \
          popto zs|mcp|mti|ptem \
          push big5euc \
          push ldterm \
          push ttcompat \
          run {stty defeucw} \
          catch
#
GenericEUC|EUCthrow \
          popto zs|mcp|mti|ptem \
          push ldterm \
          push ttcompat \
          run {stty defeucw} \
          catch
#
ASCIIthrow \
          popto zs|mcp|mti|ptem \
          push ldterm \
          push ttcompat \
          catch
#
```

For more information, refer to the `setterm(1)` man page. Once configured, conversion is enabled by default. Applications can also set features through `ioctl()` function calls.

Terminal Support

The Traditional Chinese Solaris operating environment supports Big5 terminals. The terminals should have built-in Traditional Chinese fonts and input methods.

Installing a Terminal

If you have not added a terminal to your system before, first try installing a terminal in ASCII mode only. For more information, see *Solaris System Administration Guide*.

Serial Port Interface for Adding Terminals

Serial Ports is available from the Admintool menu to configure serial ports for terminals. Serial Ports provides the easiest method of installing a terminal. Serial Ports is invoked by `admintool`. For more information on `admintool`, see *Solaris System Administration Guide*.

▼ Accessing Serial Ports

1. **Become superuser.**

   ```
   system% su
   ```

2. **Type** `admintool.`
   The admintool menu will appear.

   ```
   system# admintool
   ```

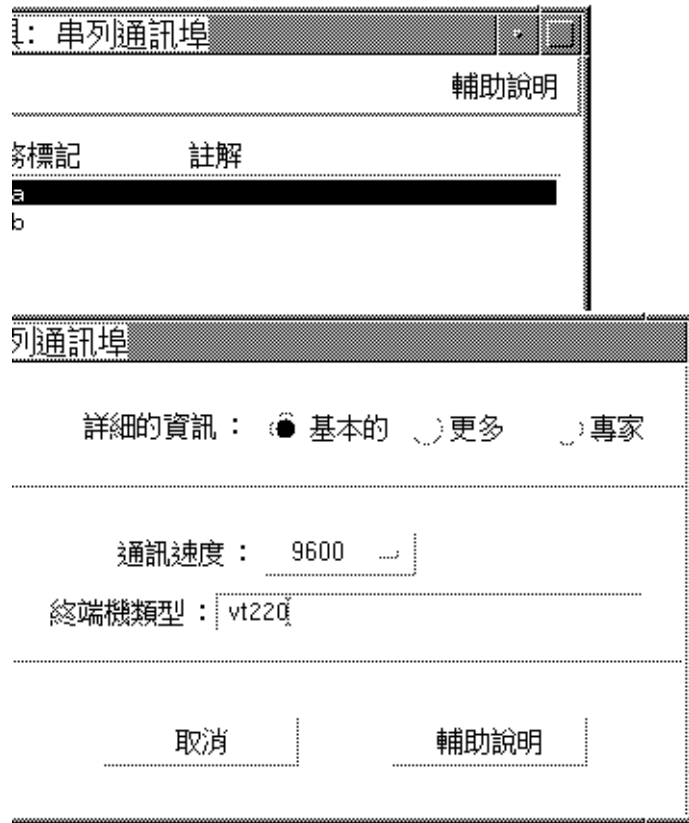3. **Select the Serial Ports icon.**

▼ Using Serial Ports Menus

A Chinese terminal that supports CNS 11643 is installed as you would install an ASCII terminal.

1. **Select Edit on the Serial Ports menu.**
   Serial Ports: Modify Service submenu appears.

2. **On Admintool: Modify Service submenu select Enabled, Baud Rate 9600, and enter the terminal type.**

3. **Under Expert Options, select "Create utemp entry" and enter the appropriate module in the Streams Modules field:**

Note – To install a terminal that supports Big5 code for the Traditional Chinese Solaris operating environment, add big5euc to the Streams Modules field.

管理工具：串列通訊埠

輔助說明

器	服務標記	註解
	ttya	
	ttyb	

修正串列通訊埠

詳細的資訊： ◯基本的 ◯更多 ⦿專家

通訊速度： 9600

終端機類型： vt220

登入提示： ttya login

註解：

服務標記： ttya

通訊埠監控器標記： zsmon

服務： /usr/bin/login

串流模組： ldterm,ttcompat,big5euc

逾時（秒）： 從不

重設　　　　　　取消　　　　　　輔助說明

Command Line Interface for Adding Terminals

The following procedure is required to set up a terminal on `ttya` port via the command line:

1. **Determine the port monitor version number.**

   The port monitor version number will display.

   ```
   # ttyadm -V
   ```

2. **Enter the following commands, substituting the port monitor version number for** *ver*.

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

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

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

   For EUC terminals, use the following command:

   ```
   # pmadm -a -p zsmon -s ttya -i root -fu -v ver -m " 'ttyadm -S y \
     -T terminal_type -d /dev/ttya -l 9600 -m ldterm,ttcompat -s \
     /usr/bin/login'"
   ```

   For Big5 code terminals, use the STREAMS module `big5euc` in the `ttyadm` command:

   ```
   # pmadm -a -p zsmon -s ttya -i root -fu -v ver -m "'ttyadm -S y \
     -T terminal_type -d /dev/ttya -l 9600 -m big5euc,ldterm,ttcompat -s \
     /usr/bin/login'"
   ```

4. **Turn on the terminal.**

   Follow the documentation that accompanies the terminal.

5. **Log in the terminal.**

6. **Check the correctness of the installation:**

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

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

Setting a User's TTY

To verify that your TTY is properly set up:

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

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

2. **If the values from above (**cs8, -istrip**) are not listed, then use the following command to set them:**

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

   This is the last step in setting up a terminal.

Using Big5 TTY With EUC Locale

1. **If you are using a Big5 type terminal, you must load the STREAMS module into the kernel by using the following command as a superuser:**

   ```
   system% su
   Password: (Type superuser password here if required.)
   # modload /kernel/strmod/big5euc
   ```

2. **Type the following command:**

   ```
   system% setterm -x big5
   ```

Index