## Contents

**Preface** 15

1. **Solaris Internationalization Overview** 19
   - New Internationalization and Localization Features in Solaris 8 19
   - Internationalization and Localization Defined 20
     - Basic Steps in Internationalization 20
   - What Is a Locale? 21
     - Full and Partial Locales 22
   - Cultural Conventions 22
     - Locale Categories 23
   - Using Locale Categories for Localization 24
     - Time Formats 24
     - Date Formats 25
     - Numbers 25
     - Currency 26
     - Language Word and Letter Differences 28
   - Keyboard Differences 30
   - Other Differences 30
     - Paper Sizes 30

*Creating Worldwide Software: The Book* 31
2. **Internationalization Framework in the Solaris 8 Environment** 33
   - Support for Codeset Independence 33
   - The CSI Approach 34
   - CSI-enabled Commands 34
   - Solaris 8 CSI-enabled Libraries 36
   - Locale Database 37
   - Process Code Format 37
   - Multibyte Support Environment (MSE) 37
   - Dynamically Linked Applications 38
   - libw and libintl 39
   - ctype Macros 40
   - Internationalization APIs in libc 41
   - genmsg Utility 47

3. **Contents of Solaris 8 Products** 49
   - Overview of the Solaris 8 Locales 49
   - Summary of the Solaris 8 Locale 50
     - Localization Content on Solaris 8 CD-ROMs 50
     - Localization Functions in Solaris Interfaces 51
     - Script Enabling for Solaris 8 52
     - Localization in the Base and Multilingual Solaris Product 53
   - European Localization 61
     - Multiple Key Compose Sequences for Locales 62
     - Keyboard Support in the Solaris 8 Product 62
     - Changing Between Keyboards on SPARC 63
     - Changing Between Keyboards on IA 63
   - Codesets for IA 64
   - Font Formats 66
   - Summary of Asian Locales 67
How to Use the \texttt{en\_US.UTF-8} Locale Environment 108

TTY Environment Setup 108

Code Conversions 113

Printing 113

DtMail 115

Programming Environment 117

FontSet Used with X Applications 117

XmFontList Definition as CDE/Motif Applications 119

5. **X/DPS** 121

Localization Resource Category 122

Information on Language Interpreters 122

6. **Desktop Environments** 123

Overview of CDE 123

Setting Locales 124

Integrating Fonts 124

Internationalization and CDE 125

Matching Fonts to Character Sets 125

Storage of Localized Text 125

Xlib Dependencies 125

Message Guidelines 126

Internationalization and Distributed Networks 126

Mail Interchange 126

OpenWindows 127

7. **Complex Text Layout** 129

Overview of CTL Technology 129

Overview of CTL Architecture 130

Changes in Motif to Support CTL Technology 130

XmDirection 130
Layout Direction 131
For More Information 131
XmStringDirection 131
XmRendition 132
132
Additional Layout Behavior 133
XmText, XmTextField 134
Character Orientation Action Routines 135
Character Orientation Additional Behavior 135
XmText Action Routines 136
XmTextFieldGetLayoutModifier 144
Purpose 144
Synopsis 144
Description 144
Return Value 144
Related Information 145
XmTextGetLayoutModifier 145
Purpose 145
Synopsis 145
Description 145
Return Value 145
Related Information 145
XmTextFieldSetLayoutModifier 145
Purpose 145
Synopsis 146
Description 146
Related Information 146
XmTextSetLayoutModifier 146

Contents 7
B. Partial L10N Package Names on OS CD 181
C. Languages CD Packages List 187
Index 213
Tables
# Figures

<table>
<thead>
<tr>
<th>Figure 3–1</th>
<th>Functions and Structure of Locales in Solaris</th>
<th>51</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 4–1</td>
<td>Cyrillic Keyboard</td>
<td>98</td>
</tr>
<tr>
<td>Figure 4–2</td>
<td>Greek Euro Keyboard</td>
<td>99</td>
</tr>
<tr>
<td>Figure 4–3</td>
<td>Greek UNIX Keyboard</td>
<td>99</td>
</tr>
<tr>
<td>Figure 4–4</td>
<td>Arabic Keyboard</td>
<td>100</td>
</tr>
<tr>
<td>Figure 4–5</td>
<td>Hebrew Keyboard</td>
<td>101</td>
</tr>
<tr>
<td>Figure 4–6</td>
<td>Thai Keyboard</td>
<td>102</td>
</tr>
<tr>
<td>Figure 7–1</td>
<td>Layout Direction</td>
<td>149</td>
</tr>
<tr>
<td>Figure 7–2</td>
<td>Tabbing Behavior</td>
<td>153</td>
</tr>
</tbody>
</table>
Preface

The International Language Environments Guide describes internationalization features that are new in the Solaris™ 8 operating environment. It contains important information on how to use this release to build global software products that support various languages and cultural conventions.

Specifically, this guide contains:

- Guidelines and tips for developers on how to use this release to write applications for international markets.
- An overall view of internationalization topics that apply to various layers within the Solaris operating environment.
- Pointers to more detailed documentation.

Where appropriate, this guide points you to other guides in the documentation set that contain additional or more detailed information on internationalization features in this release.

Who Should Use This Guide

This guide is intended for software developers and administrators who want to design global products and applications for the Solaris 8 operating environment.

This guide assumes knowledge of the C programming language.

All operating system information pertains to the Solaris 8 SunOS™ 5.8 operating environment.
How this Guide is Organized

The chapters in this guide are organized as follows:

- Chapter 1 tells what’s new and provides an overview of the localized products available on the base (English) and the localized multi-lingual Solaris releases.
- Chapter 2 describes the internationalization framework in the Solaris 8 product.
- Chapter 3 describes the contents of the Solaris 8 localized product.
- Chapter 4 covers the en_US.UTF-8 locales and the internationalization features incorporated into this release.
- Chapter 5 contains a detailed look at the procedures to write a localized version of codesets, formats, collation, and messaging.
- Chapter 6 explains the Solaris desktop environments: the Common Desktop Environment (CDE) and OpenWindows™. The section on CDE has an overview of the application internationalization process, including locale management, localized resources, and font management.
- Chapter 7 includes information about CTL extensions that enable Motif APIs to support writing systems that require complex transformation between logical and physical text representations, such as Arabic, Hebrew, and Thai.
- Chapter 8 explains printing support under the Solaris 8 operating environment, with specific information for European and Asian printing.
- Appendix A contains lists of tables of available iconv Conversions between UTF-8 and UTF-EBCDIC
- Appendix B contains a table of the partial L10N package Names on the OS CD.
- Appendix C contains tables representing the language packages on the language CD. There are tables for Simplified Chinese, French, German, Italian, Japanese, Korean, Spanish, Swedish, Traditional Chinese, and Shared.

Related Books and Sites

For information about the Java Development Kit, see
http://java.sun.com/docs/books/tutorial/i18n/index.html

Common Desktop Environment: Internationalization Programmer's Guide. Mountain View, California, SunSoft Press, 1996. The CDE documentation set can be ordered by title through SunExpress. The CDE Programmer's Guide is also part of the CDE Developer's AnswerBook™ set that is shipped on the Solaris documentation CD.


Ordering Sun Documents

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at http://www1.fatbrain.com/documentation/sun.

Accessing Sun Documentation Online

The docs.sun.com™ Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is http://docs.sun.com.
What Typographic Conventions Mean

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

### TABLE P-1  Typographic Conventions

<table>
<thead>
<tr>
<th>Typeface or Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AaBbCc123</td>
<td>The names of commands, files, and directories; on-screen computer output</td>
<td>Edit your .login file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code></td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>What you type, contrasted with on-screen computer output</td>
<td><code>machine_name% su</code> Password:</td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>Command-line placeholder: replace with a real name or value</td>
<td>To delete a file, type <code>rm filename</code>.</td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>Book titles, new words, or terms, or words to be emphasized.</td>
<td>Read Chapter 6 in <em>User’s Guide</em>. You must be <em>not</em> to do this.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>C shell prompt</td>
<td><code>machine_name%</code></td>
</tr>
<tr>
<td>C shell superuser prompt</td>
<td><code>machine_name#</code></td>
</tr>
<tr>
<td>Bourne shell and Korn shell prompt</td>
<td><code>$</code></td>
</tr>
<tr>
<td>Bourne shell and Korn shell superuser prompt</td>
<td><code>#</code></td>
</tr>
</tbody>
</table>
Solaris Internationalization Overview

The Solaris 8 product includes full Unicode 3.0 support, as defined in ISO-10646, for selected locales. The Solaris 8 release is a major release for Sun’s international markets. It includes a number of new features. All partial locales including multibyte locales such as Japanese locales are now available on the Base Solaris 8 product.

New Internationalization and Localization Features in Solaris 8

- Simplified Chinese UTF-8 locale. This provides broader support for Unicode with the addition of new UTF-8 locales. Unicode is often used in a mixed script environment, where it is necessary to display text from multiple languages in a single environment.
- Traditional Chinese UTF-8 locale
- Asian printing enhancements
- Support for 90 locales on the base Solaris CD. This is a new packaging approach to universal language coverage.
- Enhanced Sdtudctool — support for migration of UDC (User Defined Character) from Microsoft Windows. Localized for all Asian locales.
- Three additional locales have been added for Iceland (ISO8859–1), U.S.A. (ISO8859–15), and Russia (ANSI1251). The new U.S.A. locale adds support for the euro currency glyph. The new Russian locale is in addition to the existing ISO8859–5 and KOI8–R locales. It provides native Microsoft data encoding support. The new ISO8859–1 locale for Iceland marks the introduction of Icelandic support to the Solaris environment.
- Customer-extensible codeset conversion. New codeset conversion can be added by using the *geniconvtbl* utility. Existing codeset conversions can be modified.
- European locale repackaging
- Euro font
- Adding Japanese *iconv* modules — conversions for IBM mainframe codesets and conversions between Unicode and Shift-JIS for Microsoft codeset.
- Euro currency. All foreign exchange, banking, and finance industries in the European community are converting from using their local currencies to using the Euro. Euro currency support has been enhanced in the Solaris 8 environment with the addition of U.S. and Estonian ISO8859–15 locales.
- Multibyte Partial locale — framework of multibyte locale support is included in the Base Solaris product.
- Enhanced Unicode *iconv* modules. The *iconv* module has been enhanced for various Unicode encoding formats and international and de facto industry standard codesets.

---

**Internationalization and Localization Defined**

Internationalization is the process of making software portable between languages or regions, while localization is the process of adapting software for specific languages or regions. International software can be developed using interfaces that modify program behavior at runtime in accordance with specific cultural requirements. Localization involves establishing online information to support a language or region, called a *locale*.

Unlike software that must be completely rewritten before it can work with different native languages and customs, internationalized software does not require rewriting. It can be ported from one locale to another without change. The Solaris system is internationalized, providing the infrastructure and interfaces you need to create internationalized software.

Internationalization and localization are different procedures.

Internationalization is the process of making software that is independent of any locale. It can then be adapted to specific locales.

**Basic Steps in Internationalization**

An internationalized application’s executable image is portable between languages and regions. To internationalize software, you should:
Use the interfaces described in this book to create software with an environment that can be modified dynamically without the necessity of recompiling the software.

- Divide software into executable and messages. The messages include all printable and displayable messages that the user sees. Keep the message strings in a message catalog.

Message strings are translated for a language and a region. A locale includes the message strings and methods to specify sorting.

Locales are not the same as a language. A language can contain various regions. For example, French is spoken in France and Canada, but each country has different ways of displaying monetary and time information.

To use a localized version of a product, the user sets the environment variables. The product then displays the user messages in their translated form. Date, time, currency and other information is formatted and displayed according to locale-specific conventions.

What Is a Locale?

A locale can be composed of both a base language, the country (territory) of use, and possibly codeset (which is usually assumed). For example, German is de, an abbreviation for Deutsch, while Swiss German is de_CH, CH being an abbreviation for Confederation Helvetica. This allows for specific differences by country, such as currency units notation.

More than one locale can be associated with a particular language, which allows for regional differences. For example, an English-speaking user in the United States can select the en_US locale (English for the United States), while an English-speaking user in Great Britain can select en_GB (English for Great Britain).

The key concept for application programs is that of a program’s locale. The locale is an explicit model and definition of a native-language environment. The notion of a locale is explicitly defined and included in the library definitions of the ANSI C Language standard.

The locale consists of a number of categories for which there is country-dependent formatting or other specifications. A program’s locale defines its codesets, date and time formatting conventions, monetary conventions, decimal formatting conventions, and collation (sort) order.

Generally the locale name is specified by the LANG environment variable. Locale categories are subordinate to LANG, but can be set separately, in which case they override LANG. If LC_ALL is set, it overrides not only LANG, but all the separate locale categories as well.
Full and Partial Locales

A full Solaris locale has all of the listed functions and the localized system messages in the relevant language. If no localized messages are installed, then all locales would be classified as “partial locales”. Several locales in the Solaris environment are capable of displaying localized messages, provided that the relevant language support is installed. For example, there are several locales which can use German messages:

- de_DE.ISO8859–1
- de_DE.ISO8859–15
- de_DE.UTF-8
- de_AT.ISO8859–1
- de_AT.ISO8859–15
- de_CH.ISO8859–1

When the German messages translations are installed using the Language CD, all of the above locales will become “full locales”, because they will have access to a fully translated desktop. The language CD contains message translations for the following languages:

- German
- French
- Spanish
- Swedish
- Italian
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

All partial locales are also available in the base product, but message translations are available only in the multilingual Solaris product.

Cultural Conventions

Different cultures use different conventions for writing the date, the time, numbers, currency, delimiting words and phrases, and quoting material.

A locale defines the behavior of a program at runtime according to a language or cultural region’s conventions. Throughout the system, a locale determines the behavior of the following:
Encoding and processing of text data
- Identifying the language and encoding of resource files and their text values
- Rendering and layout of text strings
- Interchanging text that is used for interclient text communication
- Encoding and decoding for interclient text communication
- Selecting the input method (that is, which codeset is generated) and the processing of text data
- Font and icon files that are culturally specific
- Actions and file types
- User Interface Definition (UID) files
- Date and time formats
- Numeric formats
- Monetary formats
- Collation order
- Format for informative and diagnostic messages and interactive responses

The Solaris environment separates language and culture-dependent information from the application and saves it outside the application.

By separating the language and culture-dependent information from the application, the developer does not need to translate, rewrite, or recompile the application for each market. The only requirement to enter a new market is to localize the external information to the local language and customs.

Locale Categories

The locale categories are as follows:

- **LC_CTYPE**
  Controls the behavior of character handling functions

- **LC_TIME**
  Specifies date and time formats, including month names, days of the week, and common full and abbreviated representations

- **LC_MONETARY**
  Specifies monetary formats. Few SunOS system commands or library routines actually use this category

- **LC_NUMERIC**
  Specifies the decimal separator (or radix character) and the thousands separator

- **LC_COLLATE**
  Specifies the sorting order for a locale and the string conversions required to attain this ordering

- **LC_MESSAGES**
Specifies the language in which the localized messages are written

**LO_LTYPE**

Specifies the layout engine that provides information about language rendering. Language rendering (or text rendering) consists of text shaping and directionality.

---

**Using Locale Categories for Localization**

The localization of a product should be done in consultation with native users in that target language or region. Certain styles and information styles and formats might seem perfectly obvious and universal to the developer, but to the user, these look either awkward, wrong, or even offensive. The following pages describe the elements that the Solaris operating environment allows you to control and specify so that you can successfully internationalize your product.

**Time Formats**

Table 1–1 shows some of the ways to write 11:59 P.M.

<table>
<thead>
<tr>
<th>Locale</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canadian</td>
<td>23:59</td>
</tr>
<tr>
<td>Finnish</td>
<td>23.59</td>
</tr>
<tr>
<td>German</td>
<td>23.59 Uhr</td>
</tr>
<tr>
<td>Norwegian</td>
<td>Kl 23.59</td>
</tr>
<tr>
<td>Thai</td>
<td>11:59 PM</td>
</tr>
<tr>
<td>U.K.</td>
<td>11.59 PM</td>
</tr>
</tbody>
</table>

Time is represented by both a 12-hour clock and a 24-hour clock. The hour and minute separator can be either a colon (:) or a period (.)

Time zone splits occur between and within countries. Although a time zone can be described in terms of how many hours it is ahead of, or behind, Greenwich Mean Time (GMT), this number is not always an integer. For example, Nekeybfoundland is in a time zone that is half an hour different from the adjacent time zone.

Daylight Savings Time (DST) starts and ends on different dates that can vary from country to country.
Date Formats

Table 1–2 shows some of the date formats used around the world. Notice that even within a country, there can be variations.

<table>
<thead>
<tr>
<th>Locale</th>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canadian (English and French)</td>
<td>yyyy-mm-dd</td>
<td>1998-08-13</td>
</tr>
<tr>
<td>Danish</td>
<td>yyyy-mm-dd</td>
<td>1999–08–24</td>
</tr>
<tr>
<td>Finnish</td>
<td>dd.mm.yyyy</td>
<td>13.08.1998</td>
</tr>
<tr>
<td>French</td>
<td>dd/mm/yyyy</td>
<td>13/08/1999</td>
</tr>
<tr>
<td>German</td>
<td>yyyy-mm-dd</td>
<td>1999–09–18</td>
</tr>
<tr>
<td>Italian</td>
<td>dd.mm.yy</td>
<td>13.08.98</td>
</tr>
<tr>
<td>Norwegian</td>
<td>dd.mm.yy</td>
<td>13.08.98</td>
</tr>
<tr>
<td>Spanish</td>
<td>dd-mm-yy</td>
<td>13-08-98</td>
</tr>
<tr>
<td>Swedish</td>
<td>yyyy-mm-dd</td>
<td>1998-08-13</td>
</tr>
<tr>
<td>GB-English</td>
<td>dd/mm/yy</td>
<td>13/08/98</td>
</tr>
<tr>
<td>US-English</td>
<td>mm-dd-yy</td>
<td>08-13-98</td>
</tr>
<tr>
<td>Thai</td>
<td>dd/mm/yyyy</td>
<td>10/12/2009</td>
</tr>
</tbody>
</table>

Numbers

Decimal and Thousands Separators

Great Britain and the United States are two of the few places in the world that use a period to indicate the decimal place. Many other countries use a comma instead. The decimal separator is also called the radix character. Likewise, while the U.K. and U.S. use a comma to separate groups of thousands, many other countries use a period instead, and some countries separate thousands groups with a thin space. Table 1–3 shows some commonly used numeric formats.
### Table 1–3

<table>
<thead>
<tr>
<th>Locale</th>
<th>Large Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canadian (English and French)</td>
<td>4 294 967 295,000</td>
</tr>
<tr>
<td>Danish</td>
<td>4 294 967 295,000</td>
</tr>
<tr>
<td>Finnish</td>
<td>4 294 967 295,000</td>
</tr>
<tr>
<td>French</td>
<td>4 294 967 295,000</td>
</tr>
<tr>
<td>German</td>
<td>4 294 967 295,000</td>
</tr>
<tr>
<td>Italian</td>
<td>4,294,967,295,000</td>
</tr>
<tr>
<td>Norwegian</td>
<td>4,294,967,295,000</td>
</tr>
<tr>
<td>Spanish</td>
<td>4,294,967,295,000</td>
</tr>
<tr>
<td>Swedish</td>
<td>4 294 967 295,000</td>
</tr>
<tr>
<td>GB-English</td>
<td>4,294,967,295.00</td>
</tr>
<tr>
<td>US-English</td>
<td>4,294,967,295.00</td>
</tr>
<tr>
<td>Thai</td>
<td>4,294,967,295.00</td>
</tr>
</tbody>
</table>

Data files containing locale-specific formats are frequently misinterpreted when transferred to a system in a different locale. For example, a file containing numbers in a French format is not useful to a U.K.-specific program.

### List Separators

There are no particular locale conventions that specify how to separate numbers in a list. They are sometimes comma-delimited in Great Britain and the U.S., but often spaces and semicolons are used.

### Currency

Currency units and presentation order vary greatly around the world. Table 1–4 shows monetary formats in some countries.
<table>
<thead>
<tr>
<th>Locale</th>
<th>Currency</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canadian (English)</td>
<td>Dollar ($)</td>
<td>$1,234.56</td>
</tr>
<tr>
<td>Canadian (French)</td>
<td>Dollar ($)</td>
<td>1,234.56$</td>
</tr>
<tr>
<td>Danish</td>
<td>Kroner (kr)</td>
<td>kr 1,234,56</td>
</tr>
<tr>
<td>Finnish</td>
<td>Markka (mk)</td>
<td>1,234,56 mk</td>
</tr>
<tr>
<td>French</td>
<td>Franc (F)</td>
<td>1 234,56 F</td>
</tr>
<tr>
<td>German</td>
<td>Deutsche Mark (DM)</td>
<td>DM 1,234,56</td>
</tr>
<tr>
<td>Italian</td>
<td>Lira (L)</td>
<td>L1,234,56</td>
</tr>
<tr>
<td>Japanese</td>
<td>Yen</td>
<td>41,234 Yen</td>
</tr>
<tr>
<td>Norwegian</td>
<td>Krone (kr)</td>
<td>kr 1,234,56</td>
</tr>
<tr>
<td>Spanish</td>
<td>Peseta (Pts)</td>
<td>1,234,56 Pts</td>
</tr>
<tr>
<td>Swedish</td>
<td>Krona (Kr)</td>
<td>1,234,56 kr</td>
</tr>
<tr>
<td>GB-English</td>
<td>Pound</td>
<td>31,234.56 pounds</td>
</tr>
<tr>
<td>US-English</td>
<td>Dollar ($)</td>
<td>$1,234.56</td>
</tr>
<tr>
<td>Thai</td>
<td>Baht</td>
<td>2539 Baht</td>
</tr>
<tr>
<td>Euro</td>
<td>EUR</td>
<td>€400,00</td>
</tr>
</tbody>
</table>

Local and international symbols for currency can differ. For example, the designation for the French franc is “F” in France but this is often written as “FRF” internationally to distinguish it from other francs, such as the Swiss franc or the Polynesian franc.

Euro locales are based on the ISO8859–15 character set. See “European Localization” on page 61 for available locales.

Be aware also that a converted currency amount can take up more or less space than the original amount. To illustrate: $1,000 can become L1,307.000.
Language Word and Letter Differences

Word Delimiters

In English, words are separated by a space character. In languages such as Chinese, Japanese and Thai, however, there is often no delimiter between words.

Sort Order

Sorting order for particular characters is not the same in all languages. For example, the character “ö” sorts with the ordinary “o” in Germany, but sorts separately in Sweden, where it is the last letter of the alphabet. In some languages, characters have weight to determine the priority of the character sequences. For example, in Thai, the Thai dictionary defines sorting through the sequences of characters that have different weights.

Character Sets

Number of Characters

While the English alphabet contains only 26 characters, some languages contain many more characters. Japanese, for example, can contain over 40,000 characters, Chinese even more.

Western European Alphabets

The alphabets of most western European countries are similar to the standard 26-character alphabet used in English-speaking countries, but there are often some additional basic characters, some marked (or accented) characters, and some ligatures.

Japanese Text

Japanese text is composed of three different scripts mixed together: Kanji ideographs derived from Chinese, and two phonetic scripts (or syllabaries), Hiragana and Katakana.

Although each character in Hiragana has an equivalent in Katakana, Hiragana is the most common script, with cursive rather than block-like letter forms. Kanji characters are used to write root words. Katakana is mostly used to represent “foreign” words—words “imported” from languages other than Japanese.

Kanji has tens of thousands of characters, but the number commonly used has been declining steadily over the years. Now only about 3500 are frequently used, although the average Japanese writer has a vocabulary of about 2000 Kanji characters. Nonetheless, computer systems must support more than 7000 because that is what the Japan Industry Standard (JIS) requires. In addition, there are about 170 Hiragana and
Katakana characters. On average 55% of Japanese text is Hiragana, 35% Kanji, and 10% Katakana. Arabic numerals and Roman letters are also present in Japanese text.

Although it is possible to completely avoid the use of Kanji, most Japanese readers find text containing Kanji easier to understand.

**Korean Text**

Korean text can be written using a phonetic writing system called Hangul. Hangul has more than 11,000 characters, which consist of 19 consonants, 21 vowels, and an optional 27 consonants. About 3,000 Hangul characters from the entire Hangul characters are usually used in Korean computer systems. Korean also uses ideographs based on the set invented in China, called Hanja. Korean text requires over 6,000 Hanja characters. Hanja is used mostly to avoid confusion when Hangul would be ambiguous. Hangul characters are formed by combining consonants and vowels. After combining them, they can compose one syllable, which is a Hangul character. Hangul characters are often arranged in a square, so that the group takes up the same space as a Hanja character. Arabic numerals, Roman letters, and special symbol characters are also present in Korean text.

**Thai Text**

A Thai character can be defined as a column position on a display screen with four display cells. Each column position can have up to three characters. The composition of a display cell is based on the Thai character’s classification. Some Thai characters can be composed with another character’s classification. If they can be composed together, both characters are in the same cell. Otherwise, they are in separate cells.

**Chinese Text**

Chinese usually consists entirely of characters from the ideographic script called Hanzi. In the People’s Republic of China (PRC) there are about 7000 commonly used Hanzi characters in GB2312 (zh locale) and more than 20,000 characters in the GBK (zh.GBK) locale. In Taiwan, current standards require more than 13000 characters; 6000 others have been recently standardized but are considered rare.

If a character is not a root character, it usually consists of two or more parts, two being most common. In two-part characters, one part generally represents meaning, and the other represents pronunciation. Occasionally both parts represent meaning. The radical is the most important element, and characters are traditionally arranged by radical, of which there are several hundred. A single sound can be represented by many different characters, which are not interchangeable in usage. A single character can have different sounds.

Some characters are more appropriate than others in a given context—the appropriate one is distinguished phonetically by the use of tones. By contrast, spoken Japanese and Korean lack tones.
Several phonetic systems represent Chinese. In the People’s Republic of China the most common is pinyin, which uses roman characters and is widely employed in the West for place names such as Beijing. The Wade-Giles system is an older phonetic system, formerly used for place names such as Peking. In Taiwan zhuyin (or bopomofo), a phonetic alphabet with unique letter forms, is often used instead.

Commercial applications, particularly those that deal with people’s names, need to consider the impact of codeset expansion. Many Chinese people have names containing characters that do not exist in any standard codeset. You need to provide space in unassigned codesets to deal with this issue.

Keyboard Differences

Not all characters on the U.S. keyboard appear on other keyboards. Similarly, other keyboards often contain many characters not visible on the U.S. keyboard.

However, on SPARC machines, the Compose key can be used to produce any character in the ISO Latin-1 codeset on any keyboard that supports it.

The Compose key can be used with English or European locales, but not with Korean, Chinese, or Japanese locales, except the UTF-8 locales.

Other Differences

Paper Sizes

Within each country a small number of paper sizes are commonly used, normally with one of those sizes being much more common than the others. Most countries follow ISO Standard 216: “Writing paper and certain classes of printed matter—Trimmed sizes—A and B series.”

Internationalized applications should not make assumptions about the page sizes available to them. The Solaris system provides no support for tracking output page size; this is the responsibility of the application program. Table 1–5 shows Common International page sizes.
### TABLE 1–5  Common International Page Sizes

<table>
<thead>
<tr>
<th>Paper Type</th>
<th>Dimensions</th>
<th>Countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO A4</td>
<td>21.0 cm by 29.7 cm</td>
<td>Everywhere except U.S.</td>
</tr>
<tr>
<td>ISO A5</td>
<td>14.8 cm by 21.0 cm</td>
<td>Everywhere except U.S.</td>
</tr>
<tr>
<td>JIS B4</td>
<td>25.9 cm by 36.65 cm</td>
<td>Japan</td>
</tr>
<tr>
<td>JIS B5</td>
<td>18.36 cm by 25.9 cm</td>
<td>Japan</td>
</tr>
<tr>
<td>U.S. Letter</td>
<td>8.5 inch by 11 inches</td>
<td>U.S. and Canada</td>
</tr>
<tr>
<td>U.S. Legal</td>
<td>8.5 inch by 14 inches</td>
<td>U.S. and Canada</td>
</tr>
</tbody>
</table>

---

**Creating Worldwide Software: The Book**

The book *Creating Worldwide Software*, 2nd edition, by Bill Tuthill and David Smallberg (SunSoft Press, 1997), is a guide to localizing for the Solaris platform. The book is recommended for developers who work with the Solaris system. See “Related Books and Sites” on page 16 for a full citation.
Internationalization Framework in the Solaris 8 Environment

This section discusses several internationalization features contained in the Solaris 8 environment.

- Support for Codeset independence
- Locale database
- Process code format (wide character expression)
- libw and libint1
- ctype macros
- genmsg utility

This section also contains information useful for developing internationalized applications such as:

- Dynamically linked applications
- Solaris 8 internationalized APIs

Support for Codeset Independence

The Solaris 8 operating environment supports non-EUC encodings such as PC-Kanji in Japan, Big-5 in Taiwan, and GBK in the People’s Republic of China.

Because a large part of the computer market demands non-EUC codeset support, Solaris 8 provides a solid framework to enable both EUC and non-EUC codeset support. This support is called Codeset Independence, or CSI.
The goal of CSI is to remove EUC dependencies on specific codesets or encoding methods from Solaris OS libraries and commands. The CSI architecture allows the Solaris operating environment to support any UNIX file system safe encoding. CSI supports a number of new codesets, such as UTF-8, PC-Kanji, and Big-5.

The CSI Approach

Codeset Independence enables application and platform software developers to keep their code independent of encoding, such as UTF-8, and also provides the ability to adopt any new encoding without having to modify the source code. This architecture approach differs from Java internationalization in that Java requires applications to be Unicode-dependent and also requires code conversions throughout the application.

Many existing internationalized applications (for example, Motif) automatically inherit CSI support from the underlying system. These applications work in the new locales without modification. OPEN LOOK applications, however, that are XView /OLIT based, don’t work in the new locales because XView is codeset-dependent.

CSI is inherently independent from any codesets. However, the following assumptions on file code encodings (codesets) still apply to Solaris 8:

- File code is a superset of ASCII.
- Unicode (16-bits fixed width) cannot be supported as file code.
- NULL (0x00) is not part of multibyte characters for support of null-terminated multibyte character strings.
- Slash / (0x2f) is not part of multibyte characters for support of the UNIX path names.
- Only stateless file code encodings are supported.

CSI-enabled Commands

Table 2–1 contains CSI-enabled commands in Solaris 8. These commands are marked with CSI capabilities on their man page.

All commands are in the /usr/bin directory, unless otherwise noted.

<table>
<thead>
<tr>
<th>TABLE 2–1</th>
<th>CSI-enabled Commands in Solaris 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lib/diffh</td>
<td>acctcom</td>
</tr>
<tr>
<td>Command Path</td>
<td>Command 1</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>/usr/sbin/accept</td>
<td>apropos</td>
</tr>
<tr>
<td>/usr/sbin/reject</td>
<td>batch</td>
</tr>
<tr>
<td>/usr/ucb/lpr</td>
<td>bdiff</td>
</tr>
<tr>
<td>/usr/xpg4/bin/awk</td>
<td>cancel</td>
</tr>
<tr>
<td>/usr/xpg4/bin/cp</td>
<td>cat</td>
</tr>
<tr>
<td>/usr/xpg4/bin/date</td>
<td>catman</td>
</tr>
<tr>
<td>/usr/xpg4/bin/dup</td>
<td>chgrp</td>
</tr>
<tr>
<td>/usr/xpg4/bin/ed</td>
<td>chmod</td>
</tr>
<tr>
<td>/usr/xpg4/bin/edit</td>
<td>chown</td>
</tr>
<tr>
<td>/usr/xpg4/bin/egrep</td>
<td>cmp</td>
</tr>
<tr>
<td>/usr/xpg4/bin/env</td>
<td>col</td>
</tr>
<tr>
<td>/usr/xpg4/bin/ex</td>
<td>comm</td>
</tr>
<tr>
<td>/usr/xpg4/bin/expr</td>
<td>compress</td>
</tr>
<tr>
<td>/usr/xpg4/bin/fgrep</td>
<td>cpio</td>
</tr>
<tr>
<td>/usr/xpg4/bin/grep</td>
<td>csh</td>
</tr>
<tr>
<td>/usr/xpg4/bin/ln</td>
<td>csplit</td>
</tr>
<tr>
<td>/usr/xpg4/bin/ls</td>
<td>cut</td>
</tr>
<tr>
<td>/usr/xpg4/bin/more</td>
<td>diff</td>
</tr>
<tr>
<td>/usr/xpg4/bin/mv</td>
<td>diff3</td>
</tr>
<tr>
<td>/usr/xpg4/bin/nice</td>
<td>disable</td>
</tr>
<tr>
<td>/usr/xpg4/bin/nohup</td>
<td>echo</td>
</tr>
<tr>
<td>/usr/xpg4/bin/od</td>
<td>expand</td>
</tr>
<tr>
<td>/usr/xpg4/bin/pr</td>
<td>file</td>
</tr>
<tr>
<td>/usr/xpg4/bin/rm</td>
<td>fine</td>
</tr>
<tr>
<td>/usr/xpg4/bin/sed</td>
<td>fold</td>
</tr>
<tr>
<td>/usr/xpg4/bin/sort</td>
<td>ftp</td>
</tr>
<tr>
<td>/usr/xpg4/bin/tail</td>
<td></td>
</tr>
<tr>
<td>/usr/xpg4/bin/tr</td>
<td></td>
</tr>
</tbody>
</table>

Internationalization Framework in the Solaris 8 Environment 35
Solaris 8 CSI-enabled Libraries

Nearly all functions in Solaris 8 libc (/usr/lib/libc.so) are CSI-enabled. However, the following functions in libc are not CSI-enabled because they are EUC-dependent functions:

- csetcol()
- csetlen()
- euccol()

- euclen()
- eucscol()
- getwidth()

The following macros are not CSI-enabled because they are EUC dependent:

- csetno()
- wcsetno()

In the Solaris 8 product, libgen (/usr/ccs/lib/libgen.a) are internationalized, but not CSI enabled.

In the Solaris 8 product, libcurses (/usr/ccs/lib/libcurses.a) are internationalized, but not CSI enabled.

Here are the five deliverables:

- The utility (32-bit application):
  /usr/bin/geniconvtbl

- special iconv shared objects:
  /usr/lib/iconv/geniconvtbl.so
  /usr/lib/iconv/sparcv9/geniconvtbl.so

- Sample geniconvtbl(1) input source files and system-provided binary table files:
  /usr/lib/iconv/geniconvtbl/srcs/
  ISO8859-1_to_ISO646.txt
  ISO646_to_ISO8859-1.txt
  ISO8859-1_to_UTF-8.txt
  UTF-8_to_ISO8859-1.txt
  ShiftJIS_to_eucJP.txt
  eucJP_to_ShiftJIS.txt
Locale Database

The locale database format and structure is private and subject to change in a future release. Therefore, when developing an internationalized application, do not directly access the locale database. Instead, use the Solaris internationalization APIs.

When using Solaris 8, use the locale databases that are included with the Solaris 8 product. Do not use locales from previous Solaris versions.

Process Code Format

The process code format in the Solaris 8 product is private and subject to change in a future release. Therefore, when developing an international application, do not assume the process code format is the same. Instead use the Solaris internationalization APIs.

Multibyte Support Environment (MSE)

A multibyte character is a character that cannot be stored in a single byte, such as Chinese, Japanese, or Korean characters. These characters require two or three bytes of...
storage. A more precise definition can be found in ISO/IEC 9899:1990 subclause 3.13. The programming model enables these multibyte characters to be read in as logical units and stored internally as wide characters. These wide characters can be processed by the program as logical entities in their own right. Finally, these wide characters can be written out (undergoing appropriate translation) as logical units. This is analogous to the way single-byte characters are read in, manipulated, and written out again. The MSE provides a comparable set of interfaces to perform this processing. The MSE allows programs to be written to handle multibyte characters using the same programming model that is used for single-byte characters.

Dynamically Linked Applications

Solaris 8 users can choose how to link applications with the system libraries, such as libc, by using dynamic linking or static linking. However, any application that requires internationalization features in the system libraries must be dynamically linked. If the application has been statically linked, the operation to set the locale to other than C and POSIX using the setlocale function will fail. Statically linked applications can be operated only in C and POSIX locales.

By default, the linker program tries to link the application dynamically. If the command line options to the linker and the compiler include -Bstatic or -dn specifications, your application might be statically linked. You can check whether an existing application is dynamically linked using the /usr/bin/ldd command.

For example, if you type:

```
% /usr/bin/ldd /sbin/sh
```

the command displays the following message:

```
% ldd: /sbin/sh: file is not a dynamic executable or shared object
```

The message indicates the /sbin/sh command is not a dynamically linked program. Also, if you type:

```
% /usr/bin/ldd /usr/bin/ls
```

the command displays the following message:

```
% libc.so.1 => /usr/lib/libc.so.1
% libdl.so.1 => /usr/lib/libdl.so.1
```
This message indicates the `/usr/bin/ls` command has been dynamically linked with two libraries, `libc.so.1` and `libdl.so.1`.

To summarize, if the message from the `ldd` command to the application does not contain a `libc.so.1` entry, it indicates that the application has been statically linked with `libc`. In that case, you need to change the command line options to the linker so that dynamic linking is used instead, then re-link the application.

**libw and libintl**

These interfaces have moved to `libc` and are no longer in `libw` and `libintl`.

The shared objects ensure runtime compatibility for existing applications and, together with the archives, provide compilation environment compatibility for building applications. However, it is no longer necessary to build applications against `libw` or `libintl`.

For more information on filters see the *Linker and Libraries Guide*.

Table 2–2 shows the stub entry points in `libw` and `libintl`. 
### TABLE 2-2  Stub Entry Points in libw and libintl

<table>
<thead>
<tr>
<th>libw:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>fgetwc</td>
<td>fgetws</td>
<td>fputwc</td>
<td>fputws</td>
<td>getwc</td>
<td></td>
</tr>
<tr>
<td>getwchar</td>
<td>getws</td>
<td>isenglish</td>
<td>isideograms</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>isphonogram</td>
<td>isspecial</td>
<td>iswalnum</td>
<td>iswalnum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iswctype</td>
<td>iswdigit</td>
<td>iswgraph</td>
<td>iswlower</td>
<td>iswprint</td>
<td></td>
</tr>
<tr>
<td>iswpunct</td>
<td>iswspace</td>
<td>iswupper</td>
<td>iswxdigit</td>
<td>putwc</td>
<td></td>
</tr>
<tr>
<td>putwchar</td>
<td>putws</td>
<td>strtos</td>
<td>towlower</td>
<td>towupper</td>
<td></td>
</tr>
<tr>
<td>ungetwc</td>
<td>watoll</td>
<td>wcscat</td>
<td>wcshcr</td>
<td>wcscmp</td>
<td></td>
</tr>
<tr>
<td>wcscoll</td>
<td>wcscpy</td>
<td>wcscspn</td>
<td>wcsftime</td>
<td>wcslen</td>
<td></td>
</tr>
<tr>
<td>wcscat</td>
<td>wcscmp</td>
<td>wcscpy</td>
<td>wcspbrk</td>
<td>wcsrchr</td>
<td></td>
</tr>
<tr>
<td>wcsspn</td>
<td>wcstod</td>
<td>wcstok</td>
<td>wcstol</td>
<td>wcstoul</td>
<td></td>
</tr>
<tr>
<td>wcswcs</td>
<td>wcswidth</td>
<td>wcswxfrm</td>
<td>wcctype</td>
<td>wcwidth</td>
<td></td>
</tr>
<tr>
<td>wscasecmp</td>
<td>wcscat</td>
<td>wcshcr</td>
<td>wcscmp</td>
<td>wcscol</td>
<td></td>
</tr>
<tr>
<td>wscoll</td>
<td>wscpy</td>
<td>wcscspn</td>
<td>wsdup</td>
<td>wslen</td>
<td></td>
</tr>
<tr>
<td>wscasecmp</td>
<td>wsncat</td>
<td>wsncmp</td>
<td>wsncpy</td>
<td>wspbrk</td>
<td></td>
</tr>
<tr>
<td>wsscanf</td>
<td>wsrchr</td>
<td>wsscanf</td>
<td>wsscanf</td>
<td>wstod</td>
<td></td>
</tr>
<tr>
<td>wstok</td>
<td>wstol</td>
<td>wstoll</td>
<td>wstosstr</td>
<td>wcxfm</td>
<td></td>
</tr>
<tr>
<td>libintl:</td>
<td>bindtextdomain</td>
<td>dcgettext</td>
<td>dgettext</td>
<td>gettext</td>
<td>textdomain</td>
</tr>
</tbody>
</table>

## ctype Macros

Character classification and character transformation macros are defined in /usr/include/ctype.h. The Solaris 8 environment provides a new set of `ctype` macros. The new macros support character classification and transformation semantics defined by XPG4. To access the new set of macros, one of the following conditions must be met:

- `_XPG4_CHAR_CLASS` is defined.
- `_XOPEN_SOURCE` and `_XOPEN_VERSION=4` are defined.
- `_XOPEN_SOURCE` and `_XOPEN_SOURCE_EXTENDED=1` are defined.
This means that all XPG4 and XPG4.2 applications automatically have the new macros. Since \_XOPEN\_SOURCE, \_XOPEN\_VERSION, and \_XOPEN\_SOURCE\_EXTENDED bring in extra XPG4 related features in addition to new ctype macros, non-XPG4 or XPG4.2 applications should use \__XPG4\_CHAR\_CLASS\__. There are corresponding ctype functions. The Solaris 8 functions also support XPG4 semantics. Refer to the ctype(3C) man page for details.

---

### Internationalization APIs in libc

Solaris 8 offers two sets of APIs:
- Multibyte (file codes)
- Wide characters (process code)

Applications process in wide-character codes.

When a program takes input from a file, convert your file’s multibyte data into wide character process code with the mtbwc and mtbowcs APIs. To convert the file output data from wide character format into multibyte format, use the wcstombs and wctomb APIs.

Table 2-3 shows a list of internationalization APIs included in Solaris 8.

<table>
<thead>
<tr>
<th>API Type</th>
<th>Library Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Messaging functions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>catclose()</td>
<td></td>
<td>Close a message catalog.</td>
</tr>
<tr>
<td>catgets()</td>
<td></td>
<td>Read a program message.</td>
</tr>
<tr>
<td>catopen()</td>
<td></td>
<td>Open a message catalog.</td>
</tr>
<tr>
<td>dgettext()</td>
<td></td>
<td>Get a message from a message catalog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with domain specified.</td>
</tr>
<tr>
<td>dcgettext()</td>
<td></td>
<td>Get a message from a message catalog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with domain and category specified.</td>
</tr>
<tr>
<td>textdomain()</td>
<td></td>
<td>Set and query the current domain.</td>
</tr>
<tr>
<td>bindtextdomain()</td>
<td></td>
<td>Bind the path for a message domain.</td>
</tr>
</tbody>
</table>

---

Internationalization Framework in the Solaris 8 Environment
<table>
<thead>
<tr>
<th>API Type</th>
<th>Library Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iconv()</td>
<td></td>
<td>Convert codes.</td>
</tr>
<tr>
<td>iconv_close()</td>
<td></td>
<td>Deallocate the conversion descriptor.</td>
</tr>
<tr>
<td>iconv_open()</td>
<td></td>
<td>Allocate the conversion descriptor.</td>
</tr>
<tr>
<td>Regular expression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>regcomp()</td>
<td></td>
<td>Compile the regular expression.</td>
</tr>
<tr>
<td>regexec()</td>
<td></td>
<td>Execute regular expression matching.</td>
</tr>
<tr>
<td>regerror()</td>
<td></td>
<td>Provide a mapping from error codes to error message.</td>
</tr>
<tr>
<td>regfree()</td>
<td></td>
<td>Free memory allocated by regcomp().</td>
</tr>
<tr>
<td>Wide character class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wctype()</td>
<td></td>
<td>Define character class.</td>
</tr>
<tr>
<td>wctrans</td>
<td></td>
<td>Define character mapping.</td>
</tr>
<tr>
<td>towctrans</td>
<td></td>
<td>Wide-character mapping.</td>
</tr>
<tr>
<td>setlocale()</td>
<td></td>
<td>Modify and query a program’s locale.</td>
</tr>
<tr>
<td>nl_langinfo()</td>
<td></td>
<td>Get language and cultural information of current locale.</td>
</tr>
<tr>
<td>localeconv()</td>
<td></td>
<td>Get monetary and numeric formatting information of current locale.</td>
</tr>
<tr>
<td>Character classification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>isalpha()</td>
<td></td>
<td>Is character alphabetic?</td>
</tr>
<tr>
<td>isupper()</td>
<td></td>
<td>Is character uppercase?</td>
</tr>
<tr>
<td>islower()</td>
<td></td>
<td>Is character lowercase?</td>
</tr>
<tr>
<td>isdigit()</td>
<td></td>
<td>Is character a digit?</td>
</tr>
<tr>
<td>isxdigit()</td>
<td></td>
<td>Is character a hex digit?</td>
</tr>
<tr>
<td>isalnum()</td>
<td></td>
<td>Is character alphabetic or digital?</td>
</tr>
<tr>
<td>isspace()</td>
<td></td>
<td>Is character a space?</td>
</tr>
<tr>
<td>API Type</td>
<td>Library Routine</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ispunct()</td>
<td></td>
<td>Is character a punctuation mark?</td>
</tr>
<tr>
<td>isprint()</td>
<td></td>
<td>Is character printable?</td>
</tr>
<tr>
<td>iscntrl()</td>
<td></td>
<td>Is character a control character?</td>
</tr>
<tr>
<td>isascii()</td>
<td></td>
<td>Is character an ASCII character?</td>
</tr>
<tr>
<td>isgraph()</td>
<td></td>
<td>Is character a visible character?</td>
</tr>
<tr>
<td>isphonogram()</td>
<td></td>
<td>Is wide character a phonogram?</td>
</tr>
<tr>
<td>isideogram()</td>
<td></td>
<td>Is wide character an ideogram?</td>
</tr>
<tr>
<td>isenglish()</td>
<td></td>
<td>Is wide character in English alphabet from a supplementary codeset?</td>
</tr>
<tr>
<td>isnumber()</td>
<td></td>
<td>Is wide character a digit from a supplementary codeset?</td>
</tr>
<tr>
<td>isspecial()</td>
<td></td>
<td>Is special wide character from a supplementary codeset?</td>
</tr>
<tr>
<td>iswalpha()</td>
<td></td>
<td>Is wide character alphabetic?</td>
</tr>
<tr>
<td>iswupper()</td>
<td></td>
<td>Is wide character uppercase?</td>
</tr>
<tr>
<td>iswlower()</td>
<td></td>
<td>Is wide character lowercase?</td>
</tr>
<tr>
<td>iswdigit()</td>
<td></td>
<td>Is wide character a digit?</td>
</tr>
<tr>
<td>iswxdigit()</td>
<td></td>
<td>Is wide character a hex digit?</td>
</tr>
<tr>
<td>iswalpha()</td>
<td></td>
<td>Is wide character an alphabetic character or digit?</td>
</tr>
<tr>
<td>iswspace()</td>
<td></td>
<td>Is wide character a white space?</td>
</tr>
<tr>
<td>iswpunct()</td>
<td></td>
<td>Is wide character a punctuation mark?</td>
</tr>
<tr>
<td>iswprint()</td>
<td></td>
<td>Is wide character a printable character?</td>
</tr>
<tr>
<td>iswgraph()</td>
<td></td>
<td>Is wide character a visible character?</td>
</tr>
<tr>
<td>iswcntrl()</td>
<td></td>
<td>Is wide character a control character?</td>
</tr>
<tr>
<td>iswascii()</td>
<td></td>
<td>Is wide character an ASCII character?</td>
</tr>
<tr>
<td>toupper()</td>
<td></td>
<td>Convert a lowercase character to uppercase.</td>
</tr>
<tr>
<td>tolower()</td>
<td></td>
<td>Convert an uppercase character to lowercase.</td>
</tr>
<tr>
<td>API Type</td>
<td>Library Routine</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>towupper()</td>
<td>Convert a lowercase wide character to uppercase.</td>
</tr>
<tr>
<td></td>
<td>tolower()</td>
<td>Convert an uppercase wide character to lowercase.</td>
</tr>
<tr>
<td>Character collation</td>
<td>strcoll()</td>
<td>Collate character strings.</td>
</tr>
<tr>
<td></td>
<td>strxfrm()</td>
<td>Transform character strings for comparison.</td>
</tr>
<tr>
<td></td>
<td>wcscoll()</td>
<td>Collate wide character strings.</td>
</tr>
<tr>
<td></td>
<td>wcsxfrm()</td>
<td>Transform wide character strings for comparison.</td>
</tr>
<tr>
<td>Monetary handling</td>
<td>strfmon()</td>
<td>Convert monetary value to string representation.</td>
</tr>
<tr>
<td>Date and time handling</td>
<td>getdate()</td>
<td>Convert user format date and time.</td>
</tr>
<tr>
<td></td>
<td>strftime()</td>
<td>Convert date and time to string representation. The %u conversion function conforms to the X/Open CAE Specification, System Interfaces and Headers, Issue 4, Version 2. This function represents a weekday as a decimal number [1,7], with 1 now representing Monday.</td>
</tr>
<tr>
<td></td>
<td>strptime()</td>
<td>Date and time conversion.</td>
</tr>
<tr>
<td>Multibyte handling</td>
<td>btowc</td>
<td>Single-byte to wide-character conversion.</td>
</tr>
<tr>
<td></td>
<td>mbrlen()</td>
<td>Get number of bytes in character (restartable).</td>
</tr>
<tr>
<td></td>
<td>mbsinit()</td>
<td>Determine conversion object status.</td>
</tr>
<tr>
<td></td>
<td>mbtowc()</td>
<td>Convert a character to a wide-character code (restartable).</td>
</tr>
<tr>
<td></td>
<td>mbstowcs()</td>
<td>Convert a character string to a wide-character string (restartable).</td>
</tr>
</tbody>
</table>

Wide characters
<table>
<thead>
<tr>
<th>API Type</th>
<th>Library Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wcscat()</td>
<td>Concatenate wide-character strings to length ( n ).</td>
</tr>
<tr>
<td></td>
<td>wsdup()</td>
<td>Duplicate wide-character string.</td>
</tr>
<tr>
<td></td>
<td>wcscmp()</td>
<td>Compare wide-character strings.</td>
</tr>
<tr>
<td></td>
<td>wcsncmp()</td>
<td>Compare wide-character strings to length ( n ).</td>
</tr>
<tr>
<td></td>
<td>wcscpy()</td>
<td>Copy wide-character strings.</td>
</tr>
<tr>
<td></td>
<td>wcsncpy()</td>
<td>Copy wide-character strings to length ( n ).</td>
</tr>
<tr>
<td></td>
<td>wcschr()</td>
<td>Find character in wide-character string.</td>
</tr>
<tr>
<td></td>
<td>wcsrchr()</td>
<td>Find character in wide-character string from right.</td>
</tr>
<tr>
<td></td>
<td>wcslen()</td>
<td>Get length of wide-character string.</td>
</tr>
<tr>
<td></td>
<td>wscoll()</td>
<td>Return display width of wide-character string.</td>
</tr>
<tr>
<td></td>
<td>wcspn()</td>
<td>Return span of one wide-character string in another.</td>
</tr>
<tr>
<td></td>
<td>wcspn()</td>
<td>Return span of one wide-character string not in another.</td>
</tr>
<tr>
<td></td>
<td>wcspbrk()</td>
<td>Return pointer to one wide-character string in another.</td>
</tr>
<tr>
<td></td>
<td>wcstok()</td>
<td>Move token through wide-character string.</td>
</tr>
<tr>
<td></td>
<td>wcswcs()</td>
<td>Find string in wide-character string.</td>
</tr>
<tr>
<td></td>
<td>wcstombs()</td>
<td>Convert wide-character string to multibyte string.</td>
</tr>
<tr>
<td></td>
<td>wctomb()</td>
<td>Convert wide-character to multibyte character.</td>
</tr>
<tr>
<td></td>
<td>wcwidth()</td>
<td>Determine number of column positions of a wide character.</td>
</tr>
<tr>
<td></td>
<td>wcswidth()</td>
<td>Determine number of column positions of a wide-character string.</td>
</tr>
<tr>
<td></td>
<td>wctob</td>
<td>Wide-character to single-byte conversion.</td>
</tr>
<tr>
<td></td>
<td>wcrtomb</td>
<td>Convert a wide-character code to a character (restartable).</td>
</tr>
<tr>
<td>API Type</td>
<td>Library Routine</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>wcsrtombs</td>
<td>Interpret wide-character string according to format.</td>
</tr>
<tr>
<td>Wide formatting</td>
<td>wsprintf()</td>
<td>Generate wide-character string according to format.</td>
</tr>
<tr>
<td></td>
<td>wsscanf()</td>
<td>Formatted input conversion.</td>
</tr>
<tr>
<td></td>
<td>fprintf()</td>
<td>Print formatted wide-character output.</td>
</tr>
<tr>
<td></td>
<td>fscanf()</td>
<td>Convert formatted wide-character input.</td>
</tr>
<tr>
<td></td>
<td>wprintf()</td>
<td>Print formatted wide-character output.</td>
</tr>
<tr>
<td></td>
<td>wcscanf</td>
<td>Convert formatted wide-character input.</td>
</tr>
<tr>
<td></td>
<td>swprintf()</td>
<td>Print formatted wide-character output.</td>
</tr>
<tr>
<td></td>
<td>swscanf()</td>
<td>Convert formatted wide-character input.</td>
</tr>
<tr>
<td></td>
<td>vfwprintf()</td>
<td>Wide-character formatted output of a va_list.</td>
</tr>
<tr>
<td></td>
<td>vswprintf()</td>
<td>Wide-character formatted output of a va_list.</td>
</tr>
<tr>
<td>Wide numbers</td>
<td>wcstol()</td>
<td>Convert wide-character string to long integer.</td>
</tr>
<tr>
<td></td>
<td>wcstoul()</td>
<td>Convert wide-character string to unsigned long integer.</td>
</tr>
<tr>
<td></td>
<td>wcstod()</td>
<td>Convert wide-character string to double precision.</td>
</tr>
<tr>
<td>Wide strings</td>
<td>wscasecmp()</td>
<td>Compare wide-character strings, ignore case differences.</td>
</tr>
<tr>
<td></td>
<td>wsncasecmp()</td>
<td>Process code-string operations.</td>
</tr>
<tr>
<td></td>
<td>wcsstrstr()</td>
<td>Find a wide-character substring.</td>
</tr>
<tr>
<td></td>
<td>wmemchr</td>
<td>Find a wide-character in memory.</td>
</tr>
<tr>
<td></td>
<td>wmemcmp</td>
<td>Compare wide-characters in memory.</td>
</tr>
<tr>
<td></td>
<td>wmemcpy</td>
<td>Copy wide-characters in memory.</td>
</tr>
</tbody>
</table>
### TABLE 2–3  Internationalization APIs in libc  (continued)

<table>
<thead>
<tr>
<th>API Type</th>
<th>Library Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wmemmove</td>
<td>Copy wide-characters in memory with overlapping areas.</td>
</tr>
<tr>
<td></td>
<td>wmemset</td>
<td>Set wide-characters in memory.</td>
</tr>
<tr>
<td>Wide standard I/O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>fgetwc()</td>
<td>Get multibyte character from stream, convert to wide character.</td>
</tr>
<tr>
<td></td>
<td>getwchar()</td>
<td>Get multibyte character from stdin, convert to wide character.</td>
</tr>
<tr>
<td></td>
<td>fgetws()</td>
<td>Get multibyte string from stream, convert to wide character.</td>
</tr>
<tr>
<td></td>
<td>getws()</td>
<td>Get multibyte string from stdin, convert to wide character.</td>
</tr>
<tr>
<td></td>
<td>fputwc()</td>
<td>Convert wide character to multibyte character, puts to stream.</td>
</tr>
<tr>
<td></td>
<td>fwrite</td>
<td>Set stream orientation.</td>
</tr>
<tr>
<td></td>
<td>putwchar()</td>
<td>Convert wide character to multibyte character, puts to stdin.</td>
</tr>
<tr>
<td></td>
<td>fputws()</td>
<td>Convert wide character to multibyte string, puts to stream.</td>
</tr>
<tr>
<td></td>
<td>putws()</td>
<td>Convert wide character to multibyte string, puts to stdin.</td>
</tr>
<tr>
<td></td>
<td>ungetwc()</td>
<td>Push a wide character back into input stream.</td>
</tr>
</tbody>
</table>

---

**genmsg Utility**

The new *genmsg* utility can be used with the *catgets()* family of functions to create internationalized source message catalogs. The utility examines a source program file for calls to functions in *catgets* and builds a source message catalog from the information it finds. For example:

(continued)
In the above example, `genmsg` is run on the source file `example.c`, which produces a source message catalog named `example.c.msg`. The `-c` option with the argument `NOTE` causes `genmsg` to include comments in the catalog. If a comment in the source program contains the string specified, the comment appears in the message catalog after the next string extracted from a call to `catgets()`.

You can use `genmsg` to number the messages in a message set automatically.

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

The material in this section is used with permission from *Creating Worldwide Software: Solaris International Developer’s Guide*, 2nd edition by Bill Tuthill and David A. Smallberg, published by Sun Microsystems Press/Prentice Hall. (c)1997 Sun Microsystems, Inc.
Overview of the Solaris 8 Locales

Multiple environments exist within the Solaris operating system for support of different national languages. Each of these national environments is called a locale, which considers the language, its characters, fonts, and the customs used to input and format data.

A locale defines the behavior of a program at run time according to the language and cultural conventions of a user’s geographical area. Throughout the system, locales affect the following:

- Encoding and processing of text data
- Identifying the language and encoding of resource files and their text values
- Rendering and layout of text strings
- Interchanging text that is used for interclient text communication
- Encoding and decoding for interclient text communication
- Selecting the input method (that is, which codeset is generated) and the processing of text data
- Font and icon files that are culturally specific
- Actions and file types
- User Interface Definition (UID) files
- Date and time formats
- Numeric formats
- Monetary formats
- Collation order
Summary of the Solaris 8 Locale

All Solaris 8 locale packages are classified into two categories. The first category is for partial locales, which are the enablers of the locales. With partial locales installed on the system, users can run applications on the target locales, while the OS/GUI messages from Solaris are English. All partial locale packages are available on the Solaris OS CDs.

The second category is for full locale packages. These packages include translations of software messages, on-line help files, optional fonts, and language specific features. Full locale packages provide the full set of language features to 9 languages.

- German
- French
- Spanish
- Swedish
- Italian
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

Full locale packages are available on the languages CD. Partial locale packages (locale enablers) have to be installed in order for the full locales to be functional.

Localization Content on Solaris 8 CD-ROMs

Partial locales are selected at the beginning of the install procedure on the OS CD-ROM. Full locales are automatically installed from the Language CD-ROM according to the locale selections made at the beginning of the install procedure.

The distribution of locales is shown in the table below.
### Table 3–1 Solaris 8 Installation CD-ROMs

<table>
<thead>
<tr>
<th>Disk</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris OS CD-ROM</td>
<td>Solaris 8 Operating System</td>
</tr>
<tr>
<td></td>
<td>all partial locales</td>
</tr>
<tr>
<td>Language CD-ROM</td>
<td>message translations for 9 languages</td>
</tr>
<tr>
<td></td>
<td>locale specific utilities</td>
</tr>
</tbody>
</table>

As mentioned, the locales include partial locales. These are based on core locales for the main language. For example, the `fr_CA.ISO8859-1` (French Canadian) is based on the `fr_FR.ISO8859-1` (French) locale. These partial locales utilize the messages that are delivered into its parent locale (French for `fr_CA`). If a locale hasn’t been fully localized, then it might contain only English messages.

### Localization Functions in Solaris Interfaces

The OS locale layer provides the basic locale database and functions that are plugged into the OS system interface at the application’s run time. Applications will access these OS locale modules through standard APIs as described in Chapter 2.

The X11 locale layer provides the interface to X input method and X output method such that the X11 applications can allow local text input and display. Fonts are provided to allow applications to display characters from various languages.

CDE/Motif is built on top of the X11 window system. Hence, it can utilize the X11 locale capability through X11 APIs. Solaris localizations have various locale-specific configurations for CDE applications, in order to make the desktop functional within the target locale.

Message translations and on-line help contents are provided throughout different layers as described in the following diagram.

*Figure 3–1 Tabbing Behavior*
Script Enabling for Solaris 8

The Solaris 8 base product provides multiple levels of script enabling, such as simple ASCII support, Latin/European support, Asian multibyte support, and Arabic/Hebrew bidirectional support.
The interfaces defined within the X/Open specification are capable of supporting a large set of languages and territories, including the following types of script:

<table>
<thead>
<tr>
<th>Script</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latin Language</td>
<td>Americas, Eastern/Western Europe, Turkey</td>
</tr>
<tr>
<td>Greek</td>
<td>Greece</td>
</tr>
<tr>
<td>East Asia</td>
<td>Japanese, Korean, and Chinese</td>
</tr>
<tr>
<td>Indic</td>
<td>Thai</td>
</tr>
<tr>
<td>Bidirectional</td>
<td>Arabic and Hebrew</td>
</tr>
<tr>
<td>Cyrillic</td>
<td>Russian</td>
</tr>
</tbody>
</table>

Localization in the Base and Multilingual Solaris Product

The base Solaris 8 product includes all partial locales, (including multibyte locales) which provide the functionality needed to input, display, and print text in their target languages while using English user interfaces.

The multilingual Solaris 8 product is a super set of the base Solaris product. It additionally includes 9 language translations (user interface and documentation) and some additional software such as BCP support, optional fonts, and optional utilities on the Language CD.

The English Unicode locale (en_US.UTF-8) is installed as the default, while other locales are installed when the locale is selected as install locale during the Solaris install process. Since the UTF-8 locales require all the languages fonts, basic fonts supporting all languages are also installed as the default.

The File System Safe Universal Transformation Format, or UTF-8, is an encoding defined by X/Open as a multi-byte representation of Unicode. UTF-8 encompasses almost all of the characters for traditional single-byte and multi-byte locales for European and Asian languages for Solaris locales.

Additional locale support is packaged according to the geographic region which they support. During the Solaris install process, you are prompted to choose which geographic regions require your support. The locale support available after installation has finished depends on the choices made at this stage.

The following tables lists all the locales supported by the Solaris 8 environment. The locale names have been updated from the Solaris 7 environment in keeping with international naming standards.

All of these locales are also present in the base Solaris 8 release.
<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ja</td>
<td>Japanese</td>
<td>Japan</td>
<td>eucJP</td>
<td>Japanese (EUC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JISX0201-1976</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JISX0208-1990</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JISX0212-1990</td>
</tr>
<tr>
<td>ja_JP.PCK</td>
<td>Japanese</td>
<td>Japan</td>
<td>PCK</td>
<td>Japanese (PC kanji)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JISX0201-1976</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JISX0208-1990</td>
</tr>
<tr>
<td>ja_JP.UTF-8</td>
<td>Japanese</td>
<td>Japan</td>
<td>UTF-8</td>
<td>Japanese (UTF-8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unicode 3.0</td>
</tr>
<tr>
<td>ko</td>
<td>Korean</td>
<td>Korea</td>
<td>5601</td>
<td>Korean (EUC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KSC 5601-1987</td>
</tr>
<tr>
<td>ko.UTF-8</td>
<td>Korean</td>
<td>Korea</td>
<td>UTF-8</td>
<td>Korean (UTF-8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unicode 3.0</td>
</tr>
<tr>
<td>th</td>
<td>English</td>
<td>Thailand</td>
<td>TIS620.2533</td>
<td>Thai TIS620.2533</td>
</tr>
<tr>
<td>zh</td>
<td>Simplified Chinese</td>
<td>PRC</td>
<td>gb2312</td>
<td>Simplified Chinese (EUC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GB2312-1980</td>
</tr>
<tr>
<td>zh.GBK</td>
<td>Simplified Chinese</td>
<td>PRC</td>
<td>GBK</td>
<td>Simplified Chinese (GBK) GBK</td>
</tr>
<tr>
<td>zh.UTF-8</td>
<td>Simplified Chinese</td>
<td>PRC</td>
<td>UTF-8</td>
<td>Simplified Chinese (UTF-8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unicode 3.0</td>
</tr>
<tr>
<td>zh_TW</td>
<td>Traditional Chinese</td>
<td>Taiwan</td>
<td>cns11643</td>
<td>Traditional Chinese (EUC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CNS 11643-1992</td>
</tr>
</tbody>
</table>
### TABLE 3–2  Asia (continued)

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>zh_TW.BIG5</td>
<td>Traditional Chinese</td>
<td>Taiwan</td>
<td>BIG5</td>
<td>Traditional Chinese (BIG5) BIG5</td>
</tr>
<tr>
<td>zh_TW.UTF-8</td>
<td>Traditional Chinese</td>
<td>Taiwan</td>
<td>UTF-8</td>
<td>Traditional Chinese (UTF-8) Unicode 3.0</td>
</tr>
</tbody>
</table>

### TABLE 3–3  Australasia

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>en_AU.ISO8859-1</td>
<td>English</td>
<td>Australia</td>
<td>ISO8859-1</td>
<td>English (Australia)</td>
</tr>
<tr>
<td>en_NZ.ISO8859-1</td>
<td>English</td>
<td>New Zealand</td>
<td>ISO8859-1</td>
<td>English (New Zealand)</td>
</tr>
</tbody>
</table>

### TABLE 3–4  Central America

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>es_CR.ISO8859-1</td>
<td>Spanish</td>
<td>Costa Rica</td>
<td>ISO8859-1</td>
<td>Spanish (Costa Rica)</td>
</tr>
<tr>
<td>es_GT.ISO8859-1</td>
<td>Spanish</td>
<td>Guatemala</td>
<td>ISO8859-1</td>
<td>Spanish (Guatemala)</td>
</tr>
<tr>
<td>es_MX.ISO8859-1</td>
<td>Spanish</td>
<td>Mexico</td>
<td>ISO8859-1</td>
<td>Spanish (Mexico)</td>
</tr>
<tr>
<td>es_NI.ISO8859-1</td>
<td>Spanish</td>
<td>Nicaragua</td>
<td>ISO8859-1</td>
<td>Spanish (Nicaragua)</td>
</tr>
<tr>
<td>es_PA.ISO8859-1</td>
<td>Spanish</td>
<td>Panama</td>
<td>ISO8859-1</td>
<td>Spanish (Panama)</td>
</tr>
<tr>
<td>es SV.ISO8859-1</td>
<td>Spanish</td>
<td>El Salvador</td>
<td>ISO8859-1</td>
<td>Spanish (El Salvador)</td>
</tr>
</tbody>
</table>
### TABLE 3–5  Central Europe

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs_CZ.ISO8859-2</td>
<td>English</td>
<td>Czech Republic</td>
<td>ISO8859-2</td>
<td>Czech (Czech Republic)</td>
</tr>
<tr>
<td>de_AT.ISO8859-1</td>
<td>German</td>
<td>Austria</td>
<td>ISO8859-1</td>
<td>German (Austria)</td>
</tr>
<tr>
<td>de_AT.ISO8859-15</td>
<td>German</td>
<td>Austria</td>
<td>ISO8859-15</td>
<td>German (Austria, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>de_CH.ISO8859-1</td>
<td>German</td>
<td>Switzerland</td>
<td>ISO8859-1</td>
<td>German (Switzerland)</td>
</tr>
<tr>
<td>de_DE.UTF-8</td>
<td>German</td>
<td>Germany</td>
<td>UTF-8</td>
<td>German (Germany, Unicode 3.0)</td>
</tr>
<tr>
<td>de_DE.ISO8859-1</td>
<td>German</td>
<td>Germany</td>
<td>ISO8859-1</td>
<td>German (Germany)</td>
</tr>
<tr>
<td>de_DE.ISO8859-15</td>
<td>German</td>
<td>Germany</td>
<td>ISO8859-15</td>
<td>German (Germany, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>fr_CH.ISO8859-1</td>
<td>French</td>
<td>Switzerland</td>
<td>ISO8859-1</td>
<td>German (Switzerland)</td>
</tr>
<tr>
<td>hu_HU.ISO8859-2</td>
<td>English</td>
<td>Hungary</td>
<td>ISO8859-2</td>
<td>Hungarian (Hungary)</td>
</tr>
<tr>
<td>pl_PL.ISO8859-2</td>
<td>English</td>
<td>Poland</td>
<td>ISO8859-2</td>
<td>Polish (Poland)</td>
</tr>
<tr>
<td>sk_SK.ISO8859-2</td>
<td>English</td>
<td>Slovakia</td>
<td>ISO8859-2</td>
<td>Slovak (Slovakia)</td>
</tr>
</tbody>
</table>

### TABLE 3–6  Eastern Europe

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>bg_BG.ISO8859-5</td>
<td>English</td>
<td>Bulgaria</td>
<td>ISO8859-5</td>
<td>Bulgarian (Bulgaria)</td>
</tr>
<tr>
<td>et_EE.ISO8859-15</td>
<td>English</td>
<td>Estonia</td>
<td>ISO8859-15</td>
<td>Estonian (Estonia)</td>
</tr>
<tr>
<td>hr_HR.ISO8859-2</td>
<td>English</td>
<td>Croatia</td>
<td>ISO8859-2</td>
<td>Croatian (Croatia)</td>
</tr>
<tr>
<td>lt_LT.ISO8859-13</td>
<td>English</td>
<td>Lithuania</td>
<td>ISO8859-13</td>
<td>Lithuanian (Lithuania)</td>
</tr>
<tr>
<td>lv_LV.ISO8859-13</td>
<td>English</td>
<td>Latvia</td>
<td>ISO8859-13</td>
<td>Latvian (Latvia)</td>
</tr>
<tr>
<td>mk_MK.ISO8859-5</td>
<td>English</td>
<td>Macedonia</td>
<td>ISO8859-5</td>
<td>Macedonian (Macedonia)</td>
</tr>
<tr>
<td>ro_RO.ISO8859-2</td>
<td>English</td>
<td>Romania</td>
<td>ISO8859-2</td>
<td>Romanian (Romania)</td>
</tr>
<tr>
<td>Locale</td>
<td>User Interface</td>
<td>Territory</td>
<td>Codeset</td>
<td>Language Support</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
<td>-----------</td>
<td>-----------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>ru_RU.KOI8-R</td>
<td>English</td>
<td>Russia</td>
<td>KOI8-R</td>
<td>Russian (Russia, KOI8-R)</td>
</tr>
<tr>
<td>ru_RU.ANSI1251</td>
<td>English</td>
<td>Russia</td>
<td>ansi-1251</td>
<td>Russian (Russia, ANSI 1251)</td>
</tr>
<tr>
<td>ru_RU.ISO8859-5</td>
<td>English</td>
<td>Russia</td>
<td>ISO8859-5</td>
<td>Russia (Russia)</td>
</tr>
<tr>
<td>sh_BA.ISO8859-2 @bosnia</td>
<td>English</td>
<td>Bosnia</td>
<td>ISO8859-2</td>
<td>Bosnian (Bosnia)</td>
</tr>
<tr>
<td>sl_SI.ISO8859-2</td>
<td>English</td>
<td>Slovenia</td>
<td>ISO8859-2</td>
<td>Slovenian (Slovenia)</td>
</tr>
<tr>
<td>sq_AL.ISO8859-2</td>
<td>English</td>
<td>Albania</td>
<td>ISO8859-2</td>
<td>Albanian (Albania)</td>
</tr>
<tr>
<td>sr_YU.ISO8859-5</td>
<td>English</td>
<td>Serbia</td>
<td>ISO8859-5</td>
<td>Serbian (Serbia)</td>
</tr>
<tr>
<td>tr_TR.ISO8859-9</td>
<td>English</td>
<td>Turkey</td>
<td>ISO8859-9</td>
<td>Turkish (Turkey)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>he_IL.ISO8859-6</td>
<td>English</td>
<td>Israel</td>
<td>ISO8859-6</td>
<td>Hebrew (Israel)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar_EY.ISO8859-1</td>
<td>English</td>
<td>Egypt</td>
<td>ISO8859-6</td>
<td>Arabic (Egypt)</td>
</tr>
</tbody>
</table>
### TABLE 3–9  North America

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>en_CA.ISO8859-1</td>
<td>English</td>
<td>Canada</td>
<td>ISO8859-1</td>
<td>English (Canada)</td>
</tr>
<tr>
<td>en_US.ISO8859-1</td>
<td>English</td>
<td>USA</td>
<td>ISO8859-1</td>
<td>English (U.S.A.)</td>
</tr>
<tr>
<td>en_US.UTF-8</td>
<td>English</td>
<td>USA</td>
<td>UTF-8</td>
<td>English (U.S.A., Unicode 3.0)</td>
</tr>
<tr>
<td>fr_CA.ISO8859-1</td>
<td>French</td>
<td>Canada</td>
<td>ISO8859-1</td>
<td>French (Canada)</td>
</tr>
</tbody>
</table>

### TABLE 3–10  North Europe

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>da_DK.ISO8859-1</td>
<td>English</td>
<td>Denmark</td>
<td>ISO8859-1</td>
<td>Danish (Denmark)</td>
</tr>
<tr>
<td>da_DK.ISO8859-15</td>
<td>English</td>
<td>Denmark</td>
<td>ISO8859-15</td>
<td>Danish (Denmark, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>fi_FI.ISO8859-1</td>
<td>English</td>
<td>Finland</td>
<td>ISO8859-1</td>
<td>Finnish (Finland)</td>
</tr>
<tr>
<td>is_IS.ISO8859-1</td>
<td>English</td>
<td>Iceland</td>
<td>ISO8859-1</td>
<td>Icelandic (Iceland)</td>
</tr>
<tr>
<td>no_NO.ISO8859-1@bokm</td>
<td>English</td>
<td>Norway</td>
<td>ISO8859-1</td>
<td>Norwegian (Norway — Bokmal)</td>
</tr>
<tr>
<td>no_NO.ISO8859-1@nyorsk</td>
<td>English</td>
<td>Norway</td>
<td>ISO8859-1</td>
<td>Norwegian (Norway — Nynorsk)</td>
</tr>
<tr>
<td>sv_SE.ISO8859-1</td>
<td>Swedish</td>
<td>Sweden</td>
<td>ISO8859-1</td>
<td>Swedish (Sweden)</td>
</tr>
<tr>
<td>sv_SE.UTF-8</td>
<td>Swedish</td>
<td>Sweden</td>
<td>UTF-8</td>
<td>Swedish (Sweden, Unicode 3.0)</td>
</tr>
</tbody>
</table>
### TABLE 3–11  South America

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>es_Ar.ISO8859-1</td>
<td>Spanish</td>
<td>Argentina</td>
<td>ISO8859-1</td>
<td>Spanish (Argentina)</td>
</tr>
<tr>
<td>es_Bo.ISO8859-1</td>
<td>Spanish</td>
<td>Bolivia</td>
<td>ISO8859-1</td>
<td>Spanish (Bolivia)</td>
</tr>
<tr>
<td>es_Cl.ISO8859-1</td>
<td>Spanish</td>
<td>Chilie</td>
<td>ISO8859-1</td>
<td>Spanish (Chile)</td>
</tr>
<tr>
<td>es_Co.ISO8859-1</td>
<td>Spanish</td>
<td>Colombia</td>
<td>ISO8859-1</td>
<td>Spanish (Colombia)</td>
</tr>
<tr>
<td>es_Ec.ISO8859-1</td>
<td>Spanish</td>
<td>Ecuador</td>
<td>ISO8859-1</td>
<td>Spanish (Ecuador)</td>
</tr>
<tr>
<td>es_Per.ISO8859-1</td>
<td>Spanish</td>
<td>Peru</td>
<td>ISO8859-1</td>
<td>Spanish (Peru)</td>
</tr>
<tr>
<td>es_PY.ISO8859-1</td>
<td>Spanish</td>
<td>Paraguay</td>
<td>ISO8859-1</td>
<td>Spanish (Paraguay)</td>
</tr>
<tr>
<td>es_Uy.ISO8859-1</td>
<td>Spanish</td>
<td>Uruguay</td>
<td>ISO8859-1</td>
<td>Spanish (Uruguay)</td>
</tr>
<tr>
<td>es_Ve.ISO8859-1</td>
<td>Spanish</td>
<td>Venezuela</td>
<td>ISO8859-1</td>
<td>Spanish (Venezuela)</td>
</tr>
<tr>
<td>pt_Br.ISO8859-1</td>
<td>English</td>
<td>Brazil</td>
<td>ISO8859-1</td>
<td>Portuguese (Brazil)</td>
</tr>
</tbody>
</table>

### TABLE 3–12  South Europe

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>el_GR.ISO8859-7</td>
<td>English</td>
<td>Greece</td>
<td>ISO8859-7</td>
<td>Greek (Greece)</td>
</tr>
<tr>
<td>es_ES.ISO8859-1</td>
<td>Spanish</td>
<td>Spain</td>
<td>ISO8859-1</td>
<td>Spanish (Spain)</td>
</tr>
<tr>
<td>es_ES.ISO8859-15</td>
<td>Spanish</td>
<td>Spain</td>
<td>ISO8859-15</td>
<td>Spanish (Spain, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>es_ES.UTF-8</td>
<td>Spanish</td>
<td>Spain</td>
<td>UTF-8</td>
<td>Spanish (Spain, Unicode 3.0)</td>
</tr>
<tr>
<td>it_IT.ISO8859-1</td>
<td>Italian</td>
<td>Italy</td>
<td>ISO8859-1</td>
<td>Italian (Italy)</td>
</tr>
<tr>
<td>it_IT.ISO8859-15</td>
<td>Italian</td>
<td>Italy</td>
<td>ISO8859-15</td>
<td>Italian (Italy, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>it_IT.UTF-8</td>
<td>Italian</td>
<td>Italy</td>
<td>UTF-8</td>
<td>Italian (Italy, Unicode 3.0)</td>
</tr>
<tr>
<td>pt_PT.ISO8859-1</td>
<td>English</td>
<td>Portugal</td>
<td>ISO8859-1</td>
<td>Portuguese (Portugal)</td>
</tr>
<tr>
<td>Locale</td>
<td>User Interface</td>
<td>Territory</td>
<td>Codeset</td>
<td>Language Support</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>---------------</td>
<td>---------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>en_GB.ISO8859-1</td>
<td>English</td>
<td>Great Britain</td>
<td>ISO8859-1</td>
<td>English (Great Britain)</td>
</tr>
<tr>
<td>en_IE.ISO8859-1</td>
<td>English</td>
<td>Ireland</td>
<td>ISO8859-1</td>
<td>English (Ireland)</td>
</tr>
<tr>
<td>fr_BE.ISO8859-1</td>
<td>French</td>
<td>Belgium-Wallon</td>
<td>ISO8859-1</td>
<td>French (Belgium-Wallon)</td>
</tr>
<tr>
<td>fr_FR.ISO8859-1</td>
<td>French</td>
<td>France</td>
<td>ISO8859-1</td>
<td>French (France)</td>
</tr>
<tr>
<td>fr_FR.UTF-8</td>
<td>French</td>
<td>France</td>
<td>UTF-8</td>
<td>French (France, Unicode 3.0)</td>
</tr>
<tr>
<td>nl_BE.ISO8859-1</td>
<td>English</td>
<td>Belgium-Flemish</td>
<td>ISO8859-1</td>
<td>Dutch (Belgium-Flemish)</td>
</tr>
<tr>
<td>nl_NL.ISO8859-1</td>
<td>English</td>
<td>Netherlands</td>
<td>ISO8859-1</td>
<td>Dutch (Netherlands)</td>
</tr>
</tbody>
</table>

Locale naming conventions are as follows:
language[.territory][.codeset]
where language is from ISO639 and territory is from ISO3166.
All Solaris product locales preserve the Portable Character Set characters with US-ASCII code values.
A single locale can have more than one locale name. For example, ja_JP.eucJP is the same as ja. Also, fr_FR.ISO8859-1 is the same as fr.

PCK is also known as Shift JIS (SJIS).

UTF-8 is the UTF-8 of ISO/IEC 10646–1 containing various approved amendments and Unicode 3.0.

GBK signifies GB extensions. This includes all GB 2312–80 characters and all Unified Han characters of ISO/IEC 10646–1, as well as Japanese Hiragana and Katagana characters. It also includes many characters of Chinese, Japanese, and Korean character sets and of ISO/IEC 10646–1.

---

**European Localization**

Solaris 8 software supports the euro currency. Local currency symbols are still available for backward compatibility.

**TABLE 3–14**  User Locales To Support the Euro Currency

<table>
<thead>
<tr>
<th>Region</th>
<th>Locale Name</th>
<th>ISO Codeset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austria</td>
<td>de_AT.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Belgium (French)</td>
<td>fr_BE.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Belgium (Dutch)</td>
<td>nl_BE.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Denmark</td>
<td>da_DK.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Finland</td>
<td>fi_FI.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>France</td>
<td>fr_FR.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Germany</td>
<td>de_DE.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Ireland</td>
<td>en_IE.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Italy</td>
<td>it_IT.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Netherlands</td>
<td>nl_NL.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Portugal</td>
<td>pt_PT.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Region</td>
<td>Locale Name</td>
<td>ISO Codeset</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Spain</td>
<td>es_ES.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Sweden</td>
<td>sv_SE.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Great Britain</td>
<td>en_GB.ISO8859-15</td>
<td>8859-15</td>
</tr>
<tr>
<td>Europe</td>
<td>en_EU</td>
<td>8859-15</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>en_US</td>
<td>8859-15</td>
</tr>
</tbody>
</table>

### Multiple Key Compose Sequences for Locales

The Solaris 8 operating environment supports “Compose Sequences” to create the diacritical marks used in writing the scripts covered in the following codesets:

- ISO 8859-2 (Latin2) Czech, Polish, and Hungarian
- ISO 8859-13 (Latin7) Latvian and Lithuanian
- ISO 8859-9 (Latin5) Turkish

These are the diacritic characters that can be created with the following keys and the Compose key.

- diaeresis = citation (‘’ ) (for example, Compose + A + ‘’ = Å)
- caron = v (for example, Compose + E + v = E caron)
- breve = u
- ogonek = a
- doubleacute = > greater
- degree symbol = O + 0 (o plus zero)
- currency symbol = 0 + x (zero plus x)

### Keyboard Support in the Solaris 8 Product

The following locales have keyboard layouts for SPARC (X-server) and IA (Xserver PLUS console):

- Czech
- Hungary
- Poland
- Latvia
- Lithuania
- Russia
Changing Between Keyboards on SPARC

Support for changing layouts in the Solaris product is achieved only by using the dip-switch settings under the keyboard. The keyboard layout is determined by the dip switches. A list of keyboard layouts and corresponding defined dip-switch settings is at /usr/openwin/share/etc/keytables/keytable.map.

The following is a layout table for a type 4 keyboard (1=switch up, 0=switch down).

<table>
<thead>
<tr>
<th>Dip Switch</th>
<th>Keyboard</th>
<th>Setting in Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>Hungary5.kt</td>
<td>110011</td>
</tr>
<tr>
<td>52</td>
<td>Poland5.kt</td>
<td>110100</td>
</tr>
<tr>
<td>53</td>
<td>Czech5.k</td>
<td>110101</td>
</tr>
<tr>
<td>54</td>
<td>Russia5.kt</td>
<td>110110</td>
</tr>
<tr>
<td>55</td>
<td>Latvia5.k</td>
<td>110111</td>
</tr>
<tr>
<td>56</td>
<td>Turkey5.kt</td>
<td>111000</td>
</tr>
<tr>
<td>57</td>
<td>Greece5.kt</td>
<td>111001</td>
</tr>
<tr>
<td>58</td>
<td>Lithuania5.k</td>
<td>111011</td>
</tr>
</tbody>
</table>

Changing the layout from U.S./GB to Czech is done by changing the dip-switch settings to the setting defined in the file. The file defines the switches in hex. This needs to be converted into binary and then re-booted.

Russian and Greek keyboard support can be toggled on and off using the SPARC Compose key (Ctrl+Shift+F1 on IA).

Changing Between Keyboards on IA

On IA, a keyboard is selected during the kdmconfig part of install. To change this at any time after installation, use kdmconfig:

1. Exit CDE/OW to the command line.
2. Type kdmconfig -u (kdmconfig unconfigure).
3. Type kdmconfig to run the program.
4. Follow instructions to get a keyboard layout.
There are no ‘utilities’ for either SPARC or IA (apart from standard UNIX tools such as xmodmap, pcmapkeys) bundled into Solaris 8 for switching keyboards.

**Codesets for IA**

The default codeset on the Solaris system for IA is ISO-8859-1. The IBM DOS 437 codeset is provided as an option in text mode. That is, if you choose to download IBM DOS 437 codeset by typing:

```
loadfont -c 437
pcmapkeys -f /usr/share/lib/keyboards/437/en_US
```

Nonstandard U.S. date, time, currency, numbers, units, and collation are not supported. Non-English message and text presentation is not supported, nor is multibyte character support. Therefore, non-Microsoft Windows users should use the IBM DOS 437 codeset only in the default C locale.

- You must be in the text mode to download the IBM codeset, not the graphics mode.
- If you are not using the standard U.S. PC keyboard, replace `en_US` with the keyboard map related to your keyboard.
- To download the default codeset in text mode, type:

```
loadfont -c 8859
pcmapkeys -f /usr/share/lib/keyboards/8859/en_US
```

- See the `loadfont` and `pcmapkeys` man pages.

All of the locales support character input and output. There is also `iconv` support for many of the major codesets. (For more on `iconv`, see `iconv(1)`.

**TABLE 3–16  `iconv` Support**

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Target Code</th>
<th>Symbol</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO 859-2</td>
<td>iso2</td>
<td>MS 1250</td>
<td>win2</td>
<td>Windows Latin 2</td>
</tr>
<tr>
<td>ISO 859-2</td>
<td>iso2</td>
<td>MS 852</td>
<td>dos2</td>
<td>MS-DOS Latin 2</td>
</tr>
<tr>
<td>ISO 859-2</td>
<td>iso2</td>
<td>Mazovia</td>
<td>maz</td>
<td>Mazovia</td>
</tr>
<tr>
<td>ISO 859-2</td>
<td>iso2</td>
<td>DHN</td>
<td>dhn</td>
<td>Dom Handlowy Nauki</td>
</tr>
<tr>
<td>MS 1250</td>
<td>win2</td>
<td>ISO 859-2</td>
<td>iso2</td>
<td>ISO Latin 2</td>
</tr>
<tr>
<td>MS 1250</td>
<td>win2</td>
<td>MS 852</td>
<td>dos2</td>
<td>MS-DOS Latin 2</td>
</tr>
<tr>
<td>Code</td>
<td>Symbol</td>
<td>Target Code</td>
<td>Symbol</td>
<td>Language Support</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>-------------</td>
<td>--------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>MS 1250</td>
<td>win2</td>
<td>Mazovia</td>
<td>maz</td>
<td>Mazovia</td>
</tr>
<tr>
<td>MS 1250</td>
<td>win2</td>
<td>DHN</td>
<td>dhn</td>
<td>Dom Handlowy Naduki</td>
</tr>
<tr>
<td>MS 852</td>
<td>dos2</td>
<td>ISO 8859-2</td>
<td>iso2</td>
<td>ISO Latin 2</td>
</tr>
<tr>
<td>MS 852</td>
<td>dos2</td>
<td>MS 1250</td>
<td>win2</td>
<td>Windows Latin 2</td>
</tr>
<tr>
<td>MS 852</td>
<td>dos2</td>
<td>Mazovia</td>
<td>maz</td>
<td>Mazovia</td>
</tr>
<tr>
<td>MS 852</td>
<td>dos2</td>
<td>DHN</td>
<td>dhn</td>
<td>Dom Handlowy Nauki</td>
</tr>
<tr>
<td>Mazovia</td>
<td>maz</td>
<td>ISO 8859-2</td>
<td>iso2</td>
<td>ISO Latin 2</td>
</tr>
<tr>
<td>Mazovia</td>
<td>maz</td>
<td>MS 1250</td>
<td>win2</td>
<td>Windows Latin 2</td>
</tr>
<tr>
<td>Mazovia</td>
<td>maz</td>
<td>MS 852</td>
<td>dos2</td>
<td>MS-DOS Latin 2</td>
</tr>
<tr>
<td>Mazovia</td>
<td>maz</td>
<td>DHN</td>
<td>dhn</td>
<td>Dom Handlowy Nauki</td>
</tr>
<tr>
<td>DHN</td>
<td>dhn</td>
<td>ISO 8859-2</td>
<td>iso2</td>
<td>ISO Latin 2</td>
</tr>
<tr>
<td>DHN</td>
<td>dhn</td>
<td>MS 1250</td>
<td>win2</td>
<td>Windows Latin 2</td>
</tr>
<tr>
<td>DHN</td>
<td>dhn</td>
<td>MS 852</td>
<td>dos2</td>
<td>MS-DOS Latin 2</td>
</tr>
<tr>
<td>DHN</td>
<td>dhn</td>
<td>Mazovia</td>
<td>maz</td>
<td>Mazovia</td>
</tr>
<tr>
<td>ISO 8859-5</td>
<td>iso5</td>
<td>KOI8-R</td>
<td>koi8</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>ISO 8859-5</td>
<td>iso5</td>
<td>PC Cyrillic</td>
<td>alt</td>
<td>Alternative PC Cyrillic</td>
</tr>
<tr>
<td>ISO 8859-5</td>
<td>iso5</td>
<td>MS 1251</td>
<td>win5</td>
<td>Windows Cyrillic</td>
</tr>
<tr>
<td>ISO 8859-5</td>
<td>iso5</td>
<td>Mac Cyrillic</td>
<td>mac</td>
<td>Macintosh Cyrillic</td>
</tr>
<tr>
<td>OKI8-R</td>
<td>koi8</td>
<td>ISO 8859-5</td>
<td>iso5</td>
<td>ISO 8859-5 Cyrillic</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>koi8</td>
<td>PC Cyrillic</td>
<td>alt</td>
<td>Alternative PC Cyrillic</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>koi8</td>
<td>MS 1251</td>
<td>win5</td>
<td>Windows Cyrillic</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>koi8</td>
<td>Mac Cyrillic</td>
<td>mac</td>
<td>Macintosh Cyrillic</td>
</tr>
<tr>
<td>PC Cyrillic</td>
<td>alt</td>
<td>ISO 8859-5</td>
<td>iso5</td>
<td>ISO 8859-5 Cyrillic</td>
</tr>
<tr>
<td>PC Cyrillic</td>
<td>alt</td>
<td>KOI8-R</td>
<td>koi8</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>PC Cyrillic</td>
<td>alt</td>
<td>MS 1251</td>
<td>win5</td>
<td>Windows Cyrillic</td>
</tr>
<tr>
<td>PC Cyrillic</td>
<td>alt</td>
<td>Mac Cyrillic</td>
<td>mac</td>
<td>Macintosh Cyrillic</td>
</tr>
</tbody>
</table>
TABLE 3–16  iconv Support  (continued)

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Target Code</th>
<th>Symbol</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS 1251</td>
<td>win5</td>
<td>ISO 8859-5</td>
<td>iso5</td>
<td>ISO 8859-5 Cyrillic</td>
</tr>
<tr>
<td>MS 1251</td>
<td>win5</td>
<td>KOI8-R</td>
<td>koi8</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>MS 1251</td>
<td>win5</td>
<td>PC Cyrillic</td>
<td>alt</td>
<td>Alternative PC Cyrillic</td>
</tr>
<tr>
<td>MS 1251</td>
<td>win5</td>
<td>Mac Cyrillic</td>
<td>mac</td>
<td>Macintosh Cyrillic</td>
</tr>
<tr>
<td>Mac Cyrillic</td>
<td>mac</td>
<td>ISO 8859-5</td>
<td>iso5</td>
<td>ISO 8859-5 Cyrillic</td>
</tr>
<tr>
<td>Mac Cyrillic</td>
<td>mac</td>
<td>KOI8-R</td>
<td>koi8</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>Mac Cyrillic</td>
<td>mac</td>
<td>PC Cyrillic</td>
<td>alt</td>
<td>Alternative PC Cyrillic</td>
</tr>
<tr>
<td>Mac Cyrillic</td>
<td>mac</td>
<td>MS 1251</td>
<td>win5</td>
<td>Windows Cyrillic</td>
</tr>
</tbody>
</table>

Font Formats

Location of Fonts on the System

Fonts to support European locales are available in various formats, such as bitmaps, Postscript™ Type-1, and TrueType. The actual availability varies per character set.

Fonts are located at:
/usr/openwin/lib/locale/iso_8859_x/X11/fonts/

Adding and Removing Font Packages

To manually add font packages to the system:

1. Always add the required font packages before the optional font packages.
2. Remove the optional font packages first, when you are removing font packages from the system.

You must follow this procedure to add or remove fonts. The class action scripts in the font packages depend on this to function properly. The optional font packages contain scripts that concatenate information onto the required font packages that are already resident on the system. If the required font packages are not there, problems can occur.
Summary of Asian Locales

The following table shows the Asian supported locales.

<table>
<thead>
<tr>
<th>CD Set</th>
<th>Locale Name</th>
<th>Description</th>
<th>Supported Character Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Korean</td>
<td>ko</td>
<td>Korean (EUC)</td>
<td>KSC 5601-1987</td>
</tr>
<tr>
<td></td>
<td>ko.UTF-8</td>
<td>Korean (UTF-8)</td>
<td>KSC 5601-1992</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>zh</td>
<td>Simplified Chinese (EUC)</td>
<td>GB 2312-1980</td>
</tr>
<tr>
<td></td>
<td>zh_GBK</td>
<td>Simplified Chinese (GBK)</td>
<td>GBK</td>
</tr>
<tr>
<td></td>
<td>zh.UTF-8</td>
<td>Simplified Chinese (UTF-8)</td>
<td>Unicode 3.0</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>zh_TW</td>
<td>Traditional Chinese (EUC)</td>
<td>CNS 11643–1992</td>
</tr>
<tr>
<td></td>
<td>zh_TW.BIG5</td>
<td>Traditional Chinese (BIG5)</td>
<td>BIG5</td>
</tr>
<tr>
<td></td>
<td>zh_TW.UTF-8</td>
<td>Traditional Chinese (UTF-8)</td>
<td>Unicode 3.0</td>
</tr>
<tr>
<td>Japanese</td>
<td>ja</td>
<td>Japanese (EUC)</td>
<td>JIS x 0201-1976</td>
</tr>
<tr>
<td></td>
<td>ja_JP.PCK</td>
<td>Japanese (PCK)</td>
<td>JIS x 0208-1990</td>
</tr>
<tr>
<td></td>
<td>ja_JP.UTF-8</td>
<td>Japanese (UTF-8)</td>
<td>JIS x 0212-1990</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VDC ²</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UDC ³</td>
</tr>
</tbody>
</table>

1. ja_JP.PCK (doesn’t support JIS x 0212-1990)
2. VDC: Vendor Defined Character. VDCs occupy unused (reserved) code points of JIS X 0208–1990 or JIS X 0212–1990
3. UDC: User Defined Character. UDCs occupy unused (reserved) code points of JIS X 0208–1990 or JIS X 0212–1990 (also unused for VDCs).
Simplified Chinese Localization

Simplified Chinese in the Solaris 8 environment provides three locales: zh, zh.UTF-8, and zh.GBK. In the zh locale, the EUC scheme is used to encode GB2312–80. The zh.GBK locale supports the GBK codeset, which is a superset of GB2312–80.

Simplified Chinese is used mostly in the People’s Republic of China (PRC) and in Singapore.

The following input methods are supported for the zh locale:

- New QuanPin
- New ShuangPin
- Quanpy
- Location
- PinYin
- Stroke
- Golden
- Intelligent Pinyin
- Simplified Chinese Symbol

The following input methods are supported for both the zh.GBK and the zh.UTF-8 locales:

- New QuanPin
- New ShuangPin
- Quanpy
- GBK Code
- Japanese
- Hanja
- Zhuyin
- Unicode

The following table shows the TrueType fonts for the zh locale.
### TABLE 3–18  Solaris 8 TrueType Fonts for the zh Locale

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Vendor</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fangsong</td>
<td>R</td>
<td>TrueType</td>
<td>Hanyi</td>
<td>GB2312.1980</td>
</tr>
<tr>
<td>Hei</td>
<td>R</td>
<td>TrueType</td>
<td>Monotype</td>
<td>GB2312.1980</td>
</tr>
<tr>
<td>Kai</td>
<td>R</td>
<td>TrueType</td>
<td>Monotype</td>
<td>GB2312.1980</td>
</tr>
<tr>
<td>Song</td>
<td>R</td>
<td>TrueType</td>
<td>Monotype</td>
<td>GB2312.1980</td>
</tr>
</tbody>
</table>

The following table shows the Bitmap Fonts for the zh Locale.

### TABLE 3–19  Solaris 8 Bitmap Fonts for the zh Locale

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Song</td>
<td>B</td>
<td>PCF (14,16)</td>
<td>GB2312.1980</td>
</tr>
<tr>
<td>Song</td>
<td>R</td>
<td>PCF (12,14,16,20,24)</td>
<td>GB2312.1980</td>
</tr>
</tbody>
</table>

The following table shows the TrueType fonts for the zh.GBK Locale.

### TABLE 3–20  TrueType Fonts for the zh.GBK Locale

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Vendor</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fansong</td>
<td>R</td>
<td>TrueType</td>
<td>Zhongyi</td>
<td>GBK</td>
</tr>
<tr>
<td>Hei</td>
<td>R</td>
<td>TrueType</td>
<td>Zhongyi</td>
<td>GBK</td>
</tr>
<tr>
<td>Kai</td>
<td>R</td>
<td>TrueType</td>
<td>Zhongyi</td>
<td>GBK</td>
</tr>
<tr>
<td>Song</td>
<td>R</td>
<td>TrueType</td>
<td>Zhongyi</td>
<td>GBK</td>
</tr>
</tbody>
</table>

The following table shows the Bitmap Fonts for the zh.GBK Locale.

### TABLE 3–21  Bitmap Fonts for the zh.GBK Locale

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Song</td>
<td>R</td>
<td>PCF (12,14,16,20,24)</td>
<td>GBK</td>
</tr>
</tbody>
</table>

The following table shows the supported codeset conversions for Simplified Chinese.
### TABLE 3-22 Codeset Conversions for Simplified Chinese

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Target Code</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>GB2312-80</td>
<td>zh_CN.euc</td>
<td>ISO 2022-7</td>
<td>zh_CN.iso2022-7</td>
</tr>
<tr>
<td>ISO 2022-7</td>
<td>zh_CN.iso2022-7</td>
<td>GB2312-80</td>
<td>zh_CN.euc</td>
</tr>
<tr>
<td>GB2312-80</td>
<td>zh_CN.euc</td>
<td>ISO 2022-CN</td>
<td>zh_CN.iso2022-CN</td>
</tr>
<tr>
<td>HZ-GB-2312</td>
<td>HZ-GB-2312</td>
<td>GB2312-80</td>
<td>zh_CN.euc</td>
</tr>
<tr>
<td>HZ-GB-2312</td>
<td>HZ-GB-2312</td>
<td>GBK</td>
<td>zh_CN.gbk</td>
</tr>
<tr>
<td>HZ-GB-2312</td>
<td>HZ-GB-2312</td>
<td>UTF-8</td>
<td>UTF-8</td>
</tr>
<tr>
<td>ISO-2022-CN</td>
<td>zh_CN.iso2022-CN</td>
<td>GB2312-80</td>
<td>zh_CN.euc</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-8</td>
<td>GB2312-80</td>
<td>zh_CN.euc</td>
</tr>
<tr>
<td>GB2312-80</td>
<td>zh_CN.euc</td>
<td>UTF-8</td>
<td>UTF-8</td>
</tr>
<tr>
<td>zh.GBK</td>
<td>zh_CN.gbk</td>
<td>ISO2022-CN</td>
<td>zh_CN.iso2022-CN</td>
</tr>
<tr>
<td>ISO2022-CN</td>
<td>zh_CN.iso2022-CN</td>
<td>zh.GBK</td>
<td>zh_CN.gbk</td>
</tr>
<tr>
<td>zh.GBK</td>
<td>zh_CN.gbk</td>
<td>Big-5</td>
<td>zh_TW-Big5</td>
</tr>
<tr>
<td>Big-5</td>
<td>zh_TW-Big5</td>
<td>zh.GBK</td>
<td>zh_CN.gbk</td>
</tr>
<tr>
<td>GB2312-80</td>
<td>zh_CN.euc</td>
<td>Big-5</td>
<td>zh_TW-Big5</td>
</tr>
<tr>
<td>Big-5</td>
<td>zh_TW-Big5</td>
<td>GB2312-80</td>
<td>zh_CN.euc</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-8</td>
<td>zh.GBK</td>
<td>zh_CN.gbk</td>
</tr>
<tr>
<td>zh.GBK</td>
<td>zh_CN.gbk</td>
<td>UTF-8</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-8</td>
<td>ISO2022-CN</td>
<td>zh_CN.iso2022-CN</td>
</tr>
<tr>
<td>ISO2022-CN</td>
<td>zh_CN.iso2022-CN</td>
<td>UTF-8</td>
<td>UTF-8</td>
</tr>
</tbody>
</table>

### Traditional Chinese Localization

Traditional Chinese in the Solaris 8 product provides three locales: `zh_TW`, `zh_TW.UTF-8` and `zh_TW.BIG5`. In the `zh_TW` locale, the EUC scheme is used to encode CNS 11643.1992 codeset. The `zh_TW.BIG5` locale supports the Big-5 codeset. The `zh_TW.UTF-8` locale supports Unicode 3.0.
Traditional Chinese is used mostly in Taiwan and Hong Kong, and supports the following input methods:

- Chuyin
- I-Tien
- Telecode
- TsangChieh
- CheinI
- NeiMa
- ChuangHsing
- Array
- BoShiaMy
- DaYi

The following table shows Traditional Chinese Truetype Fonts for the zh_TW Locales.

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Vendor</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hei R</td>
<td>Truetype</td>
<td>Hanyi</td>
<td></td>
<td>CNS11643.1992</td>
</tr>
<tr>
<td>Kai R</td>
<td>Truetype</td>
<td>Hanyi</td>
<td></td>
<td>CNS11643.1992</td>
</tr>
<tr>
<td>Ming R</td>
<td>Truetype</td>
<td>Hanyi</td>
<td></td>
<td>CNS11643.1992</td>
</tr>
</tbody>
</table>

The following table shows the Traditional Chinese BitMap Fonts for the zh_TW Locales.

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ming R</td>
<td>PCF (12,14,16,20,24)</td>
<td>CNS11643.1992</td>
<td></td>
</tr>
</tbody>
</table>

The following table shows the Traditional Chinese TrueType Fonts for the zh_TW_BIG5 Locales.
<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Vendor</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hei R</td>
<td>TrueType</td>
<td>Hanyi</td>
<td>Big5</td>
<td></td>
</tr>
<tr>
<td>Kai R</td>
<td>TrueType</td>
<td>Hanyi</td>
<td>Big5</td>
<td></td>
</tr>
<tr>
<td>Ming R</td>
<td>TrueType</td>
<td>Hanyi</td>
<td>Big5</td>
<td></td>
</tr>
</tbody>
</table>

The following table shows the Traditional Chinese BitMap Fonts for the zh_TW.BIG5 Locales.

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ming R</td>
<td>R</td>
<td>PCF (12,14,16,20,24)</td>
<td>Big5</td>
</tr>
</tbody>
</table>

The following table shows the supported codeset conversions for Traditional Chinese.

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Target Code</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNS 11643</td>
<td>zh_TW-euc</td>
<td>Big-5</td>
<td>zh_TW-Big5</td>
</tr>
<tr>
<td>CNS 11643</td>
<td>zh_TW-euc</td>
<td>ISO 2022-7</td>
<td>zh_TW-iso2022-7</td>
</tr>
<tr>
<td>Big-5</td>
<td>zh_TW-Big5</td>
<td>CNS 11643</td>
<td>zh_TW-euc</td>
</tr>
<tr>
<td>Big-5</td>
<td>zh_TW-Big5</td>
<td>ISO 2022-7</td>
<td>zh_TW-iso2022-7</td>
</tr>
<tr>
<td>ISO 2022-7</td>
<td>zh_TW-iso2022-7</td>
<td>CNS 11643</td>
<td>zh_TW-euc</td>
</tr>
<tr>
<td>ISO 2022-7</td>
<td>zh_TW-iso2022-7</td>
<td>Big-5</td>
<td>zh_TW-Big5</td>
</tr>
<tr>
<td>CNS 11643</td>
<td>zh_TW-eu</td>
<td>ISO 2022-CN-EXT</td>
<td>zh_TW-iso2022-CN-EXT</td>
</tr>
<tr>
<td>ISO 2022-CN-EXT</td>
<td>zh_TW-iso2022-CN-EXT</td>
<td>CNS 11643</td>
<td>zh_TW-euc</td>
</tr>
<tr>
<td>Big-5</td>
<td>zh_TW-Big5</td>
<td>ISO 2022-CN</td>
<td>zh_TW-iso2022-CN</td>
</tr>
<tr>
<td>ISO 2022-CN</td>
<td>zh_TW-iso2022-CN</td>
<td>Big-5</td>
<td>zh_TW-Big5</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-8</td>
<td>CNS 11643</td>
<td>zh_TW-euc</td>
</tr>
<tr>
<td>CNS 11643</td>
<td>zh_TW-euc</td>
<td>UTF-8</td>
<td>UTF-8</td>
</tr>
</tbody>
</table>
Japanese Localization

This section describes Japanese locale-specific information.

Japanese Locales

Three Japanese locales, which support different character encoding, are available in the Solaris 8 environment. The ja, (or ja\_JP\_eucJP) locale is based on the Japanese EUC. The ja\_JP\_PCK locale is based on PC-Kanji code (known as Shift-JIS) and the ja\_JP\_UTF-8 is based on UTF-8. See eucJP(5) for a map between Japanese EUC and the character set. See PCK(5) for the map between PCK and the character set.

Japanese Character Set

Supported Japanese character sets are:

- JISX0201–1976
- JISX0208–1990
- JISX0212–1990

JISX0212–1990 is not supported in the ja\_JP\_PCK locale.

Vendor Defined Character (VDC) and User defined Character (UDC) are also supported. VDCs occupy unused (reserved) code points of JISX0208–1990 or
JISX0212–1990. UDCs occupy the same code points as VDCs except the code points are for VDCs.

**Japanese Font**

Three Japanese font formats are supported. They are: Bitmap, TrueType and Type1. The Japanese Type1 font includes only JIS X0212 for printing. Type1 font is also used by UDC.

Japanese Bitmap Fonts are shown below.

**TABLE 3-28** Japanese Bitmap Fonts

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Vendor</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>gothic</td>
<td>R, B</td>
<td>PCF(12,14,16,20,24)</td>
<td></td>
<td>JISX0208.1983, JISX0201.1976</td>
</tr>
<tr>
<td>minchou</td>
<td>R</td>
<td>PCF(12,14,16,20,24)</td>
<td></td>
<td>JISX0208.1983, JISX0201.1976</td>
</tr>
<tr>
<td>hg gothic b</td>
<td>R</td>
<td>PCF(12,14,16,18,20,24)</td>
<td>RICOH</td>
<td>JISX0208.1983, JISX0201.1976</td>
</tr>
<tr>
<td>hg mincho l</td>
<td>R</td>
<td>PCF(12,14,16,18,20,2)</td>
<td>RICOH</td>
<td>JISX0208.1983, JISX0201.1976</td>
</tr>
<tr>
<td>heiseimin</td>
<td>R</td>
<td>PCF(12,14,16,18,20,24)</td>
<td>RICOH</td>
<td>JISX0212.1990</td>
</tr>
</tbody>
</table>

Japanese TrueType Fonts are show below.

**TABLE 3-29** Japanese TrueType Fonts

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Vendor</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>hg gothic b</td>
<td>R</td>
<td>TrueType</td>
<td>RICOH</td>
<td>JISX0208.1983, JISX0201.1976</td>
</tr>
<tr>
<td>hg mincho l</td>
<td>R</td>
<td>TrueType</td>
<td>RICOH</td>
<td>JISX0208.1983, JISX0201.1976</td>
</tr>
<tr>
<td>heiseimin</td>
<td>R</td>
<td>TrueType</td>
<td>RICOH</td>
<td>JISX0212.1990</td>
</tr>
</tbody>
</table>
Japanese Input Systems

Four Japanese input systems, ATOK12, ATOK8, Wnn6, and cs00 are available in the Solaris 8 environment for all Japanese locales. It is possible to switch input systems from the workspace menu. The only Japanese input system available on the Base Solaris is cs00.

How to Input Japanese Strings by using cs00

When turning Kana-Kanji conversion mode ON, keyboard input is grabbed by Htt (X Input Method Server) and sent to the cs00 daemon through the XCI (xci(7)) interface. The cs00 daemon converts the received strings to Japanese strings by using dictionary and returns the result to the program which has a keyboard focus now. See cs00(1M) for more details.

CUI based dictionary maintenance utilities are available. See udicm(1) and mdicm(1) for details.

GUI based maintenance utilities, sdtudicm(1) or udicmttool(1), are not available in the base Solaris product.

The basic Japanese input procedure is as follows:

1. Turning Japanese conversion mode on/off: Control + Space
2. Enter Kana character text: ex: Type “nihon”
3. Conversion to Kanji character text: Control + N
4. Commit the Kanji character text: Control + K

The following table shows cs00 operation list.

<table>
<thead>
<tr>
<th>Function</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conversion mode on/off</td>
<td>Control + Space</td>
</tr>
<tr>
<td></td>
<td>Control + @</td>
</tr>
<tr>
<td>Kana/Kanji conversion</td>
<td>next Control + N</td>
</tr>
<tr>
<td></td>
<td>post Control + P</td>
</tr>
<tr>
<td></td>
<td>lookup Control + W</td>
</tr>
<tr>
<td>Commit</td>
<td>Control + K</td>
</tr>
<tr>
<td>Move focus</td>
<td>forward Control + F</td>
</tr>
<tr>
<td></td>
<td>back Control + B</td>
</tr>
</tbody>
</table>

TABLE 3–30  cs00 Operation List
**TABLE 3-30 cs00 Operation List (continued)**

<table>
<thead>
<tr>
<th>Function</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus scope</td>
<td>increase Control + I</td>
</tr>
<tr>
<td></td>
<td>decrease Control + U</td>
</tr>
<tr>
<td>Delete (1 character)</td>
<td>Control + H</td>
</tr>
<tr>
<td></td>
<td>Delete or backspace</td>
</tr>
<tr>
<td>Delete (all characters)</td>
<td>Control + ] and Control + U</td>
</tr>
<tr>
<td>Full/half Katakana =&gt; Hiragana</td>
<td>Control + ] and Control + O</td>
</tr>
<tr>
<td>Hiragana/half Katakana =&gt; full Katakana</td>
<td>Control + ] and Control + Y</td>
</tr>
<tr>
<td>Full Katakana/Hiragana =&gt; half Katakana</td>
<td>Control + ] and Control + Z</td>
</tr>
<tr>
<td>Half Roma/Num =&gt; full Roma/Num</td>
<td>Control + ] and Control + T</td>
</tr>
<tr>
<td>Full Roma/Num =&gt; half Roma/Num</td>
<td>Control + ] and Control + R</td>
</tr>
<tr>
<td>Learning Mode on/off</td>
<td>Control + ] and Control + L</td>
</tr>
<tr>
<td>Input Mode Switch:</td>
<td></td>
</tr>
<tr>
<td>• Hiragana mode</td>
<td>Control + O</td>
</tr>
<tr>
<td>• Full Katakana mode</td>
<td>Control + Y</td>
</tr>
<tr>
<td>• Full Roma/Num mode</td>
<td>Control + T</td>
</tr>
<tr>
<td>• Half Katakana mode</td>
<td>Control + Z</td>
</tr>
<tr>
<td>• Half Roma/Num mode</td>
<td>Control + R</td>
</tr>
<tr>
<td>• Kuten code input mode</td>
<td>Control + Q</td>
</tr>
<tr>
<td>• Bushu input mode</td>
<td>Control + V</td>
</tr>
</tbody>
</table>

**Terminal Setting for Japanese Terminals**

Using Japanese locales on a character based terminal (TTY) requires that you use terminal settings to make line editing work correctly.

- If your terminal is a CDE Terminal emulator (dtterm), use `stty(1)` with argument `-defeucw`, in any Japanese locale (ja, ja_JP.PCK, or ja_JP.UTF-8). An example in locale ja is:

  ```
  % setenv LANG ja
  % stty defeucw
  ```

- If your terminal is not a CDE Terminal emulator, but the codeset of your terminal is the same as that of the current locale, use this setting, too.
If your terminal’s codeset doesn’t match that of the current locale, use `setterm(1)` to enable code conversion. For example, if you are in locale `ja` but your terminal requires PCK (ShiftJIS code), specify:

```
% setenv LANG ja
% setterm -x PCK
```

See `setterm(1)` for details.

**Japanese `iconv` Module**

Several Japanese codeset conversions are supported with `iconv(1)` and `iconv(3)`. See the `iconv_ja(5)` man page for details.

The following table shows `iconv` Conversion Support.

<table>
<thead>
<tr>
<th>Source Code</th>
<th>Target Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>eucJP</td>
<td>JIS7</td>
</tr>
<tr>
<td>eucJP</td>
<td>SJIS</td>
</tr>
<tr>
<td>eucJP</td>
<td>UTF-8</td>
</tr>
<tr>
<td>eucJP</td>
<td>jis</td>
</tr>
<tr>
<td>eucJP</td>
<td>ibmj</td>
</tr>
<tr>
<td>SJIS</td>
<td>eucJP</td>
</tr>
<tr>
<td>SJIS</td>
<td>ISO-2022-JP</td>
</tr>
<tr>
<td>SJIS</td>
<td>UTF-8</td>
</tr>
<tr>
<td>SJIS</td>
<td>jis</td>
</tr>
<tr>
<td>SJIS</td>
<td>ibmj</td>
</tr>
<tr>
<td>PCK</td>
<td>eucJP</td>
</tr>
<tr>
<td>PCK</td>
<td>UTF-8</td>
</tr>
<tr>
<td>PCK</td>
<td>ISO-2022-JP</td>
</tr>
<tr>
<td>PCK</td>
<td>jis</td>
</tr>
<tr>
<td>PCK</td>
<td>ibmj</td>
</tr>
</tbody>
</table>
### TABLE 3–31  iconv Conversion Support  (continued)

<table>
<thead>
<tr>
<th>Source Code</th>
<th>Target Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO-2022-JP</td>
<td>PCK</td>
</tr>
<tr>
<td>ISO-2022-JP</td>
<td>SJIS</td>
</tr>
<tr>
<td>UTF-8</td>
<td>eucJP</td>
</tr>
<tr>
<td>UTF-8</td>
<td>SJIS</td>
</tr>
<tr>
<td>UTF-8</td>
<td>PCK</td>
</tr>
<tr>
<td>JIS7</td>
<td>eucJP</td>
</tr>
<tr>
<td>jis</td>
<td>eucJP</td>
</tr>
<tr>
<td>jis</td>
<td>PCK</td>
</tr>
<tr>
<td>jis</td>
<td>SJIS</td>
</tr>
<tr>
<td>ibmj</td>
<td>eucJP</td>
</tr>
<tr>
<td>ibmj</td>
<td>PCK</td>
</tr>
<tr>
<td>UTF-8</td>
<td>ISO-2022-JP</td>
</tr>
<tr>
<td>ISO-2022-JP</td>
<td>UTF-8</td>
</tr>
<tr>
<td>eucJP</td>
<td>UTF-8-Java</td>
</tr>
<tr>
<td>UTF-8-Java</td>
<td>eucJP</td>
</tr>
<tr>
<td>PCK</td>
<td>UTF-8-Java</td>
</tr>
<tr>
<td>UTF-8-Java</td>
<td>PCK</td>
</tr>
<tr>
<td>PCK</td>
<td>ISO-2022-JP.RFC1468</td>
</tr>
<tr>
<td>UTF-8</td>
<td>ISO-2022-JP.RFC1468</td>
</tr>
<tr>
<td>eucJP</td>
<td>ibmj-EBCDIK</td>
</tr>
<tr>
<td>ibmj-EBCDIK</td>
<td>eucJP</td>
</tr>
<tr>
<td>PCK</td>
<td>ibmj-EBCDIK</td>
</tr>
<tr>
<td>ibmj-EBCDIK</td>
<td>PCK</td>
</tr>
</tbody>
</table>

### Japanese Specific Printer Support

The Japanese Solaris 8 product supports the following Japanese-specific printers:
User Defined Character Support

To handle UDC, sdtudctool is available. Sdtudctool handles both outline (Type1) and bitmap (PCF) fonts. Some utilities are also available to migrate the UDC fonts that were created by old utilities in prior releases, such as fontedit, type3creator, and fontmanager.

Not Included on the Base Solaris Product

The following components are included in the multilingual Solaris product (on Languages CD), but not included in the base Solaris product.

- All translations such as message, help, manpage and document
- Japanese BCP support
- ATOK12, ATOK8, and Wnn6 Japanese input systems
- GUI utilities of the cs00 Japanese input system
- Mincho and Bold typeface fonts
- Japanese-specific dumb printer support
- Sdtudctool for UDC
- Legacy Japanese libraries (for example, libjapanese.a or libmle.a)
- Some Japanese specific utilities (e.g. kanji, or vled)

Korean Localization

In December 1995, the Korean government announced a standard Korean codeset, KS C 5700, which is based on ISO 10646-1/Unicode 2.0.

The ISO-10646 character set uses 2 (UCS-2); Universal Character Set (two-byte form) or 4 (UCS-4) bytes to represent each character.

The ISO-10646 character set cannot be used directly on IBM-PC-based operating systems. For example, the kernel and many other modules of the Solaris operating environment interpret certain byte values as control instructions, such as a null character (0x00) in any string. The ISO-10646 character set can be encoded with any bit combinations in the first or subsequent bytes. The ISO-10646 characters cannot be freely transmitted through the Solaris system with these limitations. In order to
establish a migration path, the ISO-10646 character set defines the UCS Transformation Format (UTF), which recodes the ISO-10646 characters without using C0 controls (0x00..0x1F), C1 controls (0x80..0x9F), space (0x20), and DEL (0x7F).

The \texttt{ko.UTF-8} is a Solaris locale to support KSC-5700, the Korean standard codeset. It supports all characters in the previous KSC 5601 and all 11,172 Korean characters. Korean UTF-8 supports the Korean language-related ISO-10646 characters and fonts. Because ISO-10646 covers all characters in the world, all of the various input methods and fonts are supplied so that you can input and output any character in any language.

Before Universal UTF/UCS becomes available, Korean UTF-8 supports the ISO-10646 code subset that is related to Korean characters as well as all other characters in the previous Korean standard codeset, and Extended ASCII.

In the \texttt{ko} locale, the EUC scheme is used to encode KSC 5601-1987. The \texttt{ko.UTF-8} locale supports the KSC 5700-1995/Unicode 2.0 codeset, which is a super set of KSC 5601-1987. These two locales look the same to the end user, but the internal character encoding is different. The Korean Solaris product supports the following Input Methods:

For the \texttt{ko} locale:
- Hangul 2–BeolSik (1 set of consonants and 1 set of vowels)
- Hangul-Hanja conversion
- Special character
- Hexadecimal code

For the \texttt{ko.UTF-8} locale:
- Hangul 2–BeolSik (1 set of consonants and 1 set of vowels)
- Hangul-Hanja conversion
- Special character
- Hexadecimal code

\begin{table}[h]
\centering
\caption{Solaris 8 Korean CID/Type 1 Fonts for the \texttt{ko} Locale}
\begin{tabular}{|l|l|l|l|l|}
\hline
Full Family Name & Subfamily & Format & Vendor & Encoding \\
\hline
Gothic & R & CID/Type 1 & Hanyang & Adobe-Korean \\
Graphic & R & CID/Type 1 & Hanyang & Adobe-Korean \\
Haeso & R & CID/Type 1 & Hanyang & Adobe-Korean \\
Kodig & R & CID/Type 1 & Hanyang & Adobe-Korean \\
Myeongijo & R & CID/Type 1 & Hanyang & Adobe-Korean \\
Pilki & R & CID/Type 1 & Hanyang & Adobe-Korean \\
Roundgothic & R & CID/Type 1 & Hanyang & Adobe-Korean \\
\hline
\end{tabular}
\end{table}
### TABLE 3–33  Solaris 8 Korean Bitmap Fonts for the ko Locale

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gothic</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1987</td>
</tr>
<tr>
<td>Graphic</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1987</td>
</tr>
<tr>
<td>Haeso</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1987</td>
</tr>
<tr>
<td>Kodig</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1987</td>
</tr>
<tr>
<td>Myeongijo</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1987</td>
</tr>
<tr>
<td>Pilki</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1987</td>
</tr>
<tr>
<td>Roundgothic</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1987</td>
</tr>
</tbody>
</table>

### TABLE 3–34  Solaris 8 Korean CID/Type 1 Fonts for the ko.UTF-8 Locale

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Vendor</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gothic</td>
<td>R</td>
<td>CID/Type 1</td>
<td>Hanyang</td>
<td>Adobe-Korean</td>
</tr>
<tr>
<td>Graphic</td>
<td>R</td>
<td>CID/Type 1</td>
<td>Hanyang</td>
<td>Adobe-Korean</td>
</tr>
<tr>
<td>Haeso</td>
<td>R</td>
<td>CID/Type 1</td>
<td>Hanyang</td>
<td>Adobe-Korean</td>
</tr>
<tr>
<td>Kodig</td>
<td>R</td>
<td>CID/Type 1</td>
<td>Hanyang</td>
<td>Adobe-Korean</td>
</tr>
<tr>
<td>Myeongijo</td>
<td>R</td>
<td>CID/Type 1</td>
<td>Hanyang</td>
<td>Adobe-Korean</td>
</tr>
<tr>
<td>Pilki</td>
<td>R</td>
<td>CID/Type 1</td>
<td>Hanyang</td>
<td>Adobe-Korean</td>
</tr>
</tbody>
</table>

### TABLE 3–35  Solaris 8 Korean Bitmap Fonts for the ko.UTF-8 Locale

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gothic</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1992</td>
</tr>
<tr>
<td>Graphic</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1992</td>
</tr>
<tr>
<td>Haeso</td>
<td>R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1992</td>
</tr>
</tbody>
</table>

Contents of Solaris 8 Products  81
### TABLE 3–35  Solaris 8 Korean Bitmap Fonts for the ko.UTF-8 Locale (continued)

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kodig R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1992 (Johap)</td>
<td></td>
</tr>
<tr>
<td>Myeongijo R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1992 (Johap)</td>
<td></td>
</tr>
<tr>
<td>Pilki R/B</td>
<td>PCF (12,14,16,18,20,24)</td>
<td>KSC 5601-1992 (Johap)</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 3–36  Solaris 8 Korean TrueType Fonts for the ko/ko.UTF-8 Locales

<table>
<thead>
<tr>
<th>Full Family Name</th>
<th>Subfamily</th>
<th>Format</th>
<th>Vendor</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kodig/Gothic R</td>
<td>True Type</td>
<td>Hanyang</td>
<td>Unicode</td>
<td></td>
</tr>
<tr>
<td>Myeongjo R</td>
<td>True Type</td>
<td>Hanyang</td>
<td>Unicode</td>
<td></td>
</tr>
<tr>
<td>Haeso R</td>
<td>True Type</td>
<td>Hanyang</td>
<td>Unicode</td>
<td></td>
</tr>
<tr>
<td>RoundGothic R</td>
<td>True Type</td>
<td>Hanyang</td>
<td>Unicode</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 3–37  Korean ICONV

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Target Code</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSC 5601-1987</td>
<td>1506</td>
<td>UTF-8</td>
<td>UTF-8</td>
</tr>
<tr>
<td>ISO 646</td>
<td>646</td>
<td>KSC 5601-1987</td>
<td>5601</td>
</tr>
<tr>
<td>KSC 5601-1987</td>
<td>EUC-KR</td>
<td>UTF-8</td>
<td>UTF-8</td>
</tr>
<tr>
<td>KSC 5601-1987</td>
<td>KSC5601</td>
<td>UTF-8</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-8</td>
<td>KSC 5601-1987</td>
<td>5601</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-8</td>
<td>KSC 5601-1987</td>
<td>EUC-KR</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-8</td>
<td>KSC 5601-1987</td>
<td>KSC 5601</td>
</tr>
<tr>
<td>UTF-8</td>
<td>ko-KR-UTF-8</td>
<td>IBM CP 933</td>
<td>cp 933</td>
</tr>
</tbody>
</table>

International Language Environments Guide • February 2000
<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Target Code</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM CP933</td>
<td>cp933</td>
<td>UTF-8</td>
<td>ko_KR-UTF-8</td>
</tr>
</tbody>
</table>
Overview of en_US.UTF-8 Locale Support

Unicode Overview

The Unicode Standard is the universal character encoding standard used for representation of text for computer processing. It is fully compatible with the International Standard ISO/IEC 10646-1:1999, and contains all the same characters and encoding points as ISO/IEC 10646. The Unicode Standard provides additional information about the characters and their use. Any implementation that conforms to Unicode also conforms to ISO/IEC 10646.

Unicode provides a consistent way of encoding multilingual plain text and brings order to a chaotic state of affairs that has made it difficult to exchange text files internationally. Computer users who deal with multilingual text, business people, linguists, researchers, scientists, and others, find that the Unicode Standard greatly simplifies their work. Mathematicians and technicians, who regularly use mathematical symbols and other technical characters, also find the Unicode Standard valuable.

The design of Unicode is based on the simplicity and consistency of ASCII, but goes beyond ASCII’s limited ability to encode only the Latin alphabet. The Unicode Standard provides the capacity to encode all of the characters used for the written languages of the world. It uses a 16-bit encoding that provides code points for more than 65,000 characters. To keep character coding simple and efficient, the Unicode Standard assigns each character a unique 16-bit value, and does not use complex modes or escape codes. While 65,000 characters are sufficient for encoding most of the many thousands of characters used in major languages of the world, the Unicode standard and ISO 10646 provide an extension mechanism called UTF-16 that allows for encoding as many as a million more characters, without use of escape codes. This is sufficient for all known character encoding requirements, including full coverage
of all historic scripts of the world. UTF-16 allows exactly 16 x 65536 additional code points and still uses the two byte entities to represent characters. However those 16 x 65536 characters require two two byte entities (for a total of four bytes) per each character. For more details on the UTF-16, refer to section C.3 of “The Unicode Standard, Version 2.0” from Unicode Consortium, or Annex C of ISO/IEC 10646–1:1999, Information Technology—Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.

Unicode Locale: en_US.UTF-8 Support Overview

The en_US.UTF-8 locale is a significant Unicode locale in the Solaris 8 product. It supports and provides multiscript processing capability by using UTF-8 as its codeset. It can input and output text in multiple scripts. This was the first locale with this capability in the Solaris operating environment.

UTF-8 is a file system safe Universal Character Set Transformation Format of Unicode / ISO/IEC 10646-1 formulated by X/Open-Uniforum Joint Internationalization Working Group (XoJIG) in 1992 and approved by ISO and IEC, as Amendment 2 to ISO/IEC 10646-1:1993 in 1996. This standard has been adopted by the Unicode Consortium, the International Standards Organization, and the International Electrotechnical Commission as a part of Unicode 2.0 and ISO/IEC 10646-1.

en_US.UTF-8 supports computation for every code point value, which is defined in Unicode 3.0 and ISO/IEC 10646-1. In the Solaris 8 environment, language script support is not limited to pan-European locales, but also includes Asian scripts such as Korean, Traditional Chinese, Simplified Chinese, and Japanese. Due to limited font resources, Solaris 8 software includes only character glyphs from the following character sets:

- ISO 8859-1 (most Western European languages, such as English, French, Spanish, and German)
- ISO 8859-2 (most Central European languages, such as Czech, Polish, and Hungarian)
- ISO 8859-4 (Scandinavian and Baltic languages)
- ISO 8859-5 (Russian)
- ISO 8859-6 (Arabic, including many more presentation form character glyphs)
- ISO 8859–7 (Greek)
- ISO 8859–8 (Hebrew)
- ISO 8859-9 (Turkish)
- TIS 620.2533 (Thai, including many more presentation form character glyphs)
ISO 8859–15 (most Western European languages with euro sign)
GB 2312–1980 (Simplified Chinese)
Big5 (Traditional Chinese)
KS C 5601–1992 Annex 3 (Korean)

If a user displays characters for which the en_US.UTF-8 locale does not have corresponding glyphs, the locale displays ‘no-glyph’ glyph instead, as in the following example:

```
Starting with the Solaris 8 environment, the locale is available for all clusters except the Core cluster.

Exactly the same level of en_US.UTF-8 locale support is provided for both 64-bit and 32-bit Solaris systems.

Motif and CDE desktop applications and libraries support the en_US.UTF-8 locale. However, OpenWindows, XView, and, OPENLOOK DeskSet applications and libraries do not support the en_US.UTF-8 locale.

Desktop Input Methods

CDE provides the ability to enter localized input for an internationalized application that is using Xm Toolkit. The XmText[Field] widgets are enabled to interface with input methods from each locale. Input methods are internationalized because some language environments write their text from right-to-left, top-to-bottom, and so forth. Within the same application, you can use several fonts that apply different input methods.

The pre-edit area displays the string that is being pre-edited. This can be done in four modes:

- OffTheSpot
- OverTheSpot (default)
In OffTheSpot mode, the location is just below the MainWindow area at the right of the status area. In OverTheSpot mode, the pre-edit area is at the cursor point. In Root mode, the pre-edit and status areas are separate from the client’s window.

In the Solaris 8 environment, there are native Asian input methods for Simplified/Traditional Chinese, Japanese, and Korean in addition to the current multi-script input methods for Unicode locales. This section includes descriptions of selected input methods, how to use them, and how to switch between them.

### Script Selection and Input Modes

The `en_US.UTF-8` locale supports multiple scripts. The `en_US.UTF-8` locale has a total of twelve input modes:

- English/European
- Cyrillic
- Greek
- Arabic
- Hebrew
- Thai
- Unicode Hexadecimal and Octal code input methods
- Table lookup input method
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

To switch into a certain input mode, you can either type in an input mode switch compose key sequence for each input mode, or press the left-most mouse button at the status area of your application to open an input mode selection window and select from the listed input modes as follows:
English/European Input Mode

The English/European input mode includes not only the English alphabet but also characters with diacritical marks (for example, á, è, í, ò, and ü) and special characters (such as ¡, §, ¿) from European scripts.

This input mode is the default mode for any application. The input mode is displayed at the bottom left corner of the GUI application.

To insert characters with diacritical marks or special characters from Latin-1, Latin-2, Latin-4, Latin-5, and Latin-9, you must type a Compose Sequence, as shown in the following examples:
For Å, press and release Compose, then A, and then ".
For ¿, press and release Compose, then ?, and then ?.

When there is no <Compose> key available on your keyboard, you can substitute for the <Compose> key by simultaneously pressing the <Control> key, the <Shift> key and the <t> keys together.

For the input of the Euro currency symbol (Unicode value U+20AC) from the locale, you can use any one of following input sequences:

- <AltGraph> and <e> together
- <AltGraph> and <4> together
- <AltGraph> and <5> together

These input sequences mean that you press both keys simultaneously. If there is no <AltGraph> key available on your keyboard, you can substitute the <Alt> key for the <AltGraph> key.

The following tables show the most commonly used Compose Sequences in Latin-1, Latin-2, Latin-4, Latin-5, and Latin-9 script input for the Solaris operating environment.

To start these sequences, press the <Compose> key and release it.

The following table lists the Common Latin-1 Compose Sequences.

### TABLE 4–1  Common Latin-1 Compose Sequences

<table>
<thead>
<tr>
<th>Press and Release</th>
<th>Press and Release</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Spacebar]</td>
<td>[Spacebar]</td>
<td>No-break space</td>
</tr>
<tr>
<td>s</td>
<td>1</td>
<td>Superscripted 1</td>
</tr>
<tr>
<td>s</td>
<td>2</td>
<td>Superscripted 2</td>
</tr>
<tr>
<td>s</td>
<td>3</td>
<td>Superscripted 3</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>Inverted exclamation mark</td>
</tr>
<tr>
<td>x</td>
<td>o</td>
<td>Currency symbol ¤</td>
</tr>
<tr>
<td>p</td>
<td>!</td>
<td>Paragraph symbol ¶</td>
</tr>
<tr>
<td>/</td>
<td>u</td>
<td>mu u</td>
</tr>
<tr>
<td>'</td>
<td>&quot;</td>
<td>acute accent</td>
</tr>
<tr>
<td>,</td>
<td>,</td>
<td>cedilla Ç</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
<td>diaeresis.</td>
</tr>
<tr>
<td>Press and Release</td>
<td>Press and Release</td>
<td>Result</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>-</td>
<td>^</td>
<td>macron.</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>degree °</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>multiplication sign ×</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>plus-minus ±</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>soft hyphen –</td>
</tr>
<tr>
<td>-</td>
<td>:</td>
<td>division sign ÷</td>
</tr>
<tr>
<td>-</td>
<td>a</td>
<td>ordinal (feminine) &quot;</td>
</tr>
<tr>
<td>-</td>
<td>o</td>
<td>ordinal (masculine) &quot;</td>
</tr>
<tr>
<td>-</td>
<td>,</td>
<td>not sign ¬</td>
</tr>
<tr>
<td>-</td>
<td></td>
<td>middle dot ·</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>vulgar fraction ½</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>vulgar fraction ¼</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>vulgar fraction ⅓</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>left double angle quotation mark «</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
<td>right double angle quotation mark »</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>inverted question mark ¿</td>
</tr>
<tr>
<td>A</td>
<td>'</td>
<td>A grave Å</td>
</tr>
<tr>
<td>A</td>
<td>’</td>
<td>A acute Å</td>
</tr>
<tr>
<td>A</td>
<td>*</td>
<td>A ring above Å</td>
</tr>
<tr>
<td>A</td>
<td>&quot;</td>
<td>A diaeresis Å</td>
</tr>
<tr>
<td>A</td>
<td>^</td>
<td>A circumflex Å</td>
</tr>
<tr>
<td>A</td>
<td>~</td>
<td>A tilde Å</td>
</tr>
<tr>
<td>A</td>
<td>E</td>
<td>AE diphthong Æ</td>
</tr>
<tr>
<td>C</td>
<td>‘</td>
<td>C cedilla Ç</td>
</tr>
<tr>
<td>C</td>
<td>©</td>
<td>copyright sign ©</td>
</tr>
<tr>
<td>Press and Release</td>
<td>Press and Release</td>
<td>Result</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>Capital eth ð</td>
</tr>
<tr>
<td>E</td>
<td>’</td>
<td>E grave È</td>
</tr>
<tr>
<td>E</td>
<td>’</td>
<td>E acute È</td>
</tr>
<tr>
<td>E</td>
<td>“</td>
<td>E diaeresis Ê</td>
</tr>
<tr>
<td>E</td>
<td>^</td>
<td>E circumflex Ê</td>
</tr>
<tr>
<td>I</td>
<td>’</td>
<td>I grave Ï</td>
</tr>
<tr>
<td>I</td>
<td>’</td>
<td>I acute Ï</td>
</tr>
<tr>
<td>I</td>
<td>“</td>
<td>I diaeresis Ì</td>
</tr>
<tr>
<td>I</td>
<td>^</td>
<td>I circumflex Ì</td>
</tr>
<tr>
<td>L</td>
<td>-</td>
<td>pound sign £</td>
</tr>
<tr>
<td>N</td>
<td>~</td>
<td>N tilde Ñ</td>
</tr>
<tr>
<td>O</td>
<td>’</td>
<td>O grave Ô</td>
</tr>
<tr>
<td>O</td>
<td>’</td>
<td>O acute Ô</td>
</tr>
<tr>
<td>O</td>
<td>/</td>
<td>O slash Ø</td>
</tr>
<tr>
<td>O</td>
<td>“</td>
<td>O diaeresis Ô</td>
</tr>
<tr>
<td>O</td>
<td>^</td>
<td>O circumflex Ô</td>
</tr>
<tr>
<td>O</td>
<td>~</td>
<td>O tilde Ô</td>
</tr>
<tr>
<td>R</td>
<td>O</td>
<td>registered mark ®</td>
</tr>
<tr>
<td>T</td>
<td>H</td>
<td>Thorn þ</td>
</tr>
<tr>
<td>U</td>
<td>’</td>
<td>U grave Ü</td>
</tr>
<tr>
<td>U</td>
<td>’</td>
<td>U acute Ü</td>
</tr>
<tr>
<td>U</td>
<td>“</td>
<td>U diaeresis Ü</td>
</tr>
<tr>
<td>U</td>
<td>^</td>
<td>U circumflex Ü</td>
</tr>
<tr>
<td>Y</td>
<td>’</td>
<td>Y acute ý</td>
</tr>
<tr>
<td>Press and Release</td>
<td>Press and Release</td>
<td>Result</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Y</td>
<td>-</td>
<td>yen sign ¥</td>
</tr>
<tr>
<td>a</td>
<td>'</td>
<td>a grave à</td>
</tr>
<tr>
<td>a</td>
<td>'</td>
<td>a acute à</td>
</tr>
<tr>
<td>a</td>
<td>*</td>
<td>a ring above à</td>
</tr>
<tr>
<td>a</td>
<td>&quot;</td>
<td>a diaeresis å</td>
</tr>
<tr>
<td>a</td>
<td>~</td>
<td>a tilde à</td>
</tr>
<tr>
<td>a</td>
<td>^</td>
<td>a circumflex à</td>
</tr>
<tr>
<td>a</td>
<td>e</td>
<td>ae diphthong æ</td>
</tr>
<tr>
<td>c</td>
<td>,</td>
<td>c cedilla ç</td>
</tr>
<tr>
<td>c</td>
<td>/</td>
<td>cent sign ¢</td>
</tr>
<tr>
<td>c</td>
<td>o</td>
<td>copyright sign ©</td>
</tr>
<tr>
<td>d</td>
<td>-</td>
<td>eth ð</td>
</tr>
<tr>
<td>e</td>
<td>'</td>
<td>e grave è</td>
</tr>
<tr>
<td>e</td>
<td>'</td>
<td>e acute è</td>
</tr>
<tr>
<td>e</td>
<td>&quot;</td>
<td>e diaeresis ë</td>
</tr>
<tr>
<td>e</td>
<td>^</td>
<td>e circumflex ê</td>
</tr>
<tr>
<td>i</td>
<td>'</td>
<td>i grave i</td>
</tr>
<tr>
<td>i</td>
<td>'</td>
<td>i acute í</td>
</tr>
<tr>
<td>i</td>
<td>&quot;</td>
<td>i diaeresis î</td>
</tr>
<tr>
<td>i</td>
<td>^</td>
<td>i circumflex î</td>
</tr>
<tr>
<td>n</td>
<td>~</td>
<td>n tilde ñ</td>
</tr>
<tr>
<td>o</td>
<td>'</td>
<td>o grave ó</td>
</tr>
<tr>
<td>o</td>
<td>'</td>
<td>o acute ô</td>
</tr>
<tr>
<td>o</td>
<td>/</td>
<td>o slash ø</td>
</tr>
<tr>
<td>o</td>
<td>&quot;</td>
<td>o diaeresis ö</td>
</tr>
<tr>
<td>o</td>
<td>^</td>
<td>o circumflex ô</td>
</tr>
</tbody>
</table>
**TABLE 4–1** Common Latin-1 Compose Sequences  

<table>
<thead>
<tr>
<th>Press and Release</th>
<th>Press and Release</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>~</td>
<td>o tilde ó</td>
</tr>
<tr>
<td>s</td>
<td>s</td>
<td>German double s ß</td>
</tr>
<tr>
<td>t</td>
<td>h</td>
<td>thorn þ</td>
</tr>
<tr>
<td>u</td>
<td>'</td>
<td>u grave ū</td>
</tr>
<tr>
<td>u</td>
<td>'</td>
<td>u acute ū</td>
</tr>
<tr>
<td>u</td>
<td>=</td>
<td>u diaeresis ü</td>
</tr>
<tr>
<td>u</td>
<td>^</td>
<td>u circumflex ū</td>
</tr>
<tr>
<td>y</td>
<td>'</td>
<td>y acute y</td>
</tr>
<tr>
<td>y</td>
<td>=</td>
<td>y diaeresis ſ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>broken bar †</td>
</tr>
</tbody>
</table>

The following table lists the Common Latin-2 and Latin-4 Compose Sequences.

**TABLE 4–2** Common Latin-2 Compose Sequences

<table>
<thead>
<tr>
<th>Press and Release</th>
<th>Press and Release</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>' '</td>
<td>ogonek</td>
</tr>
<tr>
<td>u</td>
<td>' '</td>
<td>breve</td>
</tr>
<tr>
<td>v</td>
<td>' '</td>
<td>caron</td>
</tr>
<tr>
<td>&quot;</td>
<td>' '</td>
<td>double acute</td>
</tr>
<tr>
<td>A</td>
<td>a</td>
<td>A ogonek</td>
</tr>
<tr>
<td>A</td>
<td>u</td>
<td>A breve</td>
</tr>
<tr>
<td>C</td>
<td>'</td>
<td>C acute</td>
</tr>
<tr>
<td>C</td>
<td>v</td>
<td>C caron</td>
</tr>
<tr>
<td>D</td>
<td>v</td>
<td>D caron</td>
</tr>
<tr>
<td>D</td>
<td>'</td>
<td>D stroke</td>
</tr>
<tr>
<td>E</td>
<td>v</td>
<td>E caron</td>
</tr>
</tbody>
</table>
### TABLE 4–2  Common Latin-2 Compose Sequences  (continued)

<table>
<thead>
<tr>
<th>Press and Release</th>
<th>Press and Release</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>a</td>
<td>E ogonek</td>
</tr>
<tr>
<td>L</td>
<td>’</td>
<td>L acute</td>
</tr>
<tr>
<td>L</td>
<td>-</td>
<td>L stroke</td>
</tr>
<tr>
<td>L</td>
<td>&gt;</td>
<td>L caron</td>
</tr>
<tr>
<td>N</td>
<td>’</td>
<td>N acute</td>
</tr>
<tr>
<td>N</td>
<td>v</td>
<td>N caron</td>
</tr>
<tr>
<td>O</td>
<td>&gt;</td>
<td>O double acute</td>
</tr>
<tr>
<td>S</td>
<td>’</td>
<td>S acute</td>
</tr>
<tr>
<td>S</td>
<td>v</td>
<td>S caron</td>
</tr>
<tr>
<td>S</td>
<td>,</td>
<td>S cedilla</td>
</tr>
<tr>
<td>R</td>
<td>’</td>
<td>R acute</td>
</tr>
<tr>
<td>R</td>
<td>v</td>
<td>R caron</td>
</tr>
<tr>
<td>T</td>
<td>v</td>
<td>T caron</td>
</tr>
<tr>
<td>T</td>
<td>,</td>
<td>T cedilla</td>
</tr>
<tr>
<td>U</td>
<td>*</td>
<td>U ring above</td>
</tr>
<tr>
<td>U</td>
<td>&gt;</td>
<td>U double acute</td>
</tr>
<tr>
<td>Z</td>
<td>’</td>
<td>Z acute</td>
</tr>
<tr>
<td>Z</td>
<td>v</td>
<td>Z caron</td>
</tr>
<tr>
<td>Z</td>
<td>.</td>
<td>Z dot above</td>
</tr>
<tr>
<td>k</td>
<td>k</td>
<td>kra</td>
</tr>
<tr>
<td>A</td>
<td>¯</td>
<td>A macron</td>
</tr>
<tr>
<td>E</td>
<td>¯</td>
<td>E macron</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>E dot above</td>
</tr>
<tr>
<td>G</td>
<td>,</td>
<td>G cedilla</td>
</tr>
<tr>
<td>I</td>
<td>¯</td>
<td>I macron</td>
</tr>
<tr>
<td>I</td>
<td>~</td>
<td>I tilde</td>
</tr>
</tbody>
</table>
TABLE 4–2  Common Latin-2 Compose Sequences  (continued)

<table>
<thead>
<tr>
<th>Press and Release</th>
<th>Press and Release</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>a</td>
<td>I ogonek</td>
</tr>
<tr>
<td>K</td>
<td>,</td>
<td>K cedilla</td>
</tr>
<tr>
<td>L</td>
<td>,</td>
<td>L cedilla</td>
</tr>
<tr>
<td>N</td>
<td>,</td>
<td>N cedilla</td>
</tr>
<tr>
<td>O</td>
<td>_</td>
<td>O macron</td>
</tr>
<tr>
<td>R</td>
<td>,</td>
<td>R cedilla</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>~</td>
<td>U tilde</td>
</tr>
<tr>
<td>U</td>
<td>a</td>
<td>U ogonek</td>
</tr>
<tr>
<td>U</td>
<td>_</td>
<td>U macron</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>Eng</td>
</tr>
<tr>
<td>a</td>
<td>_</td>
<td>a macron</td>
</tr>
<tr>
<td>e</td>
<td>_</td>
<td>e macron</td>
</tr>
<tr>
<td>e</td>
<td>.</td>
<td>e dot above</td>
</tr>
<tr>
<td>g</td>
<td>,</td>
<td>g cedilla</td>
</tr>
<tr>
<td>i</td>
<td>_</td>
<td>i macron</td>
</tr>
<tr>
<td>i</td>
<td>~</td>
<td>i tilde</td>
</tr>
<tr>
<td>i</td>
<td>a</td>
<td>i ogonek</td>
</tr>
<tr>
<td>k</td>
<td>,</td>
<td>k cedilla</td>
</tr>
<tr>
<td>l</td>
<td>,</td>
<td>l cedilla</td>
</tr>
<tr>
<td>n</td>
<td>,</td>
<td>n cedilla</td>
</tr>
<tr>
<td>o</td>
<td>_</td>
<td>o macron</td>
</tr>
<tr>
<td>r</td>
<td>,</td>
<td>r cedilla</td>
</tr>
<tr>
<td>t</td>
<td></td>
<td></td>
</tr>
<tr>
<td>u</td>
<td>~</td>
<td>u tilde</td>
</tr>
<tr>
<td>u</td>
<td>a</td>
<td>u ogonek</td>
</tr>
</tbody>
</table>
### TABLE 4–2  Common Latin-2 Compose Sequences  
(continued)

<table>
<thead>
<tr>
<th>Press and Release</th>
<th>Press and Release</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>_</td>
<td>u macron</td>
</tr>
<tr>
<td>n</td>
<td>n</td>
<td>eng</td>
</tr>
</tbody>
</table>

The following table lists the Common Latin-5 Compose Sequences.

### TABLE 4–3  Common Latin-5 Compose Sequences

<table>
<thead>
<tr>
<th>Press and Release</th>
<th>Press and Release</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>u</td>
<td>G breve</td>
</tr>
<tr>
<td>I</td>
<td>.</td>
<td>I dot above</td>
</tr>
<tr>
<td>g</td>
<td>u</td>
<td>g breve</td>
</tr>
<tr>
<td>i</td>
<td>.</td>
<td>i dotless</td>
</tr>
</tbody>
</table>

Any Compose Sequences already described do not re-appear in this table.

The following table lists the Common Latin-9 Compose Sequences.

### TABLE 4–4  Common Latin-9 Compose Sequences

<table>
<thead>
<tr>
<th>Press and Release</th>
<th>Press and Release</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>e</td>
<td>Diphthong oe</td>
</tr>
<tr>
<td>O</td>
<td>E</td>
<td>Diphthong OE</td>
</tr>
<tr>
<td>Y</td>
<td>´</td>
<td>Y diaeresis</td>
</tr>
</tbody>
</table>

### Cyrillic Input Mode

To switch to Cyrillic input mode, either press `<Compose> <c> <c> at your keyboard, or press the left-most mouse button at the status area of your application and select “[Cyrillic]” from the Input Mode Selection Window.

The input mode is displayed at the bottom left corner of your GUI application.
After you switch to Cyrillic input mode, you cannot enter English or European text. To switch back to the English/European input mode, type <Control> + <Space> from your keyboard, or select “[English/European]” input mode from the Input Mode Selection Window by using your mouse. The Russian keyboard layout appears in the following figure.

![Russian Keyboard Layout](image)

**Figure 4–1** Tabbing Behavior

You can also switch into other input modes by typing the corresponding input mode switch key sequence.

**Greek Input Mode**

To switch to Greek input mode, either press Compose <g> <g> at your keyboard, or press the left-most mouse button at the status area of your application and select “[Greek]”, from the Input Mode Selection Window.

The input mode is displayed at the left bottom corner of your GUI application.

![Greek Keyboard Layout](image)
After you switch to Greek input mode, you cannot enter English or European text. To switch back to the English/European input mode, type <Control> + <Space> from your keyboard, or select “[English/European]” input mode from the Input Mode Selection Window by using your mouse. The Greek keyboard layouts appear in the following two figures.

**Figure 4–2 Tabbing Behavior**

<table>
<thead>
<tr>
<th>Alpha</th>
<th>Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>(</td>
<td>)</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>_</td>
<td>=</td>
</tr>
<tr>
<td>`</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>`</td>
</tr>
<tr>
<td>~</td>
<td>`</td>
</tr>
<tr>
<td>@</td>
<td>^</td>
</tr>
<tr>
<td>#</td>
<td>&amp;</td>
</tr>
<tr>
<td>$</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4–3 Tabbing Behavior**

Arabic Input Mode

To switch to Arabic input mode, type <Compose> <a> <r> from your current input mode. The input mode is displayed at the left bottom corner of your GUI application. After you switch to the Arabic input mode, you have to switch back to English/European input mode to enter English/European characters by typing <Control> and <Space> together.
You can also switch into other input modes by either typing the corresponding input mode switch key sequence from your keyboard, or selecting an input mode from the Input Mode Selection Window by using your mouse.

![Arabic Input Mode](image)

*Figure 4–4   Tabbing Behavior*

**Hebrew Input Mode**

To switch into Hebrew input mode, type `<Compose> <h> <h>` from your current input mode. The input mode is displayed at the bottom left corner of your GUI application. You can also switch into the Hebrew input mode by pressing the left-most mouse button at the status area of your application and then selecting “[Hebrew]” from the Input Mode Selection Window.

After you have switched into the Hebrew input mode, you have to switch back to the English/European input mode to enter English/European characters. To switch your input mode, you can either type the corresponding input mode switch key sequence of your next input mode from your keyboard, or select an input mode from the Input Mode Selection Window by using your mouse. The Hebrew keyboard layout is shown in the following figure:
Thai Input Mode

To switch into Thai input mode, type `<Compose>` `<t>` `<t>` from your current input mode. The input mode displays at the left bottom corner of your GUI application.

After you have switched into the Thai input mode, you have to switch back to English/European input mode to enter English/European characters. To switch your input mode, either type the corresponding input mode switch key sequence of your next input mode from your keyboard, or select an input mode from the Input Mode Selection Window by using your mouse. The Thai keyboard layout is shown in the following figure:
Unicode Hexadecimal and Octal Code Input Method Input Modes

To switch into the Unicode hexadecimal code input method input mode, type <Compose> <u> <h> from your current input mode. You can also select “[Unicode Hex]” from the Input Mode Selection Window by using your mouse. The input mode is displayed at the left bottom corner of your application.

If you prefer the octal number system, you can also switch into the Unicode octal code input method input mode by typing <Compose> <u> <o> from your current input mode or by selecting “[Unicode Octal]” from the Input Mode Selection Window.

To use these input mode, you need to know about either the hexadecimal or the octal code point values of the characters. Refer to The Unicode Standard, Version 3.0 for the mapping between code point values and characters. To input a character, type four hexadecimal digits if you are in the Unicode hexadecimal code input method input mode, for instance, 00a1 for Inverted Exclamation Mark, 03b2 for Greek Small Letter Beta, ac00 for a Korean Hangul Syllable KA, 30a2 for Japanese Katakana Letter A, 4e58 for a Unified Han character, and so on. Users can use both uppercase and lowercase letters of A, B, C, D, E, and, F for hexadecimal digits. If you prefer the octal number system instead of hexadecimal numbers, you can input octal digits, 0 to 7. If you mistype a digit or two, you can delete the digits by using the <Delete> key or the <Backspace> key.
Table Lookup Input Method Input Mode

To switch into table lookup input method input mode, type `<Compose> <l> <l>` from your current input mode. The input mode is displayed at the bottom left corner of your GUI application.
After you turn on the input mode, there is a lookup group window showing multiple groups of characters. You can choose any one of the groups to enter characters from the group. Once you select a group, there will be the second lookup window showing multiple candidates of available Unicode characters belonging to the group of your choice. You can choose any one of the candidates by moving your pointer and clicking...
the left button on your mouse. You can also select any one of the candidates by choosing a left-hand-side letter associated with each of the candidates.

You can also see the next set of candidates by typing <Control> and <n> keys together. Similarly, to see the previous set of candidates, type the <Control> and <p> keys together. The <n> stands for ‘next’ and the <p> stands for ‘previous’.

After you are finished using the current input mode, you can switch into another input mode by typing a corresponding input mode switch key sequence.

Japanese Input Mode

To switch into the Japanese input mode, type either <Compose> <j> <a> from your keyboard or select “[ Japanese ]” from the Input Mode Selection Window by using your mouse. The input mode is displayed at the left bottom corner of your application. The following figure shows a Japanese input method mode of ATOK12:

![Japanese Input Method Mode](image)

To use the native Japanese input system, you need to install one or more of Japanese locales on your system. Once you install the Japanese locales, you will be able to use any one of native Japanese input systems like ATOK12, ATOK8, Wnn6, or cs00.


Korean Input Mode

To switch into the Korean input mode, type either <Compose> <k> <o> from your keyboard, or select “[ Korean ]” from the Input Mode Selection Window by using your mouse. The input mode is displayed at the left bottom corner of your application. The following figure shows Phonetic Hangul input method which is one of many native Korean input methods available.
To have the native Korean input system, you need to install one or more Korean locale on your system. Once you install the Korean locale, you will be able to use the native Korean input system. For more details on how to use the Korean Input System, refer to “Korean Solaris User’s Guide”.

Simplified Chinese Input Mode

To switch input Simplified Chinese input mode, type either <Compose> <s> <c> from your keyboard, or select “[ S-Chinese ]” from the Input Mode Selection Window by using your mouse. The input mode is displayed at the left bottom corner of your application. The following figure shows New Pin Yin input method which is one of many native Simplified Chinese input methods available.

To use the native Simplified Chinese input system, you need to install one or more Simplified Chinese locales on your system. Once you install the Simplified Chinese locales, you will be able to use the native Simplified Chinese input system. For more details on how to use Simplified Chinese Input System, refer to “Simplified Chinese Solaris User’s Guide.”

Traditional Chinese Input Mode

To switch input Traditional Chinese input mode, type either <Compose> <t> <c> from your keyboard or select “[ T-Chinese ]” from the Input Mode Selection Window by using your mouse. The input mode is displayed at the left bottom corner of your application. The following figure shows the TsangChieh input method which is one of many native Traditional Chinese input methods available.
To have the native Traditional Chinese input system, you need to install one or more of
Traditional Chinese locales at your system. Once you install the Traditional Chinese
locales, you will be able to use the native Traditional Chinese input system. For more
details on how to use the Traditional Chinese Input System, refer to “Traditional
Chinese Solaris User’s Guide”.

Input Mode Switch Key Sequence Summary

Users can switch from one input mode to another without any restrictions. The
following table shows the input mode switch key sequences for each input mode.

<table>
<thead>
<tr>
<th>Input Mode</th>
<th>Key Sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>English/European</td>
<td>&lt;Control&gt; + &lt;Space&gt;</td>
</tr>
<tr>
<td>Cyrillic</td>
<td>&lt;Compose&gt; &lt;&lt; &lt;o&gt;</td>
</tr>
<tr>
<td>Greek</td>
<td>&lt;Compose&gt; &lt;g&gt; &lt;g&gt;</td>
</tr>
<tr>
<td>Arabic</td>
<td>&lt;Compose&gt; &lt;a&gt; &lt;r&gt;</td>
</tr>
<tr>
<td>Hebrew</td>
<td>&lt;Compose&gt; &lt;h&gt; &lt;h&gt;</td>
</tr>
<tr>
<td>Thai</td>
<td>&lt;Compose&gt; &lt;i&gt; &lt;i&gt;</td>
</tr>
<tr>
<td>Unicode hexadecimal code</td>
<td>&lt;Compose&gt; &lt;u&gt; &lt;h&gt;</td>
</tr>
<tr>
<td>Table lookup input method</td>
<td>&lt;Compose&gt; &lt;i&gt; &lt;i&gt;</td>
</tr>
<tr>
<td>Unicode octal code input</td>
<td>&lt;Compose&gt; &lt;u&gt; &lt;o&gt;</td>
</tr>
<tr>
<td>Japanese</td>
<td>&lt;Compose&gt; &lt;a&gt; &lt;a&gt;</td>
</tr>
<tr>
<td>Korean</td>
<td>&lt;Compose&gt; &lt;k&gt; &lt;o&gt;</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>&lt;Compose&gt; &lt;s&gt; &lt;s&gt;</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>&lt;Compose&gt; &lt;t&gt; &lt;c&gt;</td>
</tr>
</tbody>
</table>
System Environment

Locale Environment Variable

To use the \texttt{en\_US.UTF-8} locale environment, choose the locale first. Be sure you have the \texttt{en\_US.UTF-8} locale installed on your system.

\textbf{How to Use the \texttt{en\_US.UTF-8} Locale Environment}

1. In a TTY environment, choose the locale first, by setting the \texttt{LANG} environment variable to \texttt{en\_US.UTF-8}, as in the following C-shell example:

```
setenv LANG en_US.UTF-8
```

Make sure that other categories are not set (or are set to \texttt{en\_US.UTF-8}), since the \texttt{LANG} environment variable has a lower priority than other environment variables, such as \texttt{LC\_ALL}, \texttt{LC\_COLLATE}, \texttt{LC\_CTYPE}, \texttt{LC\_MESSAGES}, \texttt{LC\_NUMERIC}, \texttt{LC\_MONETARY} and \texttt{LC\_TIME} have at setting the locale. See the \texttt{setlocale(3C)} man page for more details about the hierarchy of environment variables.

To check current locale settings in various categories, use the \texttt{locale(1)} utility.

```
locale
LANG=en_US.UTF-8
LC\_CTYPE="en\_US.UTF-8"
LC\_NUMERIC="en\_US.UTF-8"
LC\_TIME="en\_US.UTF-8"
LC\_COLLATE="en\_US.UTF-8"
LC\_MONETARY="en\_US.UTF-8"
LC\_MESSAGES="en\_US.UTF-8"
LC\_ALL=
```

You can also start the \texttt{en\_US.UTF-8} environment from the CDE desktop. At the CDE login screen’s Options -> Language menu, choose \texttt{en\_US.UTF-8}.

TTY Environment Setup

Depending on the terminal and terminal emulator, such as \texttt{dtterm(1)} that you are using, you may need to push certain codeset-specific STREAMS modules onto your Streams.

For more information on STREAMS modules and streams in general, see the \textit{STREAMS Programming Guide}.

The following table shows STREAMS modules supported by the \texttt{en\_US.UTF-8} locale in the terminal environment:
TABLE 4–6  32-bit STREAMS Modules Supported by en_US.UTF-8

<table>
<thead>
<tr>
<th>32-bit STREAMS Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/kernel/strmod/u8lat1</td>
<td>Code conversion STREAMS module between UTF-8 and ISO 8859-1 (Western European)</td>
</tr>
<tr>
<td>/usr/kernel/strmod/u8lat2</td>
<td>Code conversion STREAMS module between UTF-8 and ISO 8859-2 (Eastern European)</td>
</tr>
<tr>
<td>/usr/kernel/strmod/u8koi8</td>
<td>Code conversion STREAMS module between UTF-8 and KOI8-R (Cyrillic)</td>
</tr>
</tbody>
</table>

The following table lists the 64–bit STREAMS Modules Supported by en_US.UTF-8.

TABLE 4–7  64–bit STREAMS Modules Supported by en_US.UTF-8

<table>
<thead>
<tr>
<th>64-bit STREAMS module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/kernel/strmod/sparcv9/u8lat1</td>
<td>Code conversion STREAMS module between UTF-8 and ISO 8859-1 (Western European)</td>
</tr>
<tr>
<td>/usr/kernel/strmod/sparcv9/u8lat2</td>
<td>Code conversion STREAMS module between UTF-8 and ISO 8859-2 (Eastern European)</td>
</tr>
<tr>
<td>/usr/kernel/strmod/sparcv9/u8koi8</td>
<td>Code conversion STREAMS module between UTF-8 and KOI8-R (Cyrillic)</td>
</tr>
</tbody>
</table>

Loading a STREAMS Module at Kernel

To load a STREAMS module at kernel, first become root:

```
    system% su
    Password: system#
```

To determine whether you are running a 64-bit Solaris or 32-bit Solaris system, use the isainfo(1) utility as follows:

```
    system# isainfo -v
    64-bit sparcv9 applications
    32-bit sparc applications
    system#
```
If the command returns this information, you are running the 64-bit Solaris system. If you are running the 32-bit Solaris system, the utility shows the following:

```
$ isainfo -v
32-bit sparc applications
$ system#
```

Use `modinfo(1M)` to be certain that your system has not already loaded the STREAMS module:

```
$ modinfo | grep u8lat1
```

If the STREAMS module, such as `u8lat1`, is already installed, the output looks as follows:

```
$ modinfo | grep u8lat1
 89 ff798000 4b13 18 1 u8lat1 (UTF-8 <--> ISO 8859-1 module)
$ system#
```

If the module is already installed, you don’t need to load it. However, if the module has not yet been loaded, use `modload(1M)` as follows:

```
$ modload /usr/kernel/strmod/u8lat1
```

This loads the 32-bit `u8lat1` STREAMS module at the kernel so you can push it onto a Stream. If you are running the 64-bit Solaris product, use `modload(1M)` as follows:

```
$ modload /usr/kernel/strmod/sparcv9/u8lat1
```

The STREAMS module is loaded at the kernel and you can now push it onto a Stream.

To unload a module from the kernel, use `modunload(1M)`, as shown below. In this example, the `u8lat1` module is being unloaded.

```
$ modinfo | grep u8lat1
 89 ff798000 4b13 18 1 u8lat1 (UTF-8 <--> ISO 8859-1 module)
$ system# modunload -i 89
```

### dtterm and Terminals Capable of Input and Output of UTF-8 Characters

Unlike in previous releases of the Solaris operating environment, the `dtterm(1)` and any other terminals that support input and output of the UTF-8 codeset do not need to have any other additional STREAMS module in their Stream. `ldterm(7M)` module is now codeset independent and supports Unicode/UTF-8 as well.
For the proper terminal environment setup for the Unicode locales, use the `stty(1)` utility as follows:

```
system% stty defeucw
```

Since `/usr/ucb/stty` is not internationalized, use `/bin/stty` instead.

Terminal Support for Latin-1, Latin-2, or KOI8-R

For terminals that support only Latin-1 (ISO 8859-1), Latin-2 (ISO 8859-2), or KOI8-R, you should have the following STREAMS configuration:

```
head <-> ttcompat <-> ldterm <-> u8lat1 <-> TTY
```

This configuration is only for terminals that support Latin-1. For Latin-2 terminals, replace the STREAMS module `u8lat1` with `u8lat2`. For KOI8-R terminals, replace the module with `u8koi8`.

Make sure you already have the STREAMS module loaded into the kernel.

To set up the STREAMS configuration shown above, use `strchg(1)`, as follows:

```
 system% cat > tmp/mystreams
ttcompat
ldterm
u8lat1
ptem
^D
system% strchg -f /tmp/mystreams
```

Be sure that you are either root or the owner of the device when you use `strchg(1)`. To see the current configuration, use `strchg(1)`, as follows:

```
 system% strconf
ttcompat
ldterm
u8lat1
ptem
pts
system%
```

To reset the original configuration, set the STREAMS configuration as follows:

```
 system% cat > /tmp/orgstreams
ttcompat
ldterm
u8lat1
ptem
system%
```

(continued)
Setting Terminal Options

To set up the UTF-8 text edit behavior on TTY, you must first set some terminal options using `stty(1)`, as follows:

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

Because `/usr/ucb/stty` is not yet internationalized, you should use `/bin/stty` instead.

You can also query the current settings using: `stty(1)` with the `-a` option, as shown below:

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

Saving the Settings in ~/.cshrc

Assuming the necessary STREAMS modules are already loaded with the kernel, you can save the following lines in your `.cshrc` file (C shell example) for convenience:

```
setenv LANG en_US.UTF-8
if ($?USER != 0 && $?prompt != 0) then
    cat >! /tmp/mystreams$$ << _EOF
    ttcompat
    u8euc
    ldterm
    eucu8
    ptem
    _EOF
    /bin/strchg -f /tmp/mystreams$$
    /bin/rm -f /tmp/mystreams$$
    /bin/stty cs8 -istrip defeucw
endif
```

With these lines in your `.cshrc` file, you do not have to type all of the commands each time. Note that the second `_EOF` should be in the first column of the file. You can also create a file called `mystreams` and save it so that `.cshrc` refers to `mystreams` instead of creating it whenever you start a C shell.
Code Conversions

The en_US.UTF-8 locale supports various code conversions among major codesets of several countries through iconv(1) and iconv(3).

In the Solaris 8 environment, the utility geniconvtbl enables user-defined code conversions. The user-defined code conversions created with the geniconvtbl utility can be used with both iconv(1) and iconv(3). For more detail on this utility, refer to geniconvtbl(1) and geniconvtbl(4) man pages.

The available fromcode and tocode names that can be applied to iconv(1) and iconv_open(3) are shown in the following table. For more details on iconv code conversion, see the iconv(1) and iconv_open(3), iconv(3), and iconv_close(3) man pages. For more information on available code conversions, see iconv_en_US.UTF-8(5).

Also see Appendix A.

UCS-2, UCS-4, UTF-16 are all fixed-width Unicode/ISO/IEC 10646 representation forms that recognizes Byte Order Mark (BOM) characters defined in the Unicode 3.0 and ISO/IEC 10646-1:1999 standards. Other forms, like UCS-2BE, UCS-4BE, and UTF-16BE, are all fixed-width Unicode/ISO/IEC 10646 representation forms that do not recognize the BOM character and also assume Big Endian byte ordering. Representation forms like UCS-2LE, UCS-4LE, UTF-16LE, on the other hand, assume Little Endian byte ordering. They also do not recognize the BOM character.

For associated scripts/languages of ISO 8859-* and KOI8-* see http://czyborra.com/charsets/iso8859.html.

Printing

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.
Starting with the next release of the Solaris environment, `xutops(1)` will be obsolete.

To use the utility, type the following:

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

You can also use the utility as a filter, since 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 lpl accepts only mp format files. This command line also installs the printer lpl on port /dev/ttya. See the `lpadmin(1M)` man page for more details.

```
system# lpadmin -p lpl -v /dev/ttya -I MP
system# accept lpl
system# enable lpl
```

Using `lpfilter(1M)`, you can add the utility for a filter as follows:

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

The command tells LP that a converter (in this case, `xutops`) is available through the filter description file named `pathname`. The `pathname` can 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 detail on `mp(1)`, refer to the `mp(1)` man page.
DtMail

As a result of increased coverage in scripts, Solaris 8 DtMail running in the en_US.UTF-8 locale supports various MIME character sets shown below.

- US-ASCII (7-bit US ASCII)
- UTF-8 (UCS Transmission Format 8 of Unicode)
- UTF-7 (UCS Transmission Format 7 of Unicode)
- ISO-8859-1 (Latin-1)
- ISO-8859-2 (Latin-2)
- ISO-8859-3 (Latin-3)
- ISO-8859-4 (Latin-4)
- ISO-8859-5 (Latin/Cyrillic)
- ISO-8859-6 (Latin/Arabic)
- ISO-8859-7 (Latin/Greek)
- ISO-8859-8 (Latin/Hebrew)
- ISO-8859-9 (Latin-5)
- ISO-8859-10 (Latin-6)
- ISO-8859-15 (Latin-9)
- KOI8-R (Cyrillic)
- ISO-2022-CN (Simplified Chinese)
- ISO-2022-TW (Traditional Chinese)
- ISO-8859–13 (Latin-7/Baltic)
- ISO-8859–14 (Latin-8/Celtic)
- KOI8–U (Cyrillic/Ukrainian)
- Shift_JIS (Japanese in Shift JIS)
- BIG5 (Traditional Chinese in BIG5)
- GB2312 (Simplified Chinese in EUC)
- TIS-620 (Thai)
- UTF-16 (UCS Transmission Format 16 of Unicode)
- UTF-16BE (UTF-16 Big-Endian of Unicode)
- UTF-16LE (UTF-16 Little-Endian of Unicode)

This support allows users to view virtually any kind of email encoded in various MIME character sets from any region of the world in a single instance of DtMail. The

Overview of en_US.UTF-8 Locale Support 115
decoding of received email is done by DtMail, which looks at the MIME character set and content transfer encoding provided with the email.

However, in case of sending, you need to specify a MIME character set that is understood by the recipient mail user agent (in other words, mail client), unless you want to use the default MIME character set provided by the en_US.UTF-8 locale. To switch the character set of out-going email, at the ‘New Message’ window, type either <CONTROL> + <y> or click the “Format” menu button and then click again on the “Change Char Set” button by using your mouse. The next available character set name displays at left bottom corner on top of the Send button.

If your email message header or message body contains characters that cannot be represented by the MIME charset specified, the system automatically switches the MIME character set to the UTF-8 that can represent any character.

If your message contains characters from the 7-bit US-ASCII character set only, your email’s default MIME character set is US-ASCII. Any mail user agent can interpret such email messages without any loss of characters or information.

If your message contains characters from a mixture of scripts, your email’s default MIME character set is UTF-8. Any 8-bit characters of UTF-8 are encoded with Quoted-Printable encoding. For more detail on MIME, registered MIME charsets, and Quoted-Printable encoding, refer to RFC 2045, 2046, 2047, 2048, 2049, 2279, 2152, 2237, 1922, 1557, 1555, and 1489.
Programming Environment

Appropriately, internationalized applications should automatically enable the en_US.UTF-8 locale, but proper FontSet/XmFontList definitions in the application’s resource file are required.


FontSet Used with X Applications

The en_US.UTF-8 locale in the Solaris 8 environment supports fonts for the following character sets.
Because the Solaris 8 environment supports the CDE desktop environment, each character set has a guaranteed sets of fonts.

The following is a list of the Latin-1 fonts that are supported in the Solaris 8 product:

- `dt-interface system-medium-r-normal-xxs sans utf-10-100-72-72-p-59-iso8859-1`
- `dt-interface system-medium-r-normal-xs sans utf-12-120-72-72-p-71-iso8859-1`
- `dt-interface system-medium-r-normal-s sans utf-14-140-72-72-p-82-iso8859-1`
- `dt-interface system-medium-r-normal-m sans utf-17-170-72-72-p-97-iso8859-1`
- `dt-interface system-medium-r-normal-xl sans utf-20-200-72-72-p-114-iso8859-1`
- `dt-interface system-medium-r-normal-xxl sans utf-24-240-72-72-p-137-iso8859-1`

For information on CDE common font aliases, including `-dt-interface user-*` and `-dt-application-*` aliases, see Common Desktop Environment: Internationalization Programmer’s Guide.

In the `en_US.UTF-8` locale, `utf` is also supported as a common font alias. A font set for an application should have a collection of fonts that contains each of the character sets, as in the following example:

(continued)
fs = XCreateFontSet(display,
    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-1,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-2,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-4,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-5,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-6,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-7,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-8,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-9,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-15,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-big5-1,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-jisx0208.1983-0,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-jisx0201.1976-0,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-ksc5601.1992-3,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-gb2312.1980-0,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-tis620.2533-0,
    -dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-unicode-fontspecific", &missing_ptr, &missing_count, &def_string);

Or, put more simply:

fs = XCreateFontSet(display,
    "-dt-interface system-medium-r-normal-s*utf*",
    &missing_ptr, &missing_count, &def_string);

XmFontList Definition as CDE/Motif Applications

As with FontSet definition, the XmFontList resource definition of an application should also include each font of the character sets that the locale supports.

CODE EXAMPLE 4–1  XmFontList Definition for the en_US.UTF-8 Locale

*fontList:*

    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-1;"
    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-2;"
    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-4;"
    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-5;"
    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-6;"
    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-7;"
    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-8;"
    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-9;"
    "-dt-interface system-medium-r-normal-s*utf*-*-*-*-*-*-*-iso8859-15;"
Or, put more simply:

```
*XmPushButton.fontList:
  -dt-interface system-medium-r-normal-*s*utf*;
```

For more details on the XmFontList and the XmNFontList, refer to the `XmFontList(3X)` man page, OSF/Motif Programmer’s Guide, and the resource section of each Motif widget in the OSF/Motif Programmer’s Reference Manual.
X/DPS

The X Window System has been extended with the X Display PostScript system (often described as X/DPS). It uses application-callable libraries on the client side and corresponding extensions on the X server side.

Internationalization and localization issues using Adobe System’s PostScript are documented in several books from Adobe Systems, Inc.:


This set of books is essential for successfully developing PostScript applications.

The *PostScript Language Reference Manual (Second Edition)* is the standard reference work for PostScript. It is the definitive documentation of every operator, Display PostScript (DPS), Level 1, and Level 2. The book covers the fundamentals of PostScript as a device-independent printing language. The special capabilities for handling fonts and characters in PostScript are explained. The book’s Appendix E also explains standard character sets and encoding vectors. It discusses the organization of fonts that are built into interpreters or supplied from other sources.

*Programming the Display PostScript System with X* is for application developers who are working with X Windows and Display PostScript. The book documents how to write applications that use Display PostScript to produce information for the screen display and the printer output. It describes coding techniques in detail.
Localization Resource Category

The localization resource category specifies which natural language (for example, English or Japanese) is supported. This category is made up of dictionaries that contain the keys Language, Country, CharSet, and others. These keys are in the %Console% device parameter set.

“<</Language/EN /Country/U.S. /CharSet/ISO-646-ISV>>
“<</Language/JA /Country null /CharSet/JIS--...>>”

In the example with Japanese, the null value shows that no dialect was selected for Japanese.

Unique names should be used for each item in the localization resource category.

Information on Language Interpreters

Page Description Language (PDL) interpreters can be assigned to a PostScript product. An application or printer driver uses the PDL resource category to see which PDL interpreter has been assigned.

Control languages can also be assigned. An application or printer driver can use ControlLanguage to see which control languages are available on a PostScript product.

The PDL and ControlLanguage resource categories are available.

The PDL and ControlLanguage resource categories are made up of key/value pairs. See the Adobe PostScript documentation for more information.
The Common Desktop Environment (CDE) is the standard GUI desktop interface for Solaris 8. Not only is it the user’s main interface to the system, but it is also the interface in which many of the user’s locale settings are apparent. The German user sees a German interface; the French user sees a French interface.

The *Common Desktop Environment: Internationalization Programmer’s Guide* provides information for internationalizing the desktop to enable applications to support various languages and cultural conventions in a consistent user interface.

### Overview of CDE

CDE is fully internationalized so that any application can run using any locale that has been installed in the system. By keeping the language- and culture-dependent information separate from the application source code, the application does not need to be rewritten or recompiled to be marketed in different countries. Instead, the external information needs to be localized only to match the target language and customs.

The application interface has been standardized to allow functionality in any locale, including East Asia. Solaris 8 complies with the Portable Operating Systems Interface for Computer Environments (POSIX and X/Open specifications, which are also referred to as XPG4.2).

Each layer within the desktop must use the proper internationalization interface standards, which are described in the following sources:

Setting Locales

Most single-display clients operate in a single locale. This is set with the environment variable, usually $LANG or a set of LC_* environment variables, including $LC_CTYPE.

The LC_CTYPE category of the locale is used by the environment to identify the locale-specific features used at runtime. The fonts and input methods are determined by the LC_CTYPE category.

Xt programs that are enabled for internationalization are expected to call the XtSetLanguageProc() function (which calls setlocale() by default) to set the locale.

Integrating Fonts

Your application might be used by someone sitting at an X terminal or by someone at a remote workstation across a network. In these situations, the fonts available to the user’s X display from the X window server might be different than your application’s defaults, and some fonts might not be available.

The standard interface font names defined by CDE are guaranteed to be available on all CDE-compliant systems. These names do not specify actual fonts. Instead, they are aliases that each system vendor maps to its best available fonts. If you use only these font names in your application, you can be sure of getting the closest matching font on any CDE-compliant system.

See Common Desktop Environment: Programmer’s Overview “Standard Font Names” in Common Desktop Environment: Programmer’s Overview and also the CDE man pages DtStdInterfaceFontNames(5) and DtStdAppFontNames(5) for additional information.
Internationalization and CDE

Multiple environments can exist within a common open system to support various languages. Each of these is called a locale. A locale specifies the language, fonts, and customs to display data. CDE is fully internationalized so that any application can run in any locale. Any application should be code-set-independent and include support for any multibyte codeset.

All components are shipped as a single, worldwide executable. These support the U.S.A., Europe (Western and Eastern), Japan, Korea, Taiwan, Thailand, China, and the Middle East.

Matching Fonts to Character Sets

Various sets of fonts are used to render a locale’s characters. The specific font character set depends on the locale. This information should be in a locale-specific app-defaults file that contains font sets, fonts, and font lists.

XmFontSet specifies the locale-dependent fonts. The resource name is *fontSet. Fonts should not be specifically named. The resource name for XFontStruct is *font. Font lists contain lists of fonts and font sets. XFontList specifies the fonts.

Storage of Localized Text

Text strings in each language should be stored outside of the application and in directories that are identified by locale names. These strings are stored in three types of files: resource files, message catalogs, and private files.

Resource files and message catalogs are both files that deliver text strings. Resource files are compiled when they are loaded and message catalogs are precompiled and ready to be accessed. Any application should be codeset-independent and include support for any multibyte codeset. Private files can be databases of information that include some text strings. Ideally, text strings should be in resource files or message catalogs. If text strings are supplied in a private file, then you should develop a tool to extract and replace text strings.

Xlib Dependencies

X locale supports one or more of the locales defined by the host environment. Direct Xlib™ conforms to the American National Standards Institute (ANSI) C library and the locale announcement method is the setlocale() function. This function configures the locale operation of both the host C library and Xlib. The operation of Xlib is governed by the LC_CTYPE category; this is called the current locale. The XSupportsLocale() function is used to determine whether the current locale is supported by X.
Message Guidelines

Message guidelines should be developed and used to create a consistent format and style for text. Use clear and simple English so that all users, including those whose command of English is minimal, can understand every message. The book *Common Desktop Environment: Internationalization Programmer’s Guide* ends with a number of guidelines for producing clear, concise, translatable messages. Messages should explain the problem and suggest how to perform the action successfully. Comments to the translators should also be included that explain concepts, variables, and so forth. The book includes several lists of suggestions for the format style of the message catalogs and the style of the messages themselves.

Before sending out the message catalogs to be translated, it is useful to have the message catalogs translated from English into international English, that is, into a simplified English that can be easily translated into other languages. This speeds up the translation process, reduces the translator queries, and saves costs.

Internationalization and Distributed Networks

This section of the book explains the exchange of information between applications on different hosts. To transfer data, you must consider several parameters:

- The sender’s and receiver’s codeset
- Whether the protocol is 7-bit or 8-bit
- The type of interchange encoding allowed by the protocol

If the remote host uses the same codeset as the local host, and, if the protocol allows 8-bit data, no conversion is needed. If the protocol allows only 7-bit data, the 8-bit code points must be mapped onto 7-bit ASCII values. There are various strategies for conversion.

If the remote host’s codeset is different from that of the local host, the following two cases might apply. The conversion depends on the specific protocol. If the protocol allows 8-bit data, the protocol must specify which side makes the conversion. If the protocol allows only 7-bit data, a 7-bit interchange encoding is needed along with an identifying character repertoire.

Mail Interchange

With the increased use of the Internet and the ease of communicating with people around the world, an email message can be viewed on many platforms and dozens of locales. Standards for email interchange, however, are restricted by desktop machines
for which the default email standard is Simple Mail Transfer Protocol (SMTP), which supports only 7-bit transmission channels.

The sending agent converts the body of the message into a standard format and labels it as body. The receiving agent looks at the body and, if it supports the character encoding, converts the body into the local character set.

Because dtmail now uses the Language Conversion Library (LCL), dtmail has the capacity to support multibyte characters in both the subject line, the mail body, and in attachments. dtmail also has the ability to use characters of different encodings within the same mail, for example, SJIS and EUC encodings for the Japanese (ja) locale.

---

**OpenWindows**

Solaris 8 does not have any changes in OpenWindows with regard to internationalization. Applications that were developed for previous versions of Solaris will run in Solaris 8 without any changes.

The XView toolkit is not codeset independent. Applications that use the XView toolkit are not supported in non-EUC locales, such as ja_JP.PCK, en_US.UTF-8, or ko.UTF-8.

For information on international XView, see the internationalization portions of the XView Developer’s Notes.
Complex Text Layout

Complex Text Layout (CTL) extensions enable Motif APIs to support writing systems that require complex transformations between logical and physical text representations, such as Arabic, Hebrew, and Thai. CTL Motif provides character shaping, such as ligatures, diacritics, and segment ordering, and supports the transformation of static and dynamic text widgets. It also supports right-to-left and left-to-right text orientation and tabbing for dynamic text widgets. Because text rendering is handled through the rendition layer, other widget libraries can be easily extended to support CTL.

Overview of CTL Technology

To leverage the new features, users must have the Portable Layout Services (PLS) library and the appropriate language engine. CTL uses PLS as the interface to the language engine, and uses the language engine to transform text before it is rendered. Applications that support CTL must include additional resources, as described in the CTL documentation.

Specifically, XmCTL supports the following complex language shaping and reordering features provided by underlying locale-dependent PLS module transformations:

- Positional variation
- Ligation (many-to-one) and character composition (one-to-many)
- Diacritics
- Bi-directionality
- Symmetrical swapping
- Numeral shaping
- String validation
Overview of CTL Architecture

The CTL architecture is organized as shown in the diagram below. Dt Apps at the top of the stack employs Motif CTL functionality for rendering text. Motif in turn interfaces with locale-specific language engines using PLS, and performs transformations to support positional variation, numeral shaping, and so on.

The CTL architecture is built to support new languages by adding a new locale-specific engine. In other words, support for Thai and Vietnamese can be added without altering Motif or Dt Apps.

Changes in Motif to Support CTL Technology

XmDirection

The XmNlayoutDirection resource controls object layout. It interacts with the orientation value of the LayoutObject in the manner described below.

---

1. See section 11.3 of the Motif Programmer’s Guide (Release 2.1) for an overview of XmNlayoutDirection, and especially for a description of the interaction between XmStringDirection and XmNlayoutDirection.
Layout Direction

First, when XmNlayoutDirection is specified as XmDEFAULT_DIRECTION, then the widget's layout direction is set at creation time from the governing pseudo-XOC. In the case of dynamic text (XmText and XmTextField), the governing pseudo-XOC is the one that is associated with the XmRendition used for the widget. In the case of static text (XmList, XmLabel, XmLabelG), the layout direction is set from the first compound string component that specifies a direction. This specification happens in one of two ways:

- Directly, if the component is of type XmSTRING_COMPONENT_LAYOUT_PUSH or XmSTRING_COMPONENT_DIRECTION
- Indirectly, if the component is of type XmSTRING_COMPONENT_LOCALE_TEXT, XmSTRING_COMPONENT_WIDECHAR_TEXT, or XmSTRING_COMPONENT_TEXT, from the component's associated XmRendition's and associated LayoutObject.

Second, if XmNlayoutDirection is not specified as XmDEFAULT_DIRECTION, and the XmNlayoutModifier @ls orientation value is not specified explicitly in the layout modifier string, then the XmNlayoutDirection value is passed through to the XOC and its LayoutObject.

If both XmNlayoutDirection and the XmNlayoutModifier @ls orientation value are explicitly specified, then the behavior is mixed; the XmNlayoutDirection controls widget object layout, and the XmNlayoutModifier @ls orientation value controls layout transformations.

For More Information


This document describes a set of portable functions for handling context-dependent and bidirectional text transformations as a logical extension to the existing POSIX locale model. It is intended for system and application programmers who want to provide support for complex-text languages.

XmStringDirection

XmStringDirection is the data type used to specify the direction in which the system displays characters of a string.

The XmNlayoutDirection resource sets a default rendering direction for any compound string (XmString) that does not have a component specifying the direction of that string. Therefore, to set the layout direction, all you need to do is to set the appropriate value for the XmNlayoutDirection resource. You do not need to create
compound strings with specific direction components. When the application renders an XmString, it should look to see if the string was created with an explicit direction (XmStringDirection). If there is no direction component, the application should check the value of the XmNlayoutDirection resource for the current widget and use that value as the default rendering direction for the XmString.

See also XmRendition and XmDirection.

**XmRendition**

CTL adds the following new pseudo resources to XmRendition:

<table>
<thead>
<tr>
<th>Name</th>
<th>Class/Type</th>
<th>Access</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmNfontType</td>
<td>XmCFontType/XmFontType</td>
<td>CSF</td>
<td>XmAS_IS</td>
</tr>
<tr>
<td>XmNlayoutAttrObject</td>
<td>XmCLayoutAttrObject</td>
<td>CSF</td>
<td>NULL</td>
</tr>
<tr>
<td>XmNlayoutModifier</td>
<td>XmCLayoutModifier</td>
<td>CSF</td>
<td>NULL</td>
</tr>
</tbody>
</table>

**XmNfontType**

Specifies the type of the Rendition font object. For CTL, the value of this resource must be the XmFONT_IS_XOC value. If it is not, then the XmNlayoutAttrObject and XmNlayoutModifier resources are ignored.

When the value of this resource is XmFont_IS_XOC, and if the XmNfont resource is not specified, then at create time the value of the XmNfontName resource is converted into an XOC object in either the locale specified by the XmNlayoutAttrObject resource or the current locale. Furthermore, the value of the XmNlayoutModifier resource is passed through to any LayoutObject associated with the XOC.

**XmNlayoutAttrObject**

Specifies the layout AttrObject argument to be used to create the Layout Object associated with the XOC associated with this XmRendition. Refer to the Layout Services m_create_layout() specification for the
syntax and semantics of this string. (See the
description of XmNfontType above for an
explanation of the interaction between the Layout
Modifier Orientation output value and the
XmNlayoutDirection widget resource.)

XmNlayoutModifier

Specifies the layout values to be passed through
to the Layout Object associated with the XOC
associated with this XmRendition. For the
syntax and semantics of this string, see CAE
Specification.

Setting this resource using
XmRendition{Retrieve,Update} causes the
string to be passed through to the LayoutObject
associated with the XOC associated with this
Rendition. This is the mechanism for configuring
layout services dynamically. Unpredictable
behavior can result if the Orientation,
Context, TypeOfText, TextShaping, or
ShapeCharset are changed.

Additional Layout Behavior

The XmNlayoutModifier affects the layout behavior of the text associated with
the XmRendition. For example, if the layout default treatment of numerals is
NUMERALS_NOMINAL, the user can change to NUMERALS_NATIONAL by setting
XmNlayoutModifier to:

- @ls numerals=nominal:national, or
- @ls numerals=:national

The layout values can be classified into the following groups:

- Encoding description: TypeOfText, TextShaping, ShapeCharset (and locale
codeset)
  TypeOfText is essentially segment ordering and can be illustrated with opaque
  blocks. Modifying these values dynamically, through the rendition object is not
  usually meaningful, and almost certain to result in unpredictable behavior.

- Layout behavior: Orientation, Context, ImplicitAlg, Swapping, Numerals
  Orientation and Context should not be modified dynamically; you can safely
  modify ImplicitAlg, Swapping, and Numerals.

- Editing behavior: CheckMode
XmText, XmTextfield

Xm CTL extends XmText and XmTextfield by adding a parallel set of movement and deletion actions that operate visually, patterned after the Motif 2.0 CSText widget. The standard Motif 2.1 Text and TextField do not distinguish between logical and physical order: “next” and “forward” mean “to the right,” while “previous” and “backward” mean “to the left.” CSText, however, makes the proper distinction and defines a new set of actions with strictly physical names (for example, left-character(), delete-right-word(), and so on). All of these action routines are defined to be sensitive to the XmNlayoutDirection of the widget and to call the appropriate “next-” or “previous-” action. The Xm CTL extensions are slightly more complex than CSText’s in that they are sensitive not to the global orientation of the widget, but to the specific directionality of the physical characters surrounding the cursor, as determined by the pseudo-XOC (including neutral stabilization).

There is also a new resource to control selection policy, to provide a rendition tag, and to control alignment.

The set of new Xm CTL actions is roughly the cross product of \{Move, Delete, Kill\} by \{Left, Right\} by \{Character, Word\}, and is listed below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Class/Type</th>
<th>Access Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmNrenditionTag</td>
<td>XmCRenditionTag/XmRString</td>
<td>CSG XmFONTLIST_DEFAULT_TAG</td>
</tr>
<tr>
<td>XmNalignment</td>
<td>XmCAlignment/XmRAlignment</td>
<td>CSG XmALIGNMENT_BEGINNING</td>
</tr>
<tr>
<td>XmNeditPolicy</td>
<td>XmCEditPolicy/XmREditPolicy</td>
<td>CSG XmEDIT_LOGICAL</td>
</tr>
</tbody>
</table>

XmNrenditionTag  Specifies the rendition tag of the XmRendition (in the XmNrenderTable resource) to be used for this widget.

XmNalignment  Specifies the text alignment to be used in the widget. Only XmALIGNMENT_END and XmALIGNMENT_CENTER are supported.

XmNeditPolicy  Specifies the editing policy to be used for the widget, either XmEDIT_LOGICAL or XmEDIT_VISUAL. In the case of XmEDIT_VISUAL, selection, cursor movement, and deletion are in a visual style. Setting this resource also changes the translations for the standard keyboard movement and deletion.
events either to the new “visual” actions list below or to the existing logical actions.

Character Orientation Action Routines

All of the actions in the following list query the orientation of the character in the direction specified. If the direction is left-to-right, they call the corresponding next-/forward- or previous-/backward- variants:

- delete-left-character()
- delete-left-word()
- delete-right-character()
- delete-right-word()
- kill-left-character()
- kill-left-word()
- kill-right-character()
- kill-right-word()
- left-character()
- left-word()
- prev-cell()
- right-character()
- right-word()
- forward-cell()

Character Orientation Additional Behavior

The actions determine the orientation of characters by using the Layout Services transformation OutToInp and Property buffers (for the nesting level). The widget’s behavior is therefore dependent on the locale-specific transformation. If the information in the OutToInp or, especially, Property buffers is inaccurate, the widget might behave unexpectedly. Moreover, as the locale-specific modules fall outside of the scope of this specification, bi-directional editing behavior can differ from platform to platform for the same text, application, resource values, and LayoutObject configuration.

The visual mode actions result in a display of cell-based behavior. The logical mode actions result in logical character-based behavior. For example, the delete-right-character() operation deletes the input buffer characters that correspond to the display cell. That is, one input buffer character whole LayoutObject transformation “property” byte “new cell indicator” is 1, and all of the succeeding characters whose “new cell indicator” is 0.

2. For more information on the Property buffer, see the specification for m_transform_layout() in CAE Specification.
Similarly, for `backward-character()`, the insertion point is moved backward one character in the input buffer, and the cursor is redrawn at the visual location corresponding to the associated output buffer character. This means that several keystrokes are required to move across a composite display cell; the cursor does not actually change display location as the insertion point moves across input buffer characters whose “new cell indicator” is 0 (that is, diacritics or ligature fragments).

This means that deletion operates either from the logical/input buffer side, or from the display cell level of the physical/output side. There is no mode for a strict, physical character-by-character deletion, since there is no one-to-one correspondence between the input and output buffers. A given physical character can represent only a fragment of a logical character, for example.

**XmText Action Routines**

The `XmText` action routines are as follows:

```c
left-character(extend)
```

If the `XmNeditPolicy` is `XmEDIT_LOGICAL` and is called without arguments, it moves the insertion cursor back logically by a character. If the insertion cursor is at the beginning of the line, it moves the insertion cursor to the logical last character of the previous line, if one exists; otherwise, the insertion cursor position doesn’t change.

If the `XmNeditPolicy` is `XmEDIT_VISUAL`, then the cursor moves to the left of the cursor position. If the insertion cursor is at the beginning of the line, then it moves to the end character of the previous line, if one exists.

If it is called with an `extend` argument, it moves the insertion cursor, as in the case of no argument, and extends the current selection.

The `left-character()` action produces calls to the `XmNmotionVerifyCallback` procedures with the reason value `XmCR_MOVING_INSERT_CURSOR`. If called with an `extend` argument, this can produce calls to the `XmNgainPrimaryCallback` procedures. See the callback description.
left-word(extend)

If the XmNeditPolicy is XmEDIT_LOGICAL and is called without any arguments, and the insertion cursor is at the logical starting of the word, it moves the insertion cursor to the logical starting of the logical preceding word, if one exists; otherwise, the insertion cursor position doesn’t change. If the insertion cursor is in the word but not at the logical start of the word, it moves the insertion cursor to the logical start of the word. If the insertion cursor is at the logical start of the line, it moves the insertion cursor to the logical start of the logical last word in the previous line, if one exists; otherwise, the insertion cursor position doesn’t change.

If the XmNeditPolicy is XmEDIT_VISUAL and is called without arguments, it moves the insertion cursor to the first non-white-space character after the first white-space character to the left or after the beginning of the line. If the insertion cursor is already at the beginning of the word, it moves the insertion cursor to the beginning of the previous word. If the insertion cursor is already at the beginning of the line, it moves to the starting of the last word in the previous line.

If called with an argument of extend, it moves the insertion cursor, as in the case of no argument, and extends the current selection.

The left-word() action produces calls to the XmNmotionVerifyCallback procedures with the reason value XmCR_MOVING_INSERT_CURSOR. If it is called with an extend argument, this can produce calls to the XmNgainPrimaryCallback procedures. See the callback description in the Motif Programmer’s Reference for more information.
right-character(extend)

If the XmNeditPolicy is XmEDIT_LOGICAL and is called without any arguments, it moves the insertion cursor logically forward by a character. If the insertion cursor is at the logical end of the line, it moves the insertion cursor to the logical starting of the next line, if one exists.

If the XmNeditPolicy is XmEDIT_VISUAL, then the cursor moves to the right of the cursor position. If the insertion cursor is at the end of the line, it moves the insertion cursor to the starting of the next line, if one exists.

If called with an argument of extend, it moves the insertion cursor, as in the case of no argument and extends the current selection.

The right-character() action produces calls to the XmNmotionVerifyCallback procedures with the reason value XmCR_MOVING_INSERT_CURSOR. If called with extend argument, this can produce calls to the XmNgainPrimaryCallback procedures. See the callback description in the Motif Programmer’s Reference for more information.

right-word(extend)

If the XmNeditPolicy is XmEDIT_LOGICAL and is called without any arguments, it moves the insertion cursor to the logical starting of the logical succeeding word if one exists; otherwise, it moves to the logical end of the current word. If the insertion cursor is at the logical end of the line or in the logical last word of the line, it moves the cursor to the logical first word in the next line, if one exists; otherwise, it moves to the logical end of the current word.
If the XmNeditPolicy is XmEDIT_VISUAL and is called without arguments, it moves the insertion cursor to the first nonwhite space character after the first white space character to the right or after the end of the line.

If called with an argument of extend, it moves the insertion cursor, as in the case of no argument, and extends the current selection.

The left-word() action produces calls to the XmNmotionVerifyCallback procedures with the reason value XmCR_MOVING_INSERT_CURSOR. If called with extend argument, this can produce calls to the XmNgainPrimaryCallback procedures. See the callback description in the Motif Programmer’s Reference for more information.

delete-left-character() If the XmNeditPolicy is XmEDIT_LOGICAL, it is equivalent to delete-previous-char. If the XmNeditPolicy is XmEDIT_VISUAL, then in normal mode, if there is a non-null selection, it deletes the selection; otherwise it deletes the character left of the insertion cursor. In add mode, if there is a non-null selection, the cursor is not disjointed from the selection and XmNpendingDelete is set to True, it deletes the selection; otherwise, it deletes the character left of the insertion cursor. This can impact the selection.

The delete-left-character() action produces calls to the XmNmodifyVerifyCallback procedures with reason value XmCR_MODIFYING_TEXT_VALUE and the XmNvalueChangedCallback procedures with reason value XmCR_VALUE_CHANGED.
delete-right-character()

If the XmNeditPolicy is XmEDIT_VISUAL, it is equivalent to delete-next-character. If the XmNeditPolicy is XmEDIT_VISUAL, then in normal mode, if there is a non-null selection, it deletes the selection; otherwise, it deletes the character right of the insertion cursor. In add mode, if there is a non-null selection and the cursor is not disjointed from the selection, the XmNpendingDelete is set to True and the selection is deleted; otherwise, the character right of the insertion cursor is deleted. This can impact the selection.

The delete-right-character() action produces calls to the XmNmodifyVerifyCallback procedures with reason value XmCR_MODIFYING_TEXT_VALUE, and the XmNvalueChangedCallback procedures with reason value XmCR_VALUE_CHANGED.

delete-left-word()

If the XmNeditPolicy is XmEDIT_VISUAL, it is equivalent to delete-prev-word(). If the XmNeditPolicy is XmEDIT_LOGICAL, then in normal mode, if there is a non-null selection, it deletes the selection; otherwise, it deletes the characters left of the insertion cursor to the next space, punctuation character, tab, or beginning-of-line character. In add mode, if there is a non-null selection, the cursor is not disjointed from the selection; otherwise it deletes the characters left of the insertion cursor the right space, tab, or beginning-of-line character. In add mode, if there is a non-null selection, the cursor is not disjointed from the selection, the XmNpendingDelete is set to True, and the selection is deleted; otherwise, it deletes the character left of the insertion cursor, the right space, tab, or beginning
of new-line character. This can impact the selection.

**delete-right-word()**

If the XmNeditPolicy is XmEDIT_VISUAL, it is equivalent to delete-right-word(). If the XmNeditPolicy is XmEDIT_LOGICAL, then in normal mode, if there is a non-null selection, it deletes the selection; otherwise, it deletes the characters right of the insertion cursor to the next space, punctuation character, tab, or end-of-line character. In add mode, if there is a non-null selection, the cursor is not disjointed from the selection, XmNpendingDelete is set to True, and deletes the selection; otherwise, it deletes the characters right of the insertion cursor to the next space, tab, or end-of-line character. This can impact the selection.

**kill-left-character()**

If the XmNeditPolicy is XmEDIT_LOGICAL, it is equivalent to kill-prev-char. If the XmNeditPolicy is XmEDIT_VISUAL, then in normal mode, if there is a non-null selection, it deletes the selection; otherwise, it kills the character left of the insertion cursor and stores the character in the cut buffer. In add mode, if there is a non-null selection, the cursor is not disjointed from the selection, XmNpendingDelete is set to True, and deletes the selection; otherwise, it deletes the character left of the insertion cursor. This can impact the selection.

The kill-left-character() action produces calls to the XmNmodifyVerifyCallback procedures with the reason value XmCR_MODIFYING_TEXT_VALUE, and produces the XmNvalueChangedCallback procedures with the reason value XmCR_VALUE_CHANGED.
kill-right-character()

If the XmNeditPolicy is XmEDIT_VISUAL, it is equivalent to delete-next-character. If the XmNeditPolicy is XmEDIT_VISUAL, then in normal mode, if there is a non-null selection, it deletes the selection; otherwise, it deletes the character right of the insertion cursor and stores it in the cut buffer. In add mode, if there is a non-null selection, the cursor is not disjointed from the selection, the XmNpendingDelete is set to True and deletes the selection; otherwise, it deletes the character right of the insertion cursor. This can impact the selection.

The kill-right-character() action produces calls to the XmNmodifyVerify-Callback procedures with reason value XmCR_MODIFYING_TEXT_VALUE, and produces calls to the XmNvalue-ChangedCallback procedures with reason value XmCR_VALUE_CHANGED.

kill-left-word()

If the XmNeditPolicy is XmEDIT_VISUAL, it is equivalent to delete-prev-word(). If the XmNeditPolicy is XmEDIT_LOGICAL, then in normal mode, if there is a non-null selection, it deletes the selection; otherwise, it deletes the characters left of the insertion cursor to the next space, punctuation character, tab, or beginning-of-line character. In add mode, if there is a non-null selection, the cursor is not disjointed from the selection; otherwise it deletes the characters left of the insertion cursor the right space, tab, or beginning-of-line character and stores it in the cut buffer. In add mode, if there is a non-null selection, the cursor is not disjointed from the selection, XmNpendingDelete is set to True and deletes the selection; otherwise it deletes the characters left of
the insertion cursor, the right space, tab, or beginning of new-line character. This
can impact the selection.

```c
kill-right-word()
```

If the `XmNeditPolicy` is `XmEDIT_VISUAL`, it is equivalent
to `delete-right-word()`. If the `XmNeditPolicy` is `XmEDIT_LOGICAL`,
then in normal mode, if there is a
non-null selection, it deletes the
selection; otherwise, it deletes the
characters right of the insertion cursor
to the next space, tab, or end-of-line
character. In add mode, if there is
a non-null selection, the cursor is
not disjointed from the selection,
`XmNpendingDelete` is set to True, and
deletes the selection; otherwise, it deletes
the characters right of the insertion
cursor to the next space, punctuation
character, tab, or end-of-line character
and stores in the cut buffer. This can
impact the selection.

A few cell-based routines are implemented to support character composition, ligatures,
and diacritics. In other words, two or more characters might be represented by a single
glyph occupying one presentation cell.

The `XmText` cell action routines are as follows:

```c
prev-cell(extend)
```

Moves the insertion cursor back one cell. If the `XmNeditPolicy` is `XmEDIT_LOGICAL`, then
the insertion cursor is moved to the start of the cell that precedes the current cell logically,
if one exists; otherwise, it moves to the start of the current cell.

If the `XmNeditPolicy` is `XmEDIT_VISUAL`, then the cursor moves to the start of cell
to the left of the cursor, if one exists.

The `prev-cell()` action produces calls
to the `XmNmotionVerifyCallback`
procedures with the reason value
`XmCR_MOVING_INSERT_CURSOR`. If called with
an extend argument, this can produce calls to the
`XmNgainPrimaryCallback` procedures. See the
callback description in the `Motif Programmer's Reference` for more information.
forward-cell(extend) Moves the insertion cursor to the start of the logical next cell, if one exists; otherwise it moves it to the end of the cell. If the XmNeditPolicy is XmEDIT_LOGICAL, then the cursor moves forward one cell.

If the XmNeditPolicy is XmEDIT_VISUAL, then the cursor moves to the start of the cell to the right of the cursor position, if one exists; otherwise it moves to the end of the current cell. The forward-cell() action produces calls to the XmNmotionVerifyCallback procedures with the reason value XmCR_MOVING_INSERT_CURSOR. If called with an extend argument, this can produce calls to the XmNgainPrimaryCallback procedures. See the callback description in the Motif Programmer’s Reference for more information.

XmTextFieldGetLayoutModifier

Purpose

A TextField function that returns the layout modifier string that reflects the state of the layout object tied to its rendition.

Synopsis

#include <Xm/TextF.h>
String XmTextFieldGetLayoutModifier(Widget widget)

Description

XmTextFieldGetLayoutModifier accesses the value of the current layout object settings of the rendition associated with the widget. When the layout object modifier values are changed using a convenience function, the XmTextFieldGetLayoutModifier function returns the complete state of the layout object, not only the changed values.

Return Value

Returns the layout object modifier values in the form of a String value.
Related Information

XmTextField

---

**XmTextGetLayoutModifier**

**Purpose**

A Text function that returns the layout modifier string that reflects the state of the layout object tied to its rendition.

**Synopsis**

```c
#include <Xm/Text.h>
String XmTextGetLayoutModifier(Widget widget)
```

**Description**

`XmTextGetLayoutModifier` accesses the value of the current layout object settings of the rendition associated with the widget. When the layout object modifier values are changed using a convenience function, the `XmTextGetLayoutModifier` function returns the complete state of the layout object, not just the changed values.

**Return Value**

Returns the layout object modifier values in the form of a String value.

**Related Information**

XmText

---

**XmTextFieldSetLayoutModifier**

**Purpose**

A TextField function that sets the layout modifier values, which changes the behavior of the layout object tied to its rendition.
Synopsis

#include <Xm/TextF.h>
void XmTextFieldSetLayoutModifier(Widget widget, string layout_modifier)

Description

XmTextFieldSetLayoutModifier modifies the layout object settings of a rendition associated with the widget. When the layout object modifier values are set using this convenience function, only the attributes specified in the input parameter are changed; the rest of the attributes remain untouched.

Related Information

XmTextField

XmTextSetLayoutModifier

Purpose

A Text function that sets the layout modifier values, which changes the behavior of the layout object tied to its rendition.

Synopsis

#include <Xm/Text.h>
void XmTextSetLayoutModifier(Widget widget, string layout_modifier)

Description

XmTextSetLayoutModifier modifies the layout object settings of a rendition associated with the widget. When the layout object modifier values are set using this convenience function, only the attributes specified in the input parameter are changed; the rest of the attributes are left untouched.

Related Information

XmText
XmStringDirectionCreate

Synopsis

#include <Xm/Xm.h>
XmString XmStringDirectionCreate(direction)
XmStringDirection direction

Description

XmStringDirectionCreate creates a compound string with a single component, a direction with the given value. On the other hand, the XmNlayoutDirection resource sets a default rendering direction for any compound string (XmString) that does not have a component specifying the direction for that string. Therefore, to set the layout direction, all you need to do is set the appropriate value for the XmNlayoutDirection resource. You need not create compound strings with specific direction components. When the application renders an XmString, it should look to see if the string was created with an explicit direction (XmStringDirection). If there is no direction component, the application should check the value of the XmNlayoutDirection resource for the current widget and use that value as the default rendering direction for the XmString.

Related Information

See also XmRendition, XmDirection.

UIL

The following table shows the UIL arguments:

<table>
<thead>
<tr>
<th>UIL Argument Name</th>
<th>Argument Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmNlayoutAttrObject</td>
<td>String</td>
</tr>
<tr>
<td>XmNlayoutModifier</td>
<td>String</td>
</tr>
<tr>
<td>XmNrenditionTag</td>
<td>String</td>
</tr>
</tbody>
</table>

Complex Text Layout 147
How to Develop CTL Applications

Layout Direction

The direction of a compound string is stored so that the data structure will be equally useful for describing text in left-to-right languages such as English, Spanish, French, and German, as well as for text in right-to-left languages, such as Hebrew and Arabic. In Motif applications, you can set the layout direction using the XmNlayoutDirection resource from the VendorShell or MenuShell. Manager and Primitive widgets (as well as Gadgets) also have an XmNlayoutDirection resource. The default value is inherited from the closest ancestor with the same resource.

In the case of an XmText widget, you must specify the vertical direction as well. Setting the layoutDirection to XmRIGHT_TO_LEFT will result in the string direction from right-to-left, but the cursor will move vertically down. If the vertical direction is important and you require top to bottom is alignment, be sure to specify XmRIGHT_TO_LEFT_TOP_TO_BOTTOM, which specifies that the components are laid out from right-to-left first and then top-to-bottom, and will result in the desired behavior.

Furthermore, the behavior of XmText and TextField widgets is influenced by the XmNalignment and XmNlayoutModifier resources of the XmRendition. These resources, in addition to XmNlayoutDirection, control the layout behavior of the Text widget. This can be illustrated using the example below.

The input string used in the illustration is:

A B ض و

The XmNlayoutModifier string @ls orientation= setting values for this illustration are shown below, in the left column.
Figure 7–1 Tabbing Behavior

As the illustration shows, XmNAlignment dictates whether the text is flush right or left in conjunction with the layout direction. On the other hand, XmNLayoutModifier breaks the text into segments and arranges them left-to-right or right-to-left, depending on the orientation value. In other words, if the XmNLayoutDirection is XmRIGHT_TO_LEFT, and the XmNAlignment value is XmALIGNMENT_BEGINNING, the string is flush right.

### Creating a Rendition

The following code creates an XmLabel whose XmNlabelString is of the type XmCHARSET_TEXT, using the Rendition whose tag is “ArabicShaped.” The Rendition is created with an XmNLayoutAttrObject of “ar” (corresponding to the locale name for the Arabic locale) and a layout modifier string that specifies for the output buffer a Numerals value of NUMERALSCONTEXTUAL and a ShapeCharset value of “unicode-3.0.”

The locale-specific layout module transforms its input text (in this example encoded in ISO 8859-6) in an output buffer of physical characters encoded using the 16-bit Unicode.
3.0 codeset. Because an explicit layout locale has been specified, this text is rendered properly independent of the runtime locale setting.

```c
int n;
Arg args[10];
Widget w;
 XmString labelString;
XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;

/* alef lam baa noon taa - iso8859-6 */
labelString = XmStringGenerate("\307\344\310\346\312", NULL
XmCHARSET_TEXT, "ArabicShaped");
w = XtVaCreateManagedWidget("a label", XmLabelWidgetClass, parent,
    XmNlabelString, labelString,
    XmNlabelType, XmSTRING,
    NULL);

n = 0;
XtSetArg(args[n], XmNfontName, "-*-*-medium-r-normal-*-24-**-**-**-**-**");
n++;
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
XtSetArg(args[n], XmNlayoutAttrObject, "ar"); n++;
XtSetArg(args[n], XmNlayoutModifier,
    "@ls numerals=:contextual, shapecharset=iso8859-6"); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args
    s, n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE_MERGE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
```

**Editing a Rendition**

The following code creates a TextField widget and a RenderTable with a single Rendition. Both the XmNlayoutAttrObject and XmNlayoutModifier pseudo resources have been left unspecified and therefore defaults to NULL. This means the LayoutObject associated with the Rendition belongs to the default locale, if one exists.

For this example to work properly, the locale must be set to one whose codeset is ISO 8859-6 and whose locale-specific layout module can support the IMPLICIT_BASIC algorithm. It then modifies the Rendition's LayoutObject's ImplicitAlg value through the Rendition's XmNlayoutModifier pseudo resource.

```c
int n;
Arg args[10];
Widget w;
 XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;

w = XmCreateTextField(parent, "text field", args, 0);

n = 0;
XtSetArg(args[n], XmNfontName, "-*-*-medium-r-normal-*-24-**-**-**-**-**");
n++;
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args
    s, n);
renderTable =
```
Creating a Render Table in a Resource File

Renditions and render tables can be specified in resource files. For a properly internationalized application, in fact, this is the preferred method. When the render tables are specified in a file, the program binaries are made independent of the particular needs of a given locale, and can be easily customized to local needs.

Render tables are specified in resource files with the following syntax:

```
resource_spec : [tag [, tag ] *]
```

where `tag` is some string suitable for the `XmNtag` resource of a rendition.

This line creates an initial render table containing one or more renditions as specified. The renditions are attached to the specified tags:

```
resource_spec [* .] rendition [* .] resource_name : value
```

The following examples illustrate the CTL resources related to `XmRendition` that can be set using resource files. The `fontType` must be set to `FONT_IS_XOC` for the layout object to take effect. The `layoutModifier` specified using `@ls` is passed on to the layout object by the rendition object.


Creating a Render Table in an Application

Before creating a render table, an application program must first have created at least one of the renditions that is part of the table. The `XmRenderTableAddRenditions`...
The following code creates a render table using a rendition created with XmNfontType set to XmFONT_IS_XOC.

```c
int n;
Arg args[10];
Widget w;
XmString labelString;
XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
/* alef lam baa noon taa - iso8859-6 */
labelString = XmStringGenerate("\307\344\310\346\312", NULL,
XmCHARSET_TEXT, "ArabicShaped");
w = XtVaCreateManagedWidget("a label", xmLabelWidgetClass, parent,
        XmNlabelString, labelString,
        XmNlabelType, XmSTRING,
        XmNlabelType, XmSTRING,
        XmNlabelType, XmSTRING,
        NULL);

n = 0;
XtSetArg(args[n], XmNfontName, "-*-*-medium-r-normal-*-24-*-*-*-*-*-*-*");
n++;
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
XtSetArg(args[n], XmNlayoutAttrObject, "ar"); n++;
XtSetArg(args[n], XmNlayoutModifier, "@ls numerals=nominal:contextual, shapecharset=iso8859-6"); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args, n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
```

**Horizontal Tabs**

To control the placement of text, a compound string can contain tab characters. To interpret those characters on display, a widget refers to the rendition in effect for that compound string, where it finds a list of tab stops. However, the dynamic widgets (TextField and XmText) do not use the tab resource of the rendition. Instead, they compute the tab width using the formula of 8*(width of character 0).

The tab measurement is the distance from the left margin of the compound string display, or from the right margin, if the layout direction is right-to-left. It is important to note that, regardless of the direction of the text (Arabic right-to-left or English left-to-right), the tab inserts space to the right or left as specified by the layout direction (XmNlayoutDirection).

The text following a tab is always aligned at the tab stop, and the tab stop is calculated from the start of the widget, which in turn is influenced by XmNlayoutDirection.
The behavior of the tabs and their interaction with directionality of the text and the `XmNlayoutDirection` of the widget is illustrated in the following figure.

The input for this illustration is `abc
def\tgh`.

```
abc  def  gh
__________
[ English ]
```

**Layout Direction: XmLEFT_TO_RIGHT**

```
gh  def  abc
__________
[ English ]
```

**Layout Direction: XmRIGHT_TO_LEFT**

*Figure 7–2  Tabbing Behavior*

---

**Mouse Selection**

The user makes a primary selection with SELECT (the left mouse button). Pressing SELECT deselects any existing selection and moves the insertion cursor and the anchor to the position in the text where the button is pressed. Dragging SELECT selects all text between the anchor and the pointer position, deselecting any text outside the range.

The text selected is influenced by the resource `XmNeditPolicy`, which can be set to `XmEDIT_LOGICAL` or `XmEDIT_VISUAL`. If the `XmNeditPolicy` is set to `XmEDIT_LOGICAL`, and if the text selected is bi-directional, the selected text is not contiguous visually and will be a collection of segments. This is because the text in the logical buffer does not have a one-to-one correspondence with the display.

As a result, the contiguous buffer of logical characters of bi-directional text, when rendered does not result in a continuous stream of characters. Conversely, when the `XmNeditPolicy` is set to `XmEDIT_VISUAL`, the text selected can be contiguous.
visually but is segmented in the logical buffer. So the sequence of selection, deletion, and insertion of bi-directional text at the same cursor point does not result in the same string.

Keyboard Selection

The selection operation available with the mouse is also available with the keyboard. The combination of Shift-arrow keys allows the selection of text.

The text selected is influenced by the resource XmNeditPolicy, which can be set to XmEDIT_LOGICAL or XmEDIT_VISUAL. If the XmNeditPolicy is set to XmEDIT_LOGICAL, and if the text selected is bi-directional, the selected text will not be contiguous visually and will be a collection of segments. This is because the text in the logical buffer does not have one-to-one correspondence with the display. As a result, the contiguous buffer of logical characters of bi-directional text, when rendered does not result in a continuous stream of characters.

Conversely, when the XmNeditPolicy is set to XmEDIT_VISUAL, the text selected can be contiguous visually but is segmented in the logical buffer. So the sequence of selection, deletion, and insertion of bi-directional text at the same cursor point does not result in the same string.

Text Resources and Geometry

Text has several resources that relate to geometry, including the following:

- The render table XmNrenderTable that the widget uses to select a font or font set and other attributes in which to display the text

  The Text and Textfield widgets can use only the font-related rendition resources, such as XmNfontType, and can also specify the attributes of the layout object, such as XmNlayoutAttrObject, usually a locale identifier, and XmNlayoutModifier, which specifies the layout values to be passed through to the Layout Object associated with the XOC associated with this XmRendition.

- A resource (XmNwordWrap) that specifies whether lines are broken at word boundaries when the text would be wider than the widget

  Breaking a line at a word boundary does not insert a new line into the text. In the case of cursive languages like Arabic, if the word length is greater than the widget length, the word is wrapped to the next line, but the first character in the next line is shaped independently of the previous character in the logical buffer.
Porting Instructions

The new CTL-enabled Motif library can be found in `/usr/dt/lib/libXm.so.4`. If your application links to `libXm.so.3` (`ldd app_name` shows which library the application is linking to), then it will not support Complex Text Layout (CTL). In order to port the existing applications to enable CTL, you need to perform the following steps.

Add `-DSUN_CTL` to your Makefile. This flag is important and includes the necessary data structures to support CTL. This should be set during compilation.

Recompile the existing application. It will automatically link with the CTL-enabled Motif library `libXm.so.4`.

Add the following resources to your application resource file. Without these resources, the layout engine of the locale will not launch.

Refer to the sample application attached to your documentation.

Use the font name that is available and appropriate to your locale in the `fontName` resource.

1. If you want the cell-based character movement (Thai) in `XmTextField` or `XmText` widgets, set the translations of the corresponding widgets as follows. Refer to the documentation for further detailed explanation.

   ```
   XmText.translations: #override
   <Key>osfRight:forward-cell() \n
   <Key>osfLeft:backward-cell() \n
   <Key>osfDelete:delete-next-cell() \n
   <Key>osfBackSpace:delete-previous-cell() \n   ```
Printing

Localization Printing Support Under the Solaris 8 Operating Environment

The Solaris environment provides support for PostScript printers. Custom print filters are available to convert localized text to PostScript. See \texttt{mp(1) anasi postprint(1)} man pages for further details. Fonts can also be downloaded onto a printer.

For more details, see the \texttt{download(1)} man pages. This support is configured for PostScript printers.

The Solaris 8 environment has a unified printing filter that replaces all the locale-specific filters described below. This section describes this filter and which scripts are supported in each locale Sun supports.

This filter uses font glyphs from printer-resident fonts and TrueType fonts in the Solaris operating environment; PCF bitmap fonts in the Solaris system depend on the configuration information defined for each locale. For more information on PCF (Portable Compiled Format), see man pages \texttt{bdftosnf(1)} and \texttt{bdftopcf(1)}.

European Printing Support

For European locales based on character sets that are not ISO-8859, such as Greek and Russian, \texttt{prolog.ps} files are supplied. The files are located in \texttt{/usr/openwin/lib/locale/print}. 
When you print in one of these locales, the files are automatically downloaded to the printer. These fonts are PostScript Type 1 and include Times, Helvetica, and Courier. These are in normal, **bold**, *italic*, and **bold—italic** styles.

This allows printing on PostScript printers from both CDE and OpenWindows desktops. From a command line, use `/usr/openwin/bin/mp <filename> | lp` in each locale that is not based on ISO 8859–1 character sets.

For the Eastern European locales such as Russian, non-iso-8859-1 encoded, prolog.ps files are supplied. The files are located in:

```
/usr/openwin/lib/locale/locale/directories/print/prolog.ps
```

for each relevant locale. At directories, insert one of the following:

```
/iso8859-2/
/iso8859-4/
/iso8859-5/
/iso8859-7/
/iso8859-9/
/iso8859-10/
```

The files are downloaded automatically when you print in one of the Eastern European locales. A minimum set of fonts allow printing.

The fonts in the prolog.ps files are defined in the following table.

### TABLE 8–1 prolog.ps Fonts

<table>
<thead>
<tr>
<th>Font Path</th>
<th>Font Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/LC_Courier</code></td>
<td>CourierCyr AliasFont</td>
</tr>
<tr>
<td><code>/LC_Courier-Italic</code></td>
<td>CourierCyr Inclined AliasFont</td>
</tr>
<tr>
<td><code>/LC_Courier-Bold</code></td>
<td>CourierCyr Bold AliasFont</td>
</tr>
<tr>
<td><code>/LC_Courier-BoldOblique</code></td>
<td>CourierCyr BoldInclined AliasFont</td>
</tr>
<tr>
<td><code>/LC_Times-Roman</code></td>
<td>TimesNewRomanCyr</td>
</tr>
<tr>
<td><code>/LC_Times-Italic</code></td>
<td>TimesNewRomanCyr-Inclined AliasFont</td>
</tr>
<tr>
<td><code>/LC_Times-Bold</code></td>
<td>TimesNewRomanCyr-Bold AliasFont</td>
</tr>
<tr>
<td><code>/LC_Times-BoldOblique</code></td>
<td>TimesNewRomanCyr-BoldIncl AliasFont</td>
</tr>
<tr>
<td><code>/LC_Helvetica</code></td>
<td>LucidaSansCyr AliasFont</td>
</tr>
<tr>
<td><code>/LC_Helvetica-Italic</code></td>
<td>LucidaSansCyr ItalicFont</td>
</tr>
<tr>
<td><code>/LC_Helvetica-Bold</code></td>
<td>LucidaSansCyr Bold AliasFont</td>
</tr>
<tr>
<td><code>/LC_Helvetica-BoldOblique</code></td>
<td>LucidaSansCyr-BoldItalic AliasFont</td>
</tr>
</tbody>
</table>
Asian Multibyte Printing Support

The `xetops` and `xutops` utilities convert Asian text into a bitmapped graphics printed image. This enables you to print Asian characters on PostScript-based printers, even without having Asian fonts resident on the printers.

A typical command line for printing such a file would be as follows:

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

or

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

(for the ko.UTF-8, zh.UTF-8 and zh_TW.UTF-8 locales)

Japanese Solaris 8 supports the following Japanese-specific printers:

- Japanese PostScript printer
- Epson VP-5085 (based on ESC/P)
- NEC PC-PR201 (based on 201PL)
- Canon LASERSHOT (based on LIPS)

Japanese texts can be printed with these printers through the LP print service. The following table shows the relation between these printers and user components. See `JFP User's Guide` for further details.

**TABLE 8–2 Japanese Printer Support**

<table>
<thead>
<tr>
<th>Printer</th>
<th>terminfo(-T)</th>
<th>interface(-i)</th>
<th>content(-I)</th>
<th>filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Japanese PS</td>
<td>PS</td>
<td>jstandard</td>
<td>postscript</td>
<td>jpostprint</td>
</tr>
<tr>
<td>Epson VP-5085</td>
<td>epson-vp5085</td>
<td>jstandard</td>
<td>None</td>
<td>jprconv</td>
</tr>
<tr>
<td>NEC PC-PR201</td>
<td>nec-pr201</td>
<td>jstandard</td>
<td>None</td>
<td>jprconv</td>
</tr>
<tr>
<td>Canon LASERSHOT</td>
<td>canon-ls-a408</td>
<td>jstandard</td>
<td>None</td>
<td>jprconv</td>
</tr>
</tbody>
</table>
Use the following to set up a Japanese PostScript printer.

In the following example, the PostScript printer name is lw. The /dev/lp1 is the device that is associated with the printer. For more information, see the lpadmin(1M) man page.

```
# lpadmin -p lw -v /dev/lp1 -T PS -I postscript
# lpadmin -p lw -i /usr/lib/lp/model/jstandard
# cd /etc/lp/fd
# lpfilter -x -f postprint
# lpfilter -f jpostprint -F jpostprint.fd
# accept lw
# enable lw
# /etc/init.d/lp stop
# /etc/init.d/lp start
```

To print, use the following operation:

```
% lp -d lw Japanese Text File
```

These features are supported only on Japanese Solaris. Input codesets to a printer depend on the system locale.

---

**Solaris Font Downloader**

The Solaris Font Downloader is a vital part of internationalized printing. PostScript printers sold in different countries do not always have a set of locale-specific fonts installed on them. The usual solution for this problem was to have these locale-specific fonts included with each print job, which tended to lead to enormously large, slowly-processed, print jobs.

An alternative is to have all the frequently used fonts reside on the printer. They can be placed either in printer RAM, or on a hard disk if a printer has one connected to it. Most modern PostScript printers have the option of connecting a hard disk to them. The process of taking font files from the workstation and placing them on a printer is called “downloading.” Fonts downloaded to RAM are available until the printer is power-cycled. Fonts downloaded to a hard disk are available until they are removed.

The Solaris Font Downloader is a GUI application for managing fonts on PostScript printers. It supports a number of different popular printers running PostScript Level 2 or Level 3 software and connected to a network with TCP/IP protocol.
Specifically, it provides the following functionality:

- Download PostScript fonts to a printer
- Convert and download TrueType fonts to a printer
- Remove previously downloaded fonts from a printer
- Report general information, or properties, about a printer, such as the amount of RAM and hard disk capacity, and a list of available fonts, for example.
- Print character samples
- Reformat hard disk on a printer

The Solaris Font Downloader works with a variety of different fonts available for a computer user. It can download the following PostScript fonts to a printer:

- Type 1
- Type 3
- Type 9 (CID Type 0)
- Type 10 (CID Type 1),
- Type 11 (CID Type 2),
- Type 42

It can also convert TrueType fonts to PostScript fonts such as Type 42 fonts or CID (Type 11) fonts “on the fly”, while these fonts are being downloaded. A PostScript software that supports such fonts uses these converted TrueType fonts as if they were regular PostScript fonts.

There are a number of user-selectable choices for converting TrueType fonts to PostScript fonts. These are fully documented along with the rest of the Solaris Font Downloader features in the man page `fdl(1)`.

Reference Documents

- TrueType 1.0 Font Files. Technical Specification Revision 1.66, November 1995 - Microsoft Corporation available from ftp.microsoft.com
# iconv Code Conversions

## TABLE A–1  Available Unicode Related iconv Code Conversion Modules in the Solaris 8 Environment

<table>
<thead>
<tr>
<th>From Code (Symbol)</th>
<th>To Code (Symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>646 (ISO 646)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>USC-2BE</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>USC-2LE</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>USC-4</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>USC-4BE</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>USC-4LE</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>646 (ISO 646)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859–11</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UCS4-LE</td>
</tr>
<tr>
<td>From Code (Symbol)</td>
<td>To Code (Symbol)</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-1 (ISO 8859-1)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-10 (ISO 8859-10)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-13 (ISO 8859-13)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UCS-4LE</td>
</tr>
</tbody>
</table>
### TABLE A–1  Available Unicode Related `iconv` Code Conversion Modules in the Solaris 8 Environment  
(continued)

<table>
<thead>
<tr>
<th>From Code (Symbol)</th>
<th>To Code (Symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-14 (ISO 8859-14)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-15 (ISO 8859-15)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-2 (ISO 8859-2)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>From Code (Symbol)</td>
<td>To Code (Symbol)</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-3 (ISO 8859-3)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UCS4-LE</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-4 (ISO 8859-4)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UCS4-LE</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-5 (ISO 8859-5)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UCS4-LE</td>
</tr>
<tr>
<td>From Code (Symbol)</td>
<td>To Code (Symbol)</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-6 (ISO 8859-6)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UCS-4BE</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-7 (ISO 8859-7)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UCS-4BE</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-8 (ISO 8859-8)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UCS-2</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UCS-4</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>From Code (Symbol)</td>
<td>To Code (Symbol)</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UTF-16</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>8859-9 (ISO 8859-9)</td>
<td>UTF-8</td>
</tr>
<tr>
<td>eucJP</td>
<td>UTF-8</td>
</tr>
<tr>
<td>gb2312</td>
<td>UTF-8</td>
</tr>
<tr>
<td>iso2022</td>
<td>UTF-8</td>
</tr>
<tr>
<td>ko_KR-cp933</td>
<td>UTF-8</td>
</tr>
<tr>
<td>ko_KR-euc</td>
<td>UTF-8</td>
</tr>
<tr>
<td>ko_KR-iso2022–7</td>
<td>UTF-8</td>
</tr>
<tr>
<td>ko_KR-johap</td>
<td>UTF-8</td>
</tr>
<tr>
<td>ko_KR-johap92</td>
<td>UTF-8</td>
</tr>
<tr>
<td>zh_TW-euc</td>
<td>UTF-8</td>
</tr>
<tr>
<td>zh_TW-big5</td>
<td>UTF-8</td>
</tr>
<tr>
<td>zh_TW-cp937</td>
<td>UTF-8</td>
</tr>
<tr>
<td>zh_TW-iso2022–7</td>
<td>UTF-8</td>
</tr>
<tr>
<td>GBK</td>
<td>UTF-8</td>
</tr>
<tr>
<td>ISO-2022–JP</td>
<td>UTF-8</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UCS-2</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UCS-4</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UCS4-BE</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UCS4LE</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UTF-8</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UTF-16</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>UTF-8</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UCS-2</td>
</tr>
<tr>
<td>From Code (Symbol)</td>
<td>To Code (Symbol)</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UCS-4</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UCS-4BE</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UTF-16</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>UTF-8</td>
</tr>
<tr>
<td>PCK</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UCS-2</td>
<td>646 (ISO 646)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-1 (ISO8859-1)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-10 (ISO 8859-10)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-13 (ISO 8859-13)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-14 (ISO 8859-14)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-15 (ISO8859-15)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-2 (ISO8859-2)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-3 (ISO8859-3)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-4 (ISO8859-4)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-5 (ISO8859-5)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-6 (ISO8859-6)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-7 (ISO8859-7)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-8 (ISO8859-8)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>8859-9 (ISO8859-9)</td>
</tr>
<tr>
<td>UCS-2</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>UCS-2</td>
<td>KOI8-U</td>
</tr>
<tr>
<td>UCS-2</td>
<td>UCS-4</td>
</tr>
<tr>
<td>UCS-2</td>
<td>UCS-4BE</td>
</tr>
<tr>
<td>UCS-2</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>UCS-2</td>
<td>UTF-7</td>
</tr>
<tr>
<td>From Code (Symbol)</td>
<td>To Code (Symbol)</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>UCS-2</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>646 (ISO 646)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-1 (ISO 8859-1)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-10 (ISO 8859-10)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-13 (ISO 8859-13)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-14 (ISO 8859-14)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-15 (ISO 8859-15)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-2 (ISO 8859-2)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-3 (ISO 8859-3)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-4 (ISO 8859-4)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-5 (ISO 8859-5)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-6 (ISO 8859-6)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-7 (ISO 8859-7)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-8 (ISO 8859-8)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>8859-9 (ISO 8859-9)</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>KOI8-U</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>UCS-4</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>UCS-4BE</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>UCS-2BE</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UCS-4</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>646 (ISO 646)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-1 (ISO 8859-1)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-10 (ISO 8859-10)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-13 (ISO 8859-13)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-14 (ISO 8859-14)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-15 (ISO 8859-15)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-2 (ISO 8859-2)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-3 (ISO 8859-3)</td>
</tr>
</tbody>
</table>
TABLE A–1  Available Unicode Related *iconv* Code Conversion Modules in the Solaris 8 Environment (continued)

<table>
<thead>
<tr>
<th>From Code (Symbol)</th>
<th>To Code (Symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCS-4LE</td>
<td>8859-4 (ISO 8859-4)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-5 (ISO 8859-5)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-6 (ISO 8859-6)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-7 (ISO 8859-7)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-8 (ISO 8859-8)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>8859-9 (ISO 8859-9)</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>KOI8-U</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>UCS-2</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>UTF-16</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>UTF-16LE</td>
</tr>
<tr>
<td>UCS-4LE</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UTF-7</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UTF-8</td>
<td>646</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-1</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-2</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-3</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-4</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-5</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-6</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-7</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-8</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-9</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-10</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-11</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-15</td>
</tr>
<tr>
<td>UTF-8</td>
<td>eucJP</td>
</tr>
</tbody>
</table>
### TABLE A–1
Available Unicode Related `iconv` Code Conversion Modules in the Solaris 8 Environment

<table>
<thead>
<tr>
<th>From Code (Symbol)</th>
<th>To Code (Symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF-8</td>
<td>gb2312</td>
</tr>
<tr>
<td>UTF-8</td>
<td>iso2022</td>
</tr>
<tr>
<td>UTF-8</td>
<td>ko_KR-euc</td>
</tr>
<tr>
<td>UTF-8</td>
<td>ko_KR-johap</td>
</tr>
<tr>
<td>UTF-8</td>
<td>ko_KR-johap92</td>
</tr>
<tr>
<td>UTF-8</td>
<td>ko_KR-iso2022–7</td>
</tr>
<tr>
<td>UTF-8</td>
<td>zh_TW-euc</td>
</tr>
<tr>
<td>UTF-8</td>
<td>zh_TW-big5</td>
</tr>
<tr>
<td>UTF-8</td>
<td>zh_TW-iso2022–7</td>
</tr>
<tr>
<td>UTF-8</td>
<td>zh_TW-cp937</td>
</tr>
<tr>
<td>UTF-8</td>
<td>GBK</td>
</tr>
<tr>
<td>UTF-8</td>
<td>ISO-2022-JP</td>
</tr>
<tr>
<td>UTF-8</td>
<td>KOI8–R</td>
</tr>
<tr>
<td>UTF-8</td>
<td>PCK</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UCS-2</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UCS-4</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-7</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-16</td>
</tr>
<tr>
<td>UTF-16</td>
<td>646 (ISO 646)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-1 (ISO8859-1)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-10 (ISO8859-10)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-13 (ISO8859-13)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-14 (ISO8859-14)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-15 (ISO8859-15)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-2 (ISO8859-2)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-3 (ISO8859-3)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-4 (ISO8859-4)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-5 (ISO8859-5)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-6 (ISO8859-6)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-7 (ISO8859-7)</td>
</tr>
<tr>
<td>From Code (Symbol)</td>
<td>To Code (Symbol)</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-8 (ISO 8859-8)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>8859-9 (ISO 8859-9)</td>
</tr>
<tr>
<td>UTF-16</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>UTF-16</td>
<td>KOI8-U</td>
</tr>
<tr>
<td>UTF-16</td>
<td>UCS-4</td>
</tr>
<tr>
<td>UTF-16</td>
<td>UCS-4BE</td>
</tr>
<tr>
<td>UTF-16</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>UTF-16</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>646 (ISO 646)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-1 (ISO 8859-1)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-10 (ISO 8859-10)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-13 (ISO 8859-13)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-14 (ISO 8859-14)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-15 (ISO 8859-15)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-2 (ISO 8859-2)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-3 (ISO 8859-3)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-4 (ISO 8859-4)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-5 (ISO 8859-5)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-6 (ISO 8859-6)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-7 (ISO 8859-7)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-8 (ISO 8859-8)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>8859-9 (ISO 8859-9)</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>KOI8-U</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>UCS-4</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>UCS-4BE</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>UTF-16BE</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>646 (ISO 646)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-1 (ISO 8859-1)</td>
</tr>
<tr>
<td>From Code (Symbol)</td>
<td>To Code (Symbol)</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-10 (ISO 8859-10)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-13 (ISO 8859-13)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-14 (ISO 8859-14)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-15 (ISO 8859-15)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-2 (ISO 8859-2)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-3 (ISO 8859-3)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-4 (ISO 8859-4)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-5 (ISO 8859-5)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-6 (ISO 8859-6)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-7 (ISO 8859-7)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-8 (ISO 8859-8)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>8859-9 (ISO 8859-9)</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>KOI8-U</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>UCS-4</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>UCS-4BE</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>UTF-16LE</td>
<td>UTF-8</td>
</tr>
<tr>
<td>UTF-7</td>
<td>UCS-2</td>
</tr>
<tr>
<td>UTF-7</td>
<td>UCS-4</td>
</tr>
<tr>
<td>UTF-7</td>
<td>UCS-8</td>
</tr>
<tr>
<td>UTF-8</td>
<td>646 (ISO 646)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-1 (ISO 8859-1)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-10 (ISO 8859-10)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-13 (ISO 8859-13)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-14 (ISO 8859-14)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-15 (ISO 8859-15)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-2 (ISO 8859-2)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-3 (ISO 8859-3)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-4 (ISO 8859-4)</td>
</tr>
</tbody>
</table>
**TABLE A–1** Available Unicode Related `iconv` Code Conversion Modules in the Solaris 8 Environment (continued)

<table>
<thead>
<tr>
<th>From Code (Symbol)</th>
<th>To Code (Symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF-8</td>
<td>8859-5 (ISO 8859-5)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-6 (ISO 8859-6)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-7 (ISO 8859-7)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-8 (ISO 8859-8)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>8859-9 (ISO 8859-9)</td>
</tr>
<tr>
<td>UTF-8</td>
<td>KOI8-R</td>
</tr>
<tr>
<td>UTF-8</td>
<td>KOI8-U</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UCS-2</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UCS-2BE</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UCS-2LE</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UCS-4</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UCS-4BE</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UCS-4LE</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-16</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-16BE</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UCS-16LE</td>
</tr>
<tr>
<td>UTF-8</td>
<td>UTF-7</td>
</tr>
</tbody>
</table>

UTF-EBCDIC is a new IBM codepage name. Starting with the Solaris 8 environment, we are also supporting bidirectional UTF-8 <--- UTF-EBCDIC conversion.

**TABLE A–2** Available Unicode and IBM/Microsoft EBCDIC and PC Code Page Related `iconv` Code Conversions Modules in the Solaris 8 Environment

<table>
<thead>
<tr>
<th>From Code (Symbol)</th>
<th>To Code (Symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF-8</td>
<td>IBM-037</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-273</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-277</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-278</td>
</tr>
<tr>
<td>From Code (Symbol)</td>
<td>To Code (Symbol)</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-280</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-284</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-285</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-297</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-420</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-424</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-500</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-870</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-875</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-880</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-1025</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-1026</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-1112</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-1122</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-850</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-852</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-855</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-856</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-857</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-862</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-864</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-866</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-869</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-921</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-922</td>
</tr>
<tr>
<td>UTF-8</td>
<td>IBM-1046</td>
</tr>
</tbody>
</table>
### TABLE A-2  Available Unicode and IBM/Microsoft EBCDIC and PC Code Page Related `iconv` Code Conversions Modules in the Solaris 8 Environment (continued)

<table>
<thead>
<tr>
<th>From Code (Symbol)</th>
<th>To Code (Symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF-8</td>
<td>CP850</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP852</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP855</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP857</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP862</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP864</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP866</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP869</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP874</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP1250</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP1251</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP1252</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP1253</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP1254</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP1255</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP1256</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP1257</td>
</tr>
<tr>
<td>UTF-8</td>
<td>CP1258</td>
</tr>
</tbody>
</table>

### TABLE A-3  Available `iconv` Code Conversions IBM and Microsoft EBCDIC/PC Code Pages to UTF-8

<table>
<thead>
<tr>
<th>From Code</th>
<th>To Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF-EBCDIC</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-037</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-273</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-277</td>
<td>UTF-8</td>
</tr>
</tbody>
</table>
TABLE A–3  Available `iconv` Code Conversions IBM and Microsoft EBCDIC/PC Code Pages to UTF-8 (continued)

<table>
<thead>
<tr>
<th>Code Page</th>
<th>Conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-278</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-280</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-284</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-285</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-297</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-420</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-424</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-500</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-870</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-875</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-880</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-1025</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-1026</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-1112</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-1122</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-850</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-852</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-855</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-856</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-857</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-862</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-864</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-866</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-869</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-921</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-922</td>
<td>UTF-8</td>
</tr>
<tr>
<td>IBM-1046</td>
<td>UTF-8</td>
</tr>
<tr>
<td>Code</td>
<td>Conversion</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>CP850</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP852</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP855</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP857</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP862</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP864</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP866</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP869</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP874</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP1250</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP1251</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP1252</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP1253</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP1254</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP1255</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP1256</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP1257</td>
<td>UTF-8</td>
</tr>
<tr>
<td>CP1258</td>
<td>UTF-8</td>
</tr>
</tbody>
</table>
### Partial L10N Package Names on OS CD

**TABLE B–1** List of Partial Locales

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWauaox</td>
<td>Australasia 64-bit OS Support</td>
</tr>
<tr>
<td>SUNWauadt</td>
<td>Australasia CDE Support</td>
</tr>
<tr>
<td>SUNWauaos</td>
<td>Australasia OS Support</td>
</tr>
<tr>
<td>SUNWauaow</td>
<td>Australasia OW Support</td>
</tr>
<tr>
<td>SUNWcamox</td>
<td>Central America 64-bit OS Support</td>
</tr>
<tr>
<td>SUNWcamdt</td>
<td>Central America CDE Support</td>
</tr>
<tr>
<td>SUNWcamos</td>
<td>Central America OS Support</td>
</tr>
<tr>
<td>SUNWcamow</td>
<td>Central America OW Support</td>
</tr>
<tr>
<td>SUNWceuox</td>
<td>Central Europe 64-bit OS Support</td>
</tr>
<tr>
<td>SUNWceudt</td>
<td>Central Europe CDE Support</td>
</tr>
<tr>
<td>SUNWceuos</td>
<td>Central Europe OS Support</td>
</tr>
<tr>
<td>SUNWceuow</td>
<td>Central Europe OW Support</td>
</tr>
<tr>
<td>SUNWalex</td>
<td>Common files shared by Chinese, Japanese and Korean locales. It is a required package to run Asian Language Environment (64-bit)</td>
</tr>
<tr>
<td>SUNWeeuox</td>
<td>Eastern Europe 64-bit OS Support</td>
</tr>
<tr>
<td>SUNWeudt</td>
<td>Eastern Europe CDE Support</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUNWeeuos</td>
<td>Eastern Europe OS Support</td>
</tr>
<tr>
<td>SUNWeeuow</td>
<td>Eastern Europe OW Support</td>
</tr>
<tr>
<td>SUNWfiris</td>
<td>French install software localization</td>
</tr>
<tr>
<td>SUNWdeis</td>
<td>German install software localization</td>
</tr>
<tr>
<td>SUNWitis</td>
<td>Italian install software localization</td>
</tr>
<tr>
<td>NSCPjacom</td>
<td>Japanese (common) localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>SUNWjc0r</td>
<td>Japanese Kana-Kanji Conversion Server cs00 Root Files</td>
</tr>
<tr>
<td>SUNWjc0u</td>
<td>Japanese Kana-Kanji Conversion Server cs00 User Files</td>
</tr>
<tr>
<td>SUNWjedt</td>
<td>Japanese (EUC) Localization for CDE DESKTOP LOGIN ENVIRONMENT</td>
</tr>
<tr>
<td>SUNWjeuc</td>
<td>Japanese (EUC) Feature Package specific files for \texttt{usr}; it is a required package to support EUC environment.</td>
</tr>
<tr>
<td>SUNWjeucx</td>
<td>Japanese (EUC) Feature Package specific 64-bit files for \texttt{usr}; it is a required package to run JFP environment.</td>
</tr>
<tr>
<td>SUNWjexpl</td>
<td>Japanese (EUC) Localizations for X Window System platform software.</td>
</tr>
<tr>
<td>SUNWjexpx</td>
<td>Japanese (EUC) Localizations for X Window System platform software (64-bit)</td>
</tr>
<tr>
<td>SUNWjfpr</td>
<td>Stream modules for Japanese Feature Package (JFP); it is a required package to run JFP environment.</td>
</tr>
<tr>
<td>SUNWjfpu</td>
<td>Japanese Feature Package (JFP) specific files for \texttt{usr}; it is a required package to run JFP environment.</td>
</tr>
<tr>
<td>SUNWjfpx</td>
<td>Japanese Feature Package (JFP) specific 64-bit files for \texttt{usr}; it is a required package to run JFP environment.</td>
</tr>
<tr>
<td>SUNWjman</td>
<td>Japanese Feature Package Man Pages to see English man pages for \texttt{SUNWjfpr} and \texttt{SUNWjfpu}</td>
</tr>
<tr>
<td>SUNWjpck</td>
<td>Japanese (PCK - PC Kanji Code) Feature Package specific files; it’s a required package to support PCK environment.</td>
</tr>
<tr>
<td>SUNWjpckx</td>
<td>Japanese (PCK) Feature Package specific 64-bit files for \texttt{usr}; it is a required package to run JFP environment.</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUNWjpdt</td>
<td>Japanese (PCK) Localization for CDE DESKTOP LOGIN ENVIRONMENT</td>
</tr>
<tr>
<td>SUNWjpxml</td>
<td>Japanese (PCK) Localizations for X Window System platform software</td>
</tr>
<tr>
<td>SUNWjpxmlx</td>
<td>Japanese (PCK) Localizations for X Window System platform software (64-bit)</td>
</tr>
<tr>
<td>SUNWju8</td>
<td>Japanese (UTF-8) Feature Package specific files; it’s a required package to support Japanese UTF-8 environment.</td>
</tr>
<tr>
<td>SUNWju8x</td>
<td>Japanese (UTF-8) Feature Package specific 64-bit files for usr; it is a required package to run JFP environment.</td>
</tr>
<tr>
<td>SUNWjudt</td>
<td>Japanese (UTF-8) Localization for CDE DESKTOP LOGIN ENVIRONMENT</td>
</tr>
<tr>
<td>SUNWjuxpl</td>
<td>Japanese (UTF-8) Localizations for X Window System platform software</td>
</tr>
<tr>
<td>SUNWjcft</td>
<td>Japanese JISX212 TrueType and bitmap fonts</td>
</tr>
<tr>
<td>SUNWkleux</td>
<td>Korean (EUC) Language Environment specific files. It is a required package to run Korean Language Environment (64-bit).</td>
</tr>
<tr>
<td>SUNWkulenx</td>
<td>Korean (UTF-8) Language Environment specific files. It is a required package to run Korean Language Environment (64-bit)</td>
</tr>
<tr>
<td>SUNWkdt</td>
<td>Korean Localizations for CDE Desktop Login Environment.</td>
</tr>
<tr>
<td>SUNSCPcpc</td>
<td>Simplified Chinese partial version of Netscape Communicator 4.7 supporting International security</td>
</tr>
<tr>
<td>SUNWSamox</td>
<td>Southern America 64-bit OS Support</td>
</tr>
<tr>
<td>SUNWSamdt</td>
<td>Southern America CDE Support</td>
</tr>
<tr>
<td>SUNWSamos</td>
<td>Southern America OS Support</td>
</tr>
<tr>
<td>SUNWSamow</td>
<td>Southern America OW Support</td>
</tr>
<tr>
<td>SUNWseuox</td>
<td>Southern Europe 64-bit OS Support</td>
</tr>
<tr>
<td>SUNWseudt</td>
<td>Southern Europe CDE Support</td>
</tr>
<tr>
<td>SUNWseuos</td>
<td>Southern Europe OS Support</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUNWseuow</td>
<td>Southern Europe OW Support</td>
</tr>
<tr>
<td>SUNWfrspla</td>
<td>Spell Checking Engine - French Dictionary</td>
</tr>
<tr>
<td>SUNWdespla</td>
<td>Spell Checking Engine - German Dictionary</td>
</tr>
<tr>
<td>SUNWitspla</td>
<td>Spell Checking Engine - Italian Dictionary</td>
</tr>
<tr>
<td>SUNWesspla</td>
<td>Spell Checking Engine - Spanish Dictionary</td>
</tr>
<tr>
<td>SUNWsvspla</td>
<td>Spell Checking Engine - Swedish Dictionary</td>
</tr>
<tr>
<td>SUNWjfp</td>
<td>Stream modules for Japanese Feature Package (JFP), it is a required package to run JFP environment</td>
</tr>
<tr>
<td>SUNWsvsis</td>
<td>Swedish install software localization</td>
</tr>
<tr>
<td>SUNWkleu</td>
<td>This package contains Korean Language Environment specific files. It is a required package to run Korean Language Environment.</td>
</tr>
<tr>
<td>SUNWkuleu</td>
<td>This package contains Korean UTF-8 Language Environment specific files. It is a required package to run Korean Language Environment.</td>
</tr>
<tr>
<td>SUNWcleu</td>
<td>This package contains Korean UTF-8 Language Environment specific files. It is a required package to run Korean Language Environment.</td>
</tr>
<tr>
<td>SUNWgleu</td>
<td>This package contains Simplified Chinese (GBK) Language Environment specific files. It is a required package to run Simplified Chinese (GBK) Language Environment.</td>
</tr>
<tr>
<td>SUNWculeu</td>
<td>This package contains Simplified Chinese (UTF-8) Language Environment specific files. It is a required package to run Simplified Chinese (UTF-8) Language Environment.</td>
</tr>
<tr>
<td>SUNWtleu</td>
<td>This package contains Thai Language Environment specific files. It is a required package to run Thai Language Environment.</td>
</tr>
<tr>
<td>SUNWhuleu</td>
<td>This package contains Traditional Chinese (UTF-8) Language Environment specific files. It is a required package to run Traditional Chinese UTF-8 Language Environment.</td>
</tr>
<tr>
<td>SUNW5leu</td>
<td>This package contains Traditional Chinese Language Environment specific files. It is a required package to run Traditional Chinese BIG5 Language Environment.</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUNWhleu</td>
<td>This package contains Traditional Chinese Language Environment specific files. It is a required package to run Traditional Chinese Language Environment</td>
</tr>
<tr>
<td>SUNWale</td>
<td>This package contains common files shared by Chinese, Japanese and Korean locales. It is a required package to run Asian Language Environment</td>
</tr>
<tr>
<td>SUNWaled</td>
<td>This package contains man pages shared by Chinese, Japanese and Korean locales.</td>
</tr>
<tr>
<td>SUNW5leux</td>
<td>Traditional Chinese (BIG5) Language Environment user files (64-bit)</td>
</tr>
<tr>
<td>SUNW5xplx</td>
<td>Traditional Chinese (BIG5) X Windows Platform Software Package (64-bit)</td>
</tr>
<tr>
<td>SUNW8leux</td>
<td>Traditional Chinese (EUC) Language Environment specific files. It is a required package to run Traditional Chinese Language Environment (64-bit)</td>
</tr>
<tr>
<td>SUNW8ulex</td>
<td>Traditional Chinese (UTF-8) Language Environment user files (64-bit)</td>
</tr>
<tr>
<td>SUNW5xplt</td>
<td>Traditional Chinese BIG5 X Windows Platform Software Package</td>
</tr>
<tr>
<td>SUNW5dt</td>
<td>Traditional Chinese Localizations for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNW8hdt</td>
<td>Traditional Chinese Localizations for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNW8hcid</td>
<td>Traditional Chinese Solaris Install CD L10N source files</td>
</tr>
<tr>
<td>SUNW5ttf</td>
<td>Traditional Chinese True Type Fonts Package</td>
</tr>
<tr>
<td>SUNW8ttf</td>
<td>Traditional Chinese True Type Fonts Package</td>
</tr>
<tr>
<td>SUNW8huplt</td>
<td>Traditional Chinese UTF-8 X Windows Platform Software Package</td>
</tr>
<tr>
<td>SUNW8xplt</td>
<td>Traditional Chinese X Windows Platform Software Package</td>
</tr>
<tr>
<td>SUNW8xfrnt</td>
<td>Traditional Chinese X Windows Platform required Fonts Package</td>
</tr>
<tr>
<td>SUNSCPhpcom</td>
<td>Traditional Chinese partial version of Netscape Communicator 4.7 supporting International security</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SUNWweuox</td>
<td>Western Europe 64-bit OS Support</td>
</tr>
<tr>
<td>SUNWweudt</td>
<td>Western Europe CDE Support</td>
</tr>
<tr>
<td>SUNWweuos</td>
<td>Western Europe OS Support</td>
</tr>
<tr>
<td>SUNWweuow</td>
<td>Western Europe OW Support</td>
</tr>
<tr>
<td>SUNWi1cs</td>
<td>X11 ISO8859-1 Codeset Support</td>
</tr>
<tr>
<td>SUNWi13cs</td>
<td>X11 ISO8859-13 Codeset Support</td>
</tr>
<tr>
<td>SUNWi15cs</td>
<td>X11 ISO8859-15 Codeset Support</td>
</tr>
<tr>
<td>SUNWi2cs</td>
<td>X11 ISO8859-2 Codeset Support</td>
</tr>
<tr>
<td>SUNWi5cs</td>
<td>X11 ISO8859–5 Codeset Support</td>
</tr>
<tr>
<td>SUNWi7cs</td>
<td>X11 ISO8859–7 Codeset Support</td>
</tr>
<tr>
<td>SUNWi9cs</td>
<td>X11 ISO8859–9 Codeset Support</td>
</tr>
<tr>
<td>SUNWi2of</td>
<td>X11 fonts for ISO-8859-2 character set (optional fonts)</td>
</tr>
<tr>
<td>SUNWi4of</td>
<td>X11 fonts for ISO-8859-4 character set (optional fonts)</td>
</tr>
<tr>
<td>SUNWi5of</td>
<td>X11 fonts for ISO-8859-5 character set (optional fonts)</td>
</tr>
<tr>
<td>SUNWi7of</td>
<td>X11 fonts for ISO-8859-7 character set (optional fonts)</td>
</tr>
<tr>
<td>SUNWi9of</td>
<td>X11 fonts for ISO-8859-9 character set (optional fonts)</td>
</tr>
<tr>
<td>SUNWkoi8f</td>
<td>X11 fonts for KOI8–R character set</td>
</tr>
</tbody>
</table>
# Languages CD Packages List

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSCPccom</td>
<td>Simplified Chinese localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>NSCPcucom</td>
<td>zh.UTF-8 localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>NSCPgcom</td>
<td>zh.GBK localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>SUNWcadis</td>
<td>Simplified Chinese (EUC) Localizations for admintool and GUI install.</td>
</tr>
<tr>
<td>SUNWcadma</td>
<td>Simplified Chinese (EUC) Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNWhadis packages for Simplified Chinese (EUC) localization.</td>
</tr>
<tr>
<td>SUNWcdbc</td>
<td>This package contains Simplified Chinese (EUC) Language Environment binary compatibility files.</td>
</tr>
<tr>
<td>SUNWcdab</td>
<td>Simplified Chinese (EUC) Localizations for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNWcdbas</td>
<td>Simplified Chinese (EUC) Localizations for CDE Base functionality</td>
</tr>
<tr>
<td>SUNWcdbst</td>
<td>Simplified Chinese (EUC) Localizations for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWcdtte</td>
<td>Simplified Chinese (EUC) Localizations for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWcedzr</td>
<td>Simplified Chinese (EUC) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>Package Code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUNWcdft</td>
<td>Simplified Chinese (EUC) Localizations for CDE Fonts</td>
</tr>
<tr>
<td>SUNWcdhe</td>
<td>Simplified Chinese (EUC) Localizations for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNWcdhev</td>
<td>Simplified Chinese (EUC) CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWcdhez</td>
<td>Simplified Chinese (EUC) (Common) Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWcdicn</td>
<td>Simplified Chinese (EUC) Localizations for CDE Icons</td>
</tr>
<tr>
<td>SUNWcdim</td>
<td>Simplified Chinese (EUC) Localizations for CDE Imagetool</td>
</tr>
<tr>
<td>SUNWcdwm</td>
<td>Simplified Chinese (EUC) Localizations for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNWcepmw</td>
<td>Simplified Chinese (EUC) Localization for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWcervl</td>
<td>Simplified Chinese (EUC) SunVideo Runtime Support Software</td>
</tr>
<tr>
<td>SUNWcevir</td>
<td>Simplified Chinese (EUC) XIL Runtime Environment</td>
</tr>
<tr>
<td>SUNWcj2p</td>
<td>Simplified Chinese localization of Java Plug-in 1.2.2</td>
</tr>
<tr>
<td>SUNWcj2rt</td>
<td>Java virtual machine and core class libraries (Simplified Chinese supplement)</td>
</tr>
<tr>
<td>SUNWcjvdv</td>
<td>Simplified Chinese Localizations for JavaVM developers package</td>
</tr>
<tr>
<td>SUNWcjvrt</td>
<td>Simplified Chinese Localizations for JavaVM run time environment</td>
</tr>
<tr>
<td>SUNWckcsr</td>
<td>Simplified Chinese (EUC) KCMS Runtime Environment</td>
</tr>
<tr>
<td>SUNWcleue</td>
<td>This package contains Simplified Chinese (EUC) Language Environment specific files. It is a required package to run Simplified Chinese (EUC) Language Environment</td>
</tr>
<tr>
<td>SUNWcoaud</td>
<td>Simplified Chinese (EUC) OPENLOOK Audio Applications Package</td>
</tr>
<tr>
<td>SUNWcodcv</td>
<td>Simplified Chinese (EUC) OPENLOOK Document and Help Viewer Applications Package</td>
</tr>
<tr>
<td>SUNWcodem</td>
<td>Simplified Chinese (EUC) OPENLOOK Demo Programs Package</td>
</tr>
<tr>
<td>SUNWcodst</td>
<td>Simplified Chinese (EUC) OPENLOOK Deskset Tools Package</td>
</tr>
<tr>
<td>SUNWcodte</td>
<td>Simplified Chinese (EUC) Core OPENLOOK Desktop Package</td>
</tr>
<tr>
<td>SUNWcoimt</td>
<td>Simplified Chinese (EUC) OPENLOOK Imagetool Package</td>
</tr>
<tr>
<td>SUNWcoman</td>
<td>Simplified Chinese (EUC) OPENLOOK Toolkit/Desktop Users Man Pages Package</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUNWcorte</td>
<td>Simplified Chinese (EUC) OPENLOOK Toolkits Runtime Environment Package</td>
</tr>
<tr>
<td>SUNWcrdm</td>
<td>Simplified Chinese (EUC) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWcreg</td>
<td>Simplified Chinese (EUC) Localizations for Solaris User Registration</td>
</tr>
<tr>
<td>SUNWcsadl</td>
<td>Simplified Chinese (EUC) Localizations for Solstice Admintool launcher and associated libraries.</td>
</tr>
<tr>
<td>SUNWctltk</td>
<td>Simplified Chinese (EUC) ToolTalk Runtime Package Package</td>
</tr>
<tr>
<td>SUNWcttte</td>
<td>Simplified Chinese (EUC) True Type Fonts</td>
</tr>
<tr>
<td>SUNWcuada</td>
<td>Simplified Chinese (UTF-8) Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNWgadis packages for Simplified Chinese (UTF-8) localization.</td>
</tr>
<tr>
<td>SUNWcuadi</td>
<td>Simplified Chinese (UTF-8) Localizations for admintool and GUI install</td>
</tr>
<tr>
<td>SUNWcubas</td>
<td>Simplified Chinese (UTF-8) Localizations for CDE Base functionality</td>
</tr>
<tr>
<td>SUNWcudab</td>
<td>Simplified Chinese (UTF-8) Localizations for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNWcudc</td>
<td>Simplified Chinese (EUC) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNWcudez</td>
<td>Simplified Chinese (UTF-8) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>SUNWcudft</td>
<td>Simplified Chinese (UTF-8) Localizations for CDE Fonts</td>
</tr>
<tr>
<td>SUNWcudhe</td>
<td>Simplified Chinese (UTF-8) Localizations for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNWcudhv</td>
<td>Simplified Chinese (UTF-8) CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWcudhz</td>
<td>Simplified Chinese (UTF-8) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWcudic</td>
<td>Simplified Chinese (UTF-8) Localizations for CDE Icons</td>
</tr>
<tr>
<td>SUNWcudim</td>
<td>Simplified Chinese (UTF-8) L10N for CDE DESKTOP ImageTool</td>
</tr>
<tr>
<td>SUNWcudst</td>
<td>Simplified Chinese (UTF-8) Localizations for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWcudte</td>
<td>Simplified Chinese (UTF-8) Localizations for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWcudwm</td>
<td>Simplified Chinese (UTF-8) Localizations for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUNWculee</td>
<td>This package contains Simplified Chinese (UTF-8) Language Environment specific files. It is a required package to run Simplified Chinese (UTF-8) Language Environment</td>
</tr>
<tr>
<td>SUNWcuman</td>
<td>Simplified Chinese (UTF-8) X Windows Online User Man Pages Package</td>
</tr>
<tr>
<td>SUNWcuodt</td>
<td>Simplified Chinese (UTF-8) Core OPENLOOK Desktop Package</td>
</tr>
<tr>
<td>SUNWcupmw</td>
<td>Simplified Chinese (UTF-8) Localization for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWcurdm</td>
<td>Simplified Chinese (UTF-8) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWcureg</td>
<td>Simplified Chinese (UTF-8) Localizations for Solaris User Registration</td>
</tr>
<tr>
<td>SUNWcusad</td>
<td>Simplified Chinese (UTF-8) Localizations for Solstice Admintool launcher and associated libraries.</td>
</tr>
<tr>
<td>SUNWcuudec</td>
<td>Simplified Chinese (UTF-8) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNWcuxe</td>
<td>Simplified Chinese (UTF-8) X Windows Platform Software Package</td>
</tr>
<tr>
<td>SUNWcwbcpl</td>
<td>Simplified Chinese (EUC) OpenWindows Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWcwsr</td>
<td>Simplified Chinese (EUC) prodreg 2.0 localizable text resources</td>
</tr>
<tr>
<td>SUNWcxe</td>
<td>Simplified Chinese (EUC) X Windows Platform Software Package</td>
</tr>
<tr>
<td>SUNWcxnt</td>
<td>Simplified Chinese (EUC) X Windows Platform Required Fonts</td>
</tr>
<tr>
<td>SUNWcxman</td>
<td>Simplified Chinese (EUC) X Windows Online User Man Pages Package</td>
</tr>
<tr>
<td>SUNWcxoft</td>
<td>Simplified Chinese (EUC) X Windows Optional Fonts Package</td>
</tr>
<tr>
<td>SUNW gadis</td>
<td>Simplified Chinese (GBK) Localizations for admintool and GUI install</td>
</tr>
<tr>
<td>SUNW gadma</td>
<td>Simplified Chinese (GBK) Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNW gadis packages for Simplified Chinese (GBK) localization.</td>
</tr>
<tr>
<td>SUNW gdbab</td>
<td>Simplified Chinese (GBK) Localizations for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNW gdbas</td>
<td>Simplified Chinese (GBK) Localizations for CDE Base functionality</td>
</tr>
<tr>
<td>SUNW gddst</td>
<td>Simplified Chinese (GBK) Localizations for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNW gddte</td>
<td>Simplified Chinese (GBK) Localizations for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNW gdez</td>
<td>Simplified Chinese (GBK) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>SUNWgdft</td>
<td>Simplified Chinese (GBK) Localizations for CDE Fonts</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>SUNWgdhe</td>
<td>Simplified Chinese (GBK) Localizations for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNWgdhev</td>
<td>Simplified Chinese (GBK) CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWgdhez</td>
<td>Simplified Chinese (GBK) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWgdicn</td>
<td>Simplified Chinese (GBK) Localizations for CDE Icons</td>
</tr>
<tr>
<td>SUNWgdim</td>
<td>Simplified Chinese (GBK) L10N for CDE DESKTOP Imagetool</td>
</tr>
<tr>
<td>SUNWgdwm</td>
<td>Simplified Chinese (GBK) Localizations for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNWgleue</td>
<td>This package contains Simplified Chinese (GBK) Language Environment specific files. It is a required package to run Simplified Chinese (GBK) Language Environment</td>
</tr>
<tr>
<td>SUNWgodte</td>
<td>Simplified Chinese (GBK) Core OPENLOOK Desktop Package</td>
</tr>
<tr>
<td>SUNWgpmw</td>
<td>Simplified Chinese (GBK) Localization for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWgrdm</td>
<td>Simplified Chinese (GBK) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWgreg</td>
<td>Simplified Chinese (GBK) Localizations for Solaris User Registration</td>
</tr>
<tr>
<td>SUNWgsadl</td>
<td>Simplified Chinese (GBK) Localizations for Solstice Admintool launcher and associated libraries.</td>
</tr>
<tr>
<td>SUNWgttfe</td>
<td>Simplified Chinese (GBK) True Type Fonts</td>
</tr>
<tr>
<td>SUNWgudc</td>
<td>Simplified Chinese (GBK) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNWgxe</td>
<td>Simplified Chinese (GBK) X Windows Platform Software Package</td>
</tr>
<tr>
<td>SUNWgxman</td>
<td>Simplified Chinese (GBK) X Windows Online User Man Pages Package</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>NSCPfrdo</td>
<td>French localization of Netscape Communicator 4.7 supporting U.S. security.</td>
</tr>
<tr>
<td>NSCPfrcom</td>
<td>French localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>SUNWf8bas</td>
<td>Base L10N fr CDE functionality to run a CDE application</td>
</tr>
<tr>
<td>SUNWf8dst</td>
<td>CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWf8dte</td>
<td>CDE Desktop Environment</td>
</tr>
<tr>
<td>SUNWf8he</td>
<td>CDE Help L10N fr Runtime Environment</td>
</tr>
<tr>
<td>SUNWf8im</td>
<td>CDE Desktop apps</td>
</tr>
<tr>
<td>SUNWf8wm</td>
<td>French UTF-8 CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNWfbcp</td>
<td>French OS Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWf0aud</td>
<td>French French OPEN LOOK (R) Audio applications</td>
</tr>
<tr>
<td>SUNWf0bk</td>
<td>French OpenWindows online handbooks</td>
</tr>
<tr>
<td>SUNWf0dev</td>
<td>French OPEN LOOK (R) document and help viewer applications</td>
</tr>
<tr>
<td>SUNWf0dem</td>
<td>French OPEN LOOK (R) demo programs</td>
</tr>
<tr>
<td>SUNWf0dst</td>
<td>French OPEN LOOK (R) deset tools</td>
</tr>
<tr>
<td>SUNWf0dte</td>
<td>French OPEN LOOK (R) desktop environment</td>
</tr>
<tr>
<td>SUNWf0smt</td>
<td>French OPEN LOOK (R) imagetool</td>
</tr>
<tr>
<td>SUNWf0tte</td>
<td>French OPEN LOOK (R) toolkits runtime environment</td>
</tr>
<tr>
<td>SUNWf1bas</td>
<td>Base L10N fr CDE functionality to run a CDE application</td>
</tr>
<tr>
<td>SUNWf1dst</td>
<td>CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWf1dte</td>
<td>CDE Desktop Environment</td>
</tr>
<tr>
<td>SUNWf1he</td>
<td>CDE Help L10N fr Runtime Environment</td>
</tr>
<tr>
<td>SUNWf1hed</td>
<td>CDE Help L10N fr Help Developer Environment</td>
</tr>
<tr>
<td>SUNWf1hev</td>
<td>CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWf1rim</td>
<td>CDE Desktop apps</td>
</tr>
<tr>
<td>SUNWf1rj2p</td>
<td>French localization of Java Plug-in 1.2.2</td>
</tr>
<tr>
<td>SUNWf1ros</td>
<td>localizable message files for the OS-Networking consolidation</td>
</tr>
</tbody>
</table>
### TABLE C–2 French (continued)

<table>
<thead>
<tr>
<th>SUNWfrpmw</th>
<th>French (EUC) Localizations for Power Management OW Utilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWfrreg</td>
<td>Solaris User Registration prompts at desktop login for user registration</td>
</tr>
<tr>
<td>SUNWfrwm</td>
<td>French CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNWftltk</td>
<td>French ToolTalk binaries and shared libraries</td>
</tr>
<tr>
<td>SUNWfwacx</td>
<td>French OPEN LOOK (R) AccessX</td>
</tr>
<tr>
<td>SUNWfwbcp</td>
<td>French OpenWindows Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWfxpltt</td>
<td>French X Windows platform software</td>
</tr>
</tbody>
</table>

### TABLE C–3 German

<table>
<thead>
<tr>
<th>NSCPdecom</th>
<th>German localization of Netscape Communicator 4.7 supporting International security.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWd8bas</td>
<td>Base L10N German UTF-8 CDE functionality to run a CDE application</td>
</tr>
<tr>
<td>SUNWd8dst</td>
<td>CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWd8dte</td>
<td>CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWd8he</td>
<td>CDE Help L10N German UTF-8 Runtime Environment</td>
</tr>
<tr>
<td>SUNWd8im</td>
<td>CDE Desktop apps</td>
</tr>
<tr>
<td>SUNWd8wm</td>
<td>German UTF-8 CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNWdbcplex</td>
<td>German OS Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWdebas</td>
<td>Base L10N German CDE functionality to run a CDE application</td>
</tr>
<tr>
<td>SUNWdedst</td>
<td>CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWdedte</td>
<td>CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWdehe</td>
<td>CDE Help L10N German Runtime Environment</td>
</tr>
<tr>
<td>SUNWdehed</td>
<td>CDE L10N German Help Developer Environment</td>
</tr>
</tbody>
</table>
**TABLE C–3**

<table>
<thead>
<tr>
<th>SUNWdehev</th>
<th>CDE Help Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWdeim</td>
<td>CDE Desktop apps</td>
</tr>
<tr>
<td>SUNWdej2p</td>
<td>German localization of Java Plug-in 1.2.2</td>
</tr>
<tr>
<td>SUNWdeos</td>
<td>localizable message files for the OS-Networking consolidation</td>
</tr>
<tr>
<td>SUNWdepmw</td>
<td>German (EUC) Localizations for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWdereg</td>
<td>Solaris User Registration prompts at desktop login for user registration</td>
</tr>
<tr>
<td>SUNWdewm</td>
<td>German CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNWdoaud</td>
<td>German OPEN LOOK (R) Audio applications</td>
</tr>
<tr>
<td>SUNWdobk</td>
<td>German OpenWindows online handbooks</td>
</tr>
<tr>
<td>SUNWdodcv</td>
<td>German OPEN LOOK (R) document and help viewer applications</td>
</tr>
<tr>
<td>SUNWdodem</td>
<td>German OPEN LOOK (R) demo programs</td>
</tr>
<tr>
<td>SUNWdodst</td>
<td>German OPEN LOOK (R) deskset tools</td>
</tr>
<tr>
<td>SUNWdodte</td>
<td>German OPEN LOOK (R) desktop environment</td>
</tr>
<tr>
<td>SUNWdoimt</td>
<td>German OPEN LOOK (R) imagetool</td>
</tr>
<tr>
<td>SUNWdorte</td>
<td>German OPEN LOOK (R) toolkits runtime environment</td>
</tr>
<tr>
<td>SUNWddtltk</td>
<td>German ToolTalk binaries and shared libraries</td>
</tr>
<tr>
<td>SUNWdwacx</td>
<td>German OPEN LOOK (R) AccessX</td>
</tr>
<tr>
<td>SUNWdwbcsp</td>
<td>German OpenWindows Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWdxplt</td>
<td>German X Windows platform software</td>
</tr>
</tbody>
</table>

**TABLE C–4**

<table>
<thead>
<tr>
<th>NSCPitcom</th>
<th>Italian localization of Netscape Communicator 4.7 supporting International security.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWi8bas</td>
<td>Base L10N it CDE functionality to run a CDE application</td>
</tr>
<tr>
<td>SUNWi8dst</td>
<td>CDE it Desktop Applications messages</td>
</tr>
<tr>
<td>SUNWi8dte</td>
<td>CDE Italian UTF-8 Desktop Login Environment</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>SUNWi8he</td>
<td>CDE Help L10N it Runtime Environment</td>
</tr>
<tr>
<td>SUNWi8im</td>
<td>CDE Italian UTF-8 Desktop Image editor</td>
</tr>
<tr>
<td>SUNWi8wm</td>
<td>Italian UTF-8 CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNWibcp</td>
<td>Italian OS Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWiaaoud</td>
<td>Italian OPEN LOOK (R) Audio applications</td>
</tr>
<tr>
<td>SUNWioobk</td>
<td>Italian OpenWindows online handbooks</td>
</tr>
<tr>
<td>SUNWiodcv</td>
<td>Italian OPEN LOOK (R) document and help viewer applications</td>
</tr>
<tr>
<td>SUNWiodem</td>
<td>Italian OPEN LOOK (R) demo programs</td>
</tr>
<tr>
<td>SUNWiodst</td>
<td>Italian OPEN LOOK (R) deskset tools</td>
</tr>
<tr>
<td>SUNWiodte</td>
<td>Italian OPEN LOOK (R) desktop environment</td>
</tr>
<tr>
<td>SUNWioimt</td>
<td>Italian OPEN LOOK (R) imagetool</td>
</tr>
<tr>
<td>SUNWiorte</td>
<td>Italian OPEN LOOK (R) toolkits runtime environment</td>
</tr>
<tr>
<td>SUNWitbas</td>
<td>Base L10N it CDE functionality to run a CDE application</td>
</tr>
<tr>
<td>SUNWitdst</td>
<td>CDE it Desktop Applications messages</td>
</tr>
<tr>
<td>SUNWitdte</td>
<td>CDE Italian Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWithe</td>
<td>CDE Help L10N it Runtime Environment</td>
</tr>
<tr>
<td>SUNWithed</td>
<td>CDE L10N it Help Developer Environment</td>
</tr>
<tr>
<td>SUNWithev</td>
<td>CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWitim</td>
<td>CDE Italian Desktop Image editor</td>
</tr>
<tr>
<td>SUNWiitj2p</td>
<td>Italian localization of Java Plug-in 1.2.2</td>
</tr>
<tr>
<td>SUNWitltk</td>
<td>Italian ToolTalk binaries and shared libraries</td>
</tr>
<tr>
<td>SUNWitos</td>
<td>localizable message files for the OS-Networking consolidation</td>
</tr>
<tr>
<td>SUNWitpmw</td>
<td>Italian (EUC) Localizations for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWitreg</td>
<td>Solaris User Registration prompts at desktop login for user registration</td>
</tr>
<tr>
<td>SUNWitwm</td>
<td>Italian CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNWiwacx</td>
<td>Italian OPEN LOOK (R) AccessX</td>
</tr>
<tr>
<td>TABLE C–4</td>
<td>Italian (continued)</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>SUNWiwbcp</td>
<td>Italian OpenWindows Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWixplt</td>
<td>Italian X Windows platform software</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE C–5</th>
<th>Japanese</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSat8xw</td>
<td>Japanese Input System ATOK8 for Japanese Solaris.</td>
</tr>
<tr>
<td>JSatsvr</td>
<td>Japanese Input System ATOKserver root files for Japanese Solaris</td>
</tr>
<tr>
<td>JSatsvu</td>
<td>Japanese Input System ATOKserver usr files for Japanese Solaris</td>
</tr>
<tr>
<td>JSatsvw</td>
<td>Japanese Input System ATOKserver X11 support files for Japanese Solaris</td>
</tr>
<tr>
<td>NSCPjecom</td>
<td>Japanese (EUC) localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>NSCPjpcom</td>
<td>Japanese (PCK) localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>NSCPjucom</td>
<td>Japanese (UTF-8) localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>SUNWjadis</td>
<td>Japanese (EUC) Localizations for admintool and GUI install.</td>
</tr>
<tr>
<td>SUNWjadm</td>
<td>Japanese (EUC) Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNWjadis packages for Japanese (EUC) localization.</td>
</tr>
<tr>
<td>SUNWjaj2p</td>
<td>Japanese localization of Java Plug-in 1.2.2</td>
</tr>
<tr>
<td>SUNWjbc</td>
<td>Japanese (EUC) utilities including libc and locale data to provide a binary-compatible execution environment for SunOS 4.x applications.</td>
</tr>
<tr>
<td>SUNWjc0d</td>
<td>Japanese Kana-Kanji Conversion Server cs00 user dictionary maintenance tool for CDE Motif</td>
</tr>
<tr>
<td>SUNWjc0w</td>
<td>Japanese Kana-Kanji Conversion Server cs00 user dictionary maintenance tool for OPEN LOOK. This package is also required to use X Input Method Server on Window System.</td>
</tr>
<tr>
<td>SUNWjcs3f</td>
<td>Japanese JIS X0212 Type1 fonts for printing</td>
</tr>
<tr>
<td>SUNWjdab</td>
<td>Japanese (Common) Localization for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNWjdbas</td>
<td>Japanese (Common) Localization for CDE application basic runtime environment</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUNWjdstd</td>
<td>Japanese (EUC) Localization for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWjdste</td>
<td>Japanese (EUC) Localization for Solaris Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWjdhcm</td>
<td>Japanese Localizations for DHCP Manager</td>
</tr>
<tr>
<td>SUNWjdhe</td>
<td>Japanese (EUC) Localization for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNWjdhed</td>
<td>Japanese (EUC) Localization for CDE Help Developer Environment</td>
</tr>
<tr>
<td>SUNWjdhev</td>
<td>Japanese (Common) Localization for CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWjdhez</td>
<td>Japanese (Common) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWjdim</td>
<td>Japanese (EUC) Localization for Solaris CDE Image Viewer</td>
</tr>
<tr>
<td>SUNWjdme</td>
<td>Japanese (EUC) Localization for Common Desktop Environment (CDE) release documentation</td>
</tr>
<tr>
<td>SUNWjdwm</td>
<td>Japanese (EUC) Localization for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNWjeab</td>
<td>Japanese (EUC) Localization for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNWjebas</td>
<td>Japanese (EUC) Localization for CDE application basic runtime environment</td>
</tr>
<tr>
<td>SUNWject</td>
<td>Japanese (EUC) Localizations for UTF-8 Code Conversion Tool</td>
</tr>
<tr>
<td>SUNWjedev</td>
<td>Japanese (EUC) Development Environment Package specific files</td>
</tr>
<tr>
<td>SUNWjeezt</td>
<td>Japanese (EUC) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>SUNWjehev</td>
<td>Japanese (EUC) Localization for CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWjehez</td>
<td>Japanese (EUC) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWjej2m</td>
<td>Japanese (EUC) man pages</td>
</tr>
<tr>
<td>SUNWjejmm</td>
<td>Japanese (EUC) JavaVM manual pages for Java programmers and users</td>
</tr>
<tr>
<td>SUNWjeman</td>
<td>Japanese Feature Package Man Pages to see Japanese (EUC) manpages</td>
</tr>
<tr>
<td>SUNWjepmm</td>
<td>Japanese (EUC) Power Management OW Utilities Man Pages</td>
</tr>
<tr>
<td>SUNWjepmw</td>
<td>Japanese (EUC) Localizations for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWjervl</td>
<td>Japanese (EUC) Localizations for XIL loadable pipelines for SunVideo capture and compression</td>
</tr>
<tr>
<td>SUNWjeuce</td>
<td>Japanese (EUC) Feature Package specific files forusr, it is a extended package to support EUC environment.</td>
</tr>
<tr>
<td>SUNWjjeudc</td>
<td>Japanese (EUC) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>SUNWjewnu</td>
<td>Japanese Input System - Wnn6 Messages, (EUC)</td>
</tr>
<tr>
<td>SUNWjexfa</td>
<td>Japanese (EUC) Localizations for Font Administration application for Solaris platforms</td>
</tr>
<tr>
<td>SUNWjexir</td>
<td>Japanese (EUC) localizations for XIL Runtime Environment</td>
</tr>
<tr>
<td>SUNWjfdl</td>
<td>Japanese Localization for Solaris Desktop Font Downloader for Adobe Postscript printers</td>
</tr>
<tr>
<td>SUNWjfpre</td>
<td>Stream modules for Japanese Feature Package (JFP), it is an extended package to run JFP environment.</td>
</tr>
<tr>
<td>SUNWjfprue</td>
<td>Japanese Feature Package (JFP) specific files for usr, it is an extended package to run JFP environment.</td>
</tr>
<tr>
<td>SUNWjfxmn</td>
<td>English manpages of Japanese features for X Window System.</td>
</tr>
<tr>
<td>SUNWjj2dv</td>
<td>Japanese Java virtual macTools and utilities including javac, jdb, javadoc, rmiregistry</td>
</tr>
<tr>
<td>SUNWjj2rt</td>
<td>Japanese Java virtual machine and core class libraries</td>
</tr>
<tr>
<td>SUNWjjvdv</td>
<td>Japanese Localizations for JavaVM developers package</td>
</tr>
<tr>
<td>SUNWjjvrt</td>
<td>Japanese Localizations for JavaVM run time environment</td>
</tr>
<tr>
<td>SUNWjkcsl</td>
<td>Japanese (EUC) Localizations for Kodak Color Management System Runtime</td>
</tr>
<tr>
<td>SUNWjlibj</td>
<td>Japanese specific library (/usr/lib/libjapanese.a), header and transition kit</td>
</tr>
<tr>
<td>SUNWjimane</td>
<td>Japanese Feature Package Man Pages (Extension) to see English manpages for SUNWjfpre and SUNWjfprue.</td>
</tr>
<tr>
<td>SUNWjmfrn</td>
<td>Japanese (EUC) Localizations for Motif 1.2.3 RunTime Kit.</td>
</tr>
<tr>
<td>SUNWjoaud</td>
<td>Japanese (EUC) Localizations for Audiotool &amp; other auxiliary audio support</td>
</tr>
<tr>
<td>SUNWjodcv</td>
<td>Japanese (EUC) Localizations for OPEN LOOK document and help viewer applications</td>
</tr>
<tr>
<td>SUNWjodem</td>
<td>Japanese (EUC) Localizations for OPEN LOOK demo programs</td>
</tr>
<tr>
<td>SUNWjodst</td>
<td>Japanese (EUC) Localizations for OPEN LOOK deskset tools</td>
</tr>
<tr>
<td>SUNWjodte</td>
<td>Japanese (EUC) Localizations for OPEN LOOK Desktop Environment (olwm, props, wsinfo, etc.)</td>
</tr>
<tr>
<td>SunWjoints</td>
<td>Japanese (EUC) Localization for OPEN LOOK imagem tool</td>
</tr>
<tr>
<td>SunWjortc</td>
<td>Japanese (EUC) Localization for OPEN LOOK toolkits runtime environment</td>
</tr>
<tr>
<td>SunWjoumn</td>
<td>Japanese (EUC) OPEN LOOK toolkit/desktop users man pages</td>
</tr>
<tr>
<td>SunWjbab</td>
<td>Japanese (PCK) Localization for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SunWjpacx</td>
<td>Japanese (PCK) Localizations for AccessX client program</td>
</tr>
<tr>
<td>SunWjpadi</td>
<td>Japanese (PCK) Localizations for admintool and GUI install.</td>
</tr>
<tr>
<td>SunWjpadm</td>
<td>Japanese (PCK) Localizations for Software used to perform system administration tasks. Admintool requires both this and SunWjpadm packages for Japanese (PCK) localization.</td>
</tr>
<tr>
<td>SunWjpbas</td>
<td>Japanese (PCK) Localization for CDE application basic runtime environment</td>
</tr>
<tr>
<td>SunWjpcke</td>
<td>Japanese (PCK - PC Kanji Code) Feature Package specific files. It’s a extended package to support PCK environment.</td>
</tr>
<tr>
<td>SunWjpct</td>
<td>Japanese (PCK) Localizations for UTF-8 Code Conversion Tool</td>
</tr>
<tr>
<td>SunWjpdas</td>
<td>Japanese localization for tools to synchronize desktop applications with the Palm Pilot PDA</td>
</tr>
<tr>
<td>SunWjpdst</td>
<td>Japanese (PCK) Localization for CDE Desktop Applications</td>
</tr>
<tr>
<td>SunWjpdtc</td>
<td>Japanese (PCK) Localization for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SunWjpzst</td>
<td>Japanese (PCK) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>SunWjphe</td>
<td>Japanese (PCK) Localization for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SunWjphefd</td>
<td>Japanese (PCK) Localization for CDE Help Developer Environment</td>
</tr>
<tr>
<td>SunWjphev</td>
<td>Japanese (PCK) Localization for CDE Help Volumes</td>
</tr>
<tr>
<td>SunWjphez</td>
<td>Japanese (PCK) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SunWjpim</td>
<td>Japanese (PCK) Localization for Solaris CDE Image Viewer</td>
</tr>
<tr>
<td>SunWjpj2m</td>
<td>Japanese (PCK) man pages</td>
</tr>
<tr>
<td>SunWjpjmn</td>
<td>Japanese (PCK) JavaVM manual pages for Java programmers and users</td>
</tr>
<tr>
<td>SunWjpccs</td>
<td>Japanese (PCK) Localizations for Kodak Color Management System Runtime</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUNWjpman</td>
<td>Japanese Feature Package Man Pages to see Japanese (PCK) manpages for SUNWjfpr and SUNWjfpu and Japanese manpages for SUNWman and SUNWaled.</td>
</tr>
<tr>
<td>SUNWjpmfr</td>
<td>Japanese (PCK) Localizations for Motif 1.2.3 RunTime Kit.</td>
</tr>
<tr>
<td>SUNWjppmm</td>
<td>Japanese (PCK) Power Management OW Utilities Man Pages</td>
</tr>
<tr>
<td>SUNWjppmw</td>
<td>Japanese (PCK) Localizations for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWjprdm</td>
<td>Japanese (PCK) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWjprme</td>
<td>Japanese (PCK) Localization for Common Desktop Environment (CDE) release documentation</td>
</tr>
<tr>
<td>SUNWjprvl</td>
<td>Japanese (PCK) Localizations for XIL loadable pipelines for SunVideo capture and compression</td>
</tr>
<tr>
<td>SUNWjiptlm</td>
<td>Japanese (PCK) ToolTalk manual pages for ToolTalk programmers, OpenWindows users, and Common Desktop Environment (CDE) users</td>
</tr>
<tr>
<td>SUNWjiptlt</td>
<td>Japanese (PCK) Localizations for ToolTalk binaries and shared libraries needed for Common Desktop Environment (CDE), OpenWindows, and all ToolTalk clients</td>
</tr>
<tr>
<td>SUNWjpudc</td>
<td>Japanese (PCK) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNWjpwm</td>
<td>Japanese (PCK) Localization for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNWjpwnu</td>
<td>Japanese Input System - Wnn6 Messages, (PCK)</td>
</tr>
<tr>
<td>SUNWjpxfa</td>
<td>Japanese (PCK) Localizations for Font Administration application for Solaris platforms</td>
</tr>
<tr>
<td>SUNWjpxir</td>
<td>Japanese (PCK) Localizations for XIL Runtime Environment</td>
</tr>
<tr>
<td>SUNWjpxpm</td>
<td>Japanese (PCK) X Window System online programmers man pages</td>
</tr>
<tr>
<td>SUNWjpxum</td>
<td>Japanese (PCK) X Window System online user man pages</td>
</tr>
<tr>
<td>SUNWjrdm</td>
<td>Japanese (EUC) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWjreg</td>
<td>Japanese Localizations for Solaris User Registration</td>
</tr>
<tr>
<td>SUNWjtltmn</td>
<td>Japanese (EUC) ToolTalk manual pages for ToolTalk programmers, OpenWindows users, and Common Desktop Environment (CDE) users</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUNWjitlk</td>
<td>Japanese (EUC) Localizations for ToolTalk binaries and shared libraries needed for Common Desktop Environment (CDE), OpenWindows, and all ToolTalk clients</td>
</tr>
<tr>
<td>SUNWju8e</td>
<td>Japanese (UTF-8) Feature Package specific files. It's an extended package to support Japanese UTF-8 environment.</td>
</tr>
<tr>
<td>SUNWjuab</td>
<td>Japanese (UTF-8) Localization for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNWjuacx</td>
<td>Japanese (UTF-8) Localizations for AccessX client program</td>
</tr>
<tr>
<td>SUNWjuadi</td>
<td>Japanese (UTF-8) Localizations for admintool and GUI install.</td>
</tr>
<tr>
<td>SUNWjuadm</td>
<td>Japanese (UTF-8) Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNWjuadi packages for Japanese (UTF-8) localization.</td>
</tr>
<tr>
<td>SUNWjubas</td>
<td>Japanese (UTF-8) Localization for CDE application basic runtime environment</td>
</tr>
<tr>
<td>SUNWjuct</td>
<td>Japanese (UTF-8) Localizations for UTF-8 Code Conversion Tool</td>
</tr>
<tr>
<td>SUNWjudst</td>
<td>Japanese (UTF-8) Localization for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWjudte</td>
<td>Japanese (UTF-8) Localization for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWjuezt</td>
<td>Japanese (UTF-8) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>SUNWjuhe</td>
<td>Japanese (UTF-8) Localization for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNWjuhed</td>
<td>Japanese (UTF-8) Localization for CDE Help Developer Environment</td>
</tr>
<tr>
<td>SUNWjuhev</td>
<td>Japanese (UTF-8) Localization for CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWjuhez</td>
<td>Japanese (UTF-8) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWjuim</td>
<td>Japanese (UTF-8) Localization for Solaris CDE Image Viewer</td>
</tr>
<tr>
<td>SUNWjuj2m</td>
<td>Japanese (UTF-8) man pages</td>
</tr>
<tr>
<td>SUNWjujmn</td>
<td>Japanese (UTF-8) JavaVM Manual pages for Java programmers and users</td>
</tr>
<tr>
<td>SUNWjukcs</td>
<td>Japanese (UTF-8) Localizations for Kodak Color Management System Runtime</td>
</tr>
<tr>
<td>SUNWjulcf</td>
<td>Japanese (UTF-8) Localizations for xutops command</td>
</tr>
<tr>
<td>SUNWjuman</td>
<td>Japanese Feature Package Man Pages to see Japanese (UTF-8) manpages for SUNWjfipr and SUNWjfipu and Japanese manpages for SUNWman and SUNWaled.</td>
</tr>
<tr>
<td>SUNWjumfr</td>
<td>Japanese (UTF-8) Localizations for Motif 1.2.3 RunTime Kit.</td>
</tr>
<tr>
<td>SUNWjupmm</td>
<td>Japanese (UTF-8) Power Management OW Utilities Man Pages</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>SUNWjupmw</td>
<td>Japanese (UTF-8) Localizations for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWjurdm</td>
<td>Japanese (UTF-8) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWjurme</td>
<td>Japanese (UTF-8) Localization for Common Desktop Environment (CDE) release documentation</td>
</tr>
<tr>
<td>SUNWjurvl</td>
<td>Japanese (UTF-8) Localizations for XIL loadable pipelines for SunVideo capture and compression</td>
</tr>
<tr>
<td>SUNWjutlm</td>
<td>Japanese (UTF-8) ToolTalk manual pages for ToolTalk programmers, OpenWindows users, and Common Desktop Environment (CDE) users</td>
</tr>
<tr>
<td>SUNWjutlt</td>
<td>Japanese (UTF-8) Localizations for ToolTalk binaries and shared libraries needed for Common Desktop Environment (CDE), OpenWindows, and all ToolTalk clients</td>
</tr>
<tr>
<td>SUNWjuudc</td>
<td>Japanese (UTF-8) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNWjuwm</td>
<td>Japanese (UTF-8) Localization for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNWjwncu</td>
<td>Japanese Input System - Wnn6 Client, (Usr)</td>
</tr>
<tr>
<td>SUNWjuwm</td>
<td>Japanese Input System - Wnn6 Messages, (UTF-8)</td>
</tr>
<tr>
<td>SUNWjuxfa</td>
<td>Japanese (UTF-8) Localizations for Font Administration application for Solaris platforms</td>
</tr>
<tr>
<td>SUNWjuxir</td>
<td>Japanese (UTF-8) Localizations for XIL Runtime Environment</td>
</tr>
<tr>
<td>SUNWjuxpm</td>
<td>Japanese (UTF-8) X Window System online programmers man pages</td>
</tr>
<tr>
<td>SUNWjuxum</td>
<td>Japanese (UTF-8) X Window System online user man pages</td>
</tr>
<tr>
<td>SUNWjwacx</td>
<td>Japanese (EUC) Localizations for AccessX client program</td>
</tr>
<tr>
<td>SUNWjwbcp</td>
<td>Japanese (EUC) Localizations for Support files, programs, and libraries for Openwindows Binary Compatibility.</td>
</tr>
<tr>
<td>SUNWjwbk</td>
<td>Japanese (EUC) Localizations for OpenWindows online handbooks</td>
</tr>
<tr>
<td>SUNWjwnccu</td>
<td>Japanese Input System - Wnn6 Client X Window System</td>
</tr>
<tr>
<td>TABLE C–5</td>
<td>Japanese (continued)</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>SUNWjwnsr</td>
<td>Japanese Input System - Wnn6 Server, (Root)</td>
</tr>
<tr>
<td>SUNWjwnsu</td>
<td>Japanese Input System - Wnn6 Server, (Usr)</td>
</tr>
<tr>
<td>SUNWjwsr</td>
<td>Japanese Solaris Product Registry</td>
</tr>
<tr>
<td>SUNWjxfa</td>
<td>Japanese (Common) Localizations for Font Administration application for Solaris platforms</td>
</tr>
<tr>
<td>SUNWjxfnt</td>
<td>Japanese X Window System Fonts (required fonts) - gothic bold fonts and TrueType map files</td>
</tr>
<tr>
<td>SUNWjxoft</td>
<td>Sun Minchou bitmap fonts</td>
</tr>
<tr>
<td>SUNWjxplt</td>
<td>Japanese Localizations for X Window System platform software (Extensions)</td>
</tr>
<tr>
<td>SUNWjxpmn</td>
<td>Japanese (EUC) X Window System online programmers man pages</td>
</tr>
<tr>
<td>SUNWjxumn</td>
<td>Japanese (EUC) X Window System online user man pages</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE C–6</th>
<th>Korean</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSCPkocom</td>
<td>Korean localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>NSCPkucom</td>
<td>ko.UTF-8 localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>SUNWkadis</td>
<td>Korean (EUC) Localizations for admintool and GUI install.</td>
</tr>
<tr>
<td>SUNWkadma</td>
<td>Korean (EUC) Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNWkadis packages for Korean (EUC) localization.</td>
</tr>
<tr>
<td>SUNWkbcp</td>
<td>This package contains Korean Language Environment binary compatibility files.</td>
</tr>
<tr>
<td>SUNWkcoft</td>
<td>Korean/Korean UTF-8 common optional font package</td>
</tr>
<tr>
<td>SUNWkdbas</td>
<td>Korean Localizations for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNWkdab</td>
<td>Korean Localizations for CDE Base functionality</td>
</tr>
<tr>
<td>SUNWkddte</td>
<td>The localized tools package for Korean.</td>
</tr>
<tr>
<td>SUNWkddst</td>
<td>Korean Localizations for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWkddte</td>
<td>Korean Localizations for CDE Desktop Login Environment</td>
</tr>
</tbody>
</table>

Printing 203
<table>
<thead>
<tr>
<th>SUNWkdext</th>
<th>Korean (EUC) Localizations for Desktop Power Pack Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWkdft</td>
<td>Fonts for the common desktop environment, Korean L10N CDE</td>
</tr>
<tr>
<td>SUNWkdhe</td>
<td>Korean Localizations for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNWkdhev</td>
<td>Korean CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWkdhez</td>
<td>Korean (Common) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWkdicn</td>
<td>Korean Localizations for CDE Icons</td>
</tr>
<tr>
<td>SUNWkdim</td>
<td>Korean Localizations for CDE Imagetool</td>
</tr>
<tr>
<td>SUNWkdwm</td>
<td>Korean Localizations for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNWkepmw</td>
<td>Korean (EUC) Localization for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWkervl</td>
<td>Korean (EUC) SunVideo Runtime Support Software</td>
</tr>
<tr>
<td>SUNWkexhr</td>
<td>Korean (EUC) XIL Runtime Environment</td>
</tr>
<tr>
<td>SUNWkj2rt</td>
<td>Java virtual machine and core class libraries (Korean supplement)</td>
</tr>
<tr>
<td>SUNWkjdvl</td>
<td>Korean Localizations for JavaVM developers package</td>
</tr>
<tr>
<td>SUNWkjvrt</td>
<td>Korean Localizations for JavaVM run time environment</td>
</tr>
<tr>
<td>SUNWkkcsr</td>
<td>Korean (EUC) KCMS Runtime Environment</td>
</tr>
<tr>
<td>SUNWkler</td>
<td>This package contains the stream modules for Korean Language Environment. It is a required package to run Korean Language Environment</td>
</tr>
<tr>
<td>SUNWklerx</td>
<td>Stream modules for Korean Language Environment. It is a required package to run Korean Language Environment (64-bit)</td>
</tr>
<tr>
<td>SUNWkleeue</td>
<td>This package contains Korean Language Environment specific files. It is a required package to run Korean Language Environment</td>
</tr>
<tr>
<td>SUNWkoaud</td>
<td>Korean OPENLOOK Audio Applications Package</td>
</tr>
<tr>
<td>SUNWkodcv</td>
<td>Korean OPENLOOK Document and Help Viewer Applications Package</td>
</tr>
<tr>
<td>SUNWkdem</td>
<td>Korean OPENLOOK Demo Programs Package</td>
</tr>
<tr>
<td>SUNWkodst</td>
<td>Korean OPENLOOK Deskset Tools Package</td>
</tr>
<tr>
<td>SUNWkodte</td>
<td>Korean Core OPENLOOK Desktop Package</td>
</tr>
<tr>
<td>SUNWkoimt</td>
<td>Korean OPENLOOK Imagetool Package</td>
</tr>
<tr>
<td>SUNWkoj2p</td>
<td>Korean localization of Java Plug-in 1.2.2</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUNWkoman</td>
<td>Korean OPENLOOK Toolkit/Desktop Users Man Pages Package</td>
</tr>
<tr>
<td>SUNWkorte</td>
<td>Korean OPENLOOK Toolkits Runtime Environment Package</td>
</tr>
<tr>
<td>SUNWkrdm</td>
<td>Korean (EUC) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWkreg</td>
<td>Korean Localizations for Solaris User Registration</td>
</tr>
<tr>
<td>SUNWksadl</td>
<td>Korean (EUC) Localizations for Solstice Admintool launcher and associated libraries.</td>
</tr>
<tr>
<td>SUNWktlk</td>
<td>Korean ToolTalk Runtime Package Package</td>
</tr>
<tr>
<td>SUNWkttte</td>
<td>Korean True Type Font Extension</td>
</tr>
<tr>
<td>SUNWkuadi</td>
<td>Korean (UTF-8) Localizations for admintool and GUI install.</td>
</tr>
<tr>
<td>SUNWkuadm</td>
<td>Korean (UTF-8) Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNWkadis packages for Korean (EUC) localization.</td>
</tr>
<tr>
<td>SUNWkudab</td>
<td>Korean/UTF-8 Localizations for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNWkudbs</td>
<td>Korean/UTF-8 Localizations for CDE Base functionality</td>
</tr>
<tr>
<td>SUNWkudc</td>
<td>Korean (EUC) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNWkudda</td>
<td>Korean/UTF-8 Localizations for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWkuddt</td>
<td>Korean/UTF-8 Localizations for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWkudft</td>
<td>Fonts for the common desktop environment, Korean/UTF-8 L10N CDE</td>
</tr>
<tr>
<td>SUNWkudhr</td>
<td>Korean/UTF-8 Localizations for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNWkudhv</td>
<td>Korean/UTF-8 CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWkudhz</td>
<td>Korean (Common) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWkudic</td>
<td>Korean/UTF-8 Localizations for CDE Icons</td>
</tr>
<tr>
<td>SUNWkudim</td>
<td>Korean/UTF-8 Localizations for CDE Imagetool</td>
</tr>
<tr>
<td>SUNWkudwm</td>
<td>Korean/UTF-8 Localizations for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNWkudzt</td>
<td>Korean (UTF-8) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>SUNWkulee</td>
<td>This package contains Korean UTF-8 Language Environment specific files. It is a required package to run Korean Language Environment</td>
</tr>
<tr>
<td>SUNWkuodf</td>
<td>Korean UTF-8 Core OPENLOOK Desktop Package</td>
</tr>
<tr>
<td>SUNWkupmw</td>
<td>Korean UTF-8 Localization for Power Management OW Utilities</td>
</tr>
<tr>
<td>TABLE C–6</td>
<td>Korean (continued)</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>SUNWkurdm</td>
<td>Korean (UTF-8) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWkusal</td>
<td>Korean (UTF-8) Localizations for Solstice Admintool launcher and associated libraries.</td>
</tr>
<tr>
<td>SUNWkuudc</td>
<td>Korean (UTF-8) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNWkuxe</td>
<td>Korean UTF-8 X Windows Platform Software Package</td>
</tr>
<tr>
<td>SUNWkuxft</td>
<td>Korean UTF-8 X Windows Platform Required Fonts</td>
</tr>
<tr>
<td>SUNWkwbcp</td>
<td>Korean OpenWindows Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWkwsr</td>
<td>Korean prodreg 2.0 localizable text resources</td>
</tr>
<tr>
<td>SUNWkxe</td>
<td>Korean X Windows Platform Software Package</td>
</tr>
<tr>
<td>SUNWkxfte</td>
<td>Korean X Windows Platform Required Fonts</td>
</tr>
<tr>
<td>SUNWkxman</td>
<td>Korean X Windows Online User Man Pages Package</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE C–7</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSCPescom</td>
<td>Spanish localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>SUNWe8bas</td>
<td>Base L10N fr CDE functionality to run a CDE application</td>
</tr>
<tr>
<td>SUNWe8dst</td>
<td>CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWe8dte</td>
<td>CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWe8he</td>
<td>CDE Help L10N es Runtime Environment</td>
</tr>
<tr>
<td>SUNWe8im</td>
<td>CDE Desktop apps</td>
</tr>
<tr>
<td>SUNWe8wm</td>
<td>Spanish UTF-8 CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNWeoaud</td>
<td>Spanish OPEN LOOK (R) Audio applications</td>
</tr>
<tr>
<td>SUNWeobk</td>
<td>Spanish OpenWindows online handbooks</td>
</tr>
<tr>
<td>SUNWeodcv</td>
<td>Spanish OPEN LOOK (R) document and help viewer applications</td>
</tr>
<tr>
<td>SUNWeodem</td>
<td>Spanish OPEN LOOK (R) demo programs</td>
</tr>
<tr>
<td>SUNWeodst</td>
<td>Spanish OPEN LOOK (R) deskset tools</td>
</tr>
<tr>
<td>SUNW</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUNW</td>
<td>Spanish OPEN LOOK (R) desktop environment</td>
</tr>
<tr>
<td>SUNW</td>
<td>Spanish OPEN LOOK (R) imagetool</td>
</tr>
<tr>
<td>SUNW</td>
<td>Spanish OPEN LOOK (R) toolkits runtime environment</td>
</tr>
<tr>
<td>SUNW</td>
<td>Base L10N fr CDE functionality to run a CDE application</td>
</tr>
<tr>
<td>SUNW</td>
<td>CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNW</td>
<td>CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNW</td>
<td>CDE Help L10N es Runtime Environment</td>
</tr>
<tr>
<td>SUNW</td>
<td>CDE L10N es Help Developer Environment</td>
</tr>
<tr>
<td>SUNW</td>
<td>CDE Help Volumes</td>
</tr>
<tr>
<td>SUNW</td>
<td>CDE Desktop apps</td>
</tr>
<tr>
<td>SUNW</td>
<td>Spanish localization of Java Plug-in 1.2.2</td>
</tr>
<tr>
<td>SUNW</td>
<td>localizable message files for the OS-Networking consolidation</td>
</tr>
<tr>
<td>SUNW</td>
<td>Spanish (EUC) Localizations for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNW</td>
<td>Solaris User Registration prompts at desktop login for user registration</td>
</tr>
<tr>
<td>SUNW</td>
<td>Spanish CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNW</td>
<td>Spanish ToolTalk binaries and shared libraries</td>
</tr>
<tr>
<td>SUNW</td>
<td>Spanish OPEN LOOK (R) AccessX</td>
</tr>
<tr>
<td>SUNW</td>
<td>Spanish X Windows platform software</td>
</tr>
</tbody>
</table>

<p>| NSCPsvcom | Swedish localization of Netscape Communicator 4.7 supporting International security. |
| SUNW|Base Swedish UTF-8 CDE functionality messages |
| SUNW|Swedish UTF-8 CDE Desktop Applications messages |
| SUNW|Swedish UTF-8 CDE Desktop Login Environment messages |
| SUNW|Swedish UTF-8 CDE Help Runtime Environment |</p>
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWs8im</td>
<td>Swedish UTF-8 CDE Image editor messages</td>
</tr>
<tr>
<td>SUNWs8wm</td>
<td>Swedish UTF-8 CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNWsoaud</td>
<td>Swedish OPEN LOOK (R) Audio applications</td>
</tr>
<tr>
<td>SUNWsobk</td>
<td>Swedish OpenWindows online handbooks</td>
</tr>
<tr>
<td>SUNWsodcv</td>
<td>Swedish OPEN LOOK (R) document and help viewer applications</td>
</tr>
<tr>
<td>SUNWsodem</td>
<td>Swedish OPEN LOOK (R) demo programs</td>
</tr>
<tr>
<td>SUNWsodst</td>
<td>Swedish OPEN LOOK (R) deskset tools</td>
</tr>
<tr>
<td>SUNWsodte</td>
<td>Swedish OPEN LOOK (R) desktop environment</td>
</tr>
<tr>
<td>SUNWssoimt</td>
<td>Swedish OPEN LOOK (R) imagetool</td>
</tr>
<tr>
<td>SUNWssoyte</td>
<td>Swedish OPEN LOOK (R) toolkits runtime environment</td>
</tr>
<tr>
<td>SUNWstltk</td>
<td>Swedish ToolTalk binaries and shared libraries</td>
</tr>
<tr>
<td>SUNWsvbas</td>
<td>Base Swedish CDE functionality messages</td>
</tr>
<tr>
<td>SUNWsvdst</td>
<td>Swedish CDE Desktop Applications messages</td>
</tr>
<tr>
<td>SUNWsvdte</td>
<td>Swedish CDE Desktop Login Environment messages</td>
</tr>
<tr>
<td>SUNWsvehe</td>
<td>Swedish CDE Help Runtime Environment</td>
</tr>
<tr>
<td>SUNWsvehed</td>
<td>Swedish CDE Help Developer Environment messages</td>
</tr>
<tr>
<td>SUNWsvehev</td>
<td>CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWsvim</td>
<td>Swedish CDE Image editor messages</td>
</tr>
<tr>
<td>SUNWsvj2p</td>
<td>Swedish localization of Java Plug-in 1.2.2</td>
</tr>
<tr>
<td>SUNWsvos</td>
<td>localizable message files for the OS-Networking consolidation</td>
</tr>
<tr>
<td>SUNWsvpmw</td>
<td>Swedish (EUC) Localizations for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWsreg</td>
<td>Solaris User Registration prompts at desktop login for user registration</td>
</tr>
<tr>
<td>SUNWswm</td>
<td>Swedish CDE Desktop Window Manages Messages</td>
</tr>
<tr>
<td>SUNWswacl</td>
<td>Swedish OPEN LOOK (R) AccessX</td>
</tr>
<tr>
<td>SUNWxspl</td>
<td>Swedish X Windows platform software</td>
</tr>
<tr>
<td>SUNWvbcf</td>
<td>Swedish OS Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWvwbcp</td>
<td>Swedish OpenWindows Binary Compatibility Package</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NSCP5com</td>
<td>zh_TW.BIG5 localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>NSCPPhcom</td>
<td>Traditional Chinese localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>NSCPPhucom</td>
<td>zh_TW.UTF-8 localization of Netscape Communicator 4.7 supporting International security.</td>
</tr>
<tr>
<td>SUNW5adi</td>
<td>Traditional Chinese Localizations for admintool and GUI install.</td>
</tr>
<tr>
<td>SUNW5adma</td>
<td>Traditional Chinese Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNW5adi packages for Localization.</td>
</tr>
<tr>
<td>SUNW5dab</td>
<td>Traditional Chinese Localizations for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNW5dbas</td>
<td>Traditional Chinese Localizations for CDE Base functionality</td>
</tr>
<tr>
<td>SUNW5ddst</td>
<td>Traditional Chinese Localizations for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNW5ddte</td>
<td>Traditional Chinese Localizations for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNW5dezt</td>
<td>Traditional Chinese (BIG5) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>SUNW5dfit</td>
<td>Traditional Chinese Localizations for CDE Fonts</td>
</tr>
<tr>
<td>SUNW5dhe</td>
<td>Traditional Chinese Localizations for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNW5dhev</td>
<td>Traditional Chinese CDE Help Volumes</td>
</tr>
<tr>
<td>SUNW5dhez</td>
<td>Traditional Chinese (Common BIG5) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNW5dicn</td>
<td>Traditional Chinese Localizations for CDE Icons</td>
</tr>
<tr>
<td>SUNW5dim</td>
<td>Traditional Chinese Localizations for CDE Imagetool</td>
</tr>
<tr>
<td>SUNW5dwm</td>
<td>Traditional Chinese Localizations for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNW5leue</td>
<td>This package contains Traditional Chinese Language Environment specific files. It is a required package to run Traditional Chinese BIG5 Language Environment.</td>
</tr>
<tr>
<td>SUNW5odte</td>
<td>Traditional Chinese BIG5 Core OPENLOOK Desktop Package</td>
</tr>
<tr>
<td>SUNW5pmw</td>
<td>Traditional Chinese BIG5 Localization for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNW5rdm</td>
<td>Taiwanese (BIG5) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>Package Code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUNW5sadl</td>
<td>Traditional Chinese Localizations for Solstice Admintool launcher and associated libraries.</td>
</tr>
<tr>
<td>SUNW5ttfe</td>
<td>Traditional Chinese True Type Fonts Package Extension</td>
</tr>
<tr>
<td>SUNW5udc</td>
<td>Traditional Chinese (BIG5) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNW5xfnt</td>
<td>Traditional Chinese BIG5 X Windows Platform required Fonts Package</td>
</tr>
<tr>
<td>SUNWhadis</td>
<td>Traditional Chinese (EUC) Localizations for Admintool and GUI install.</td>
</tr>
<tr>
<td>SUNWhadma</td>
<td>Traditional Chinese (EUC) Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNWhadis packages for Traditional Chinese (EUC) localization.</td>
</tr>
<tr>
<td>SUNWhbcp</td>
<td>This package contains Traditional Chinese Language Environment binary compatibility files.</td>
</tr>
<tr>
<td>SUNWhdab</td>
<td>Traditional Chinese Localizations for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNWhdbas</td>
<td>Traditional Chinese Localizations for CDE Base functionality</td>
</tr>
<tr>
<td>SUNWhddst</td>
<td>Traditional Chinese Localizations for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWhddte</td>
<td>Traditional Chinese Localizations for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWhdez</td>
<td>Traditional Chinese (EUC) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>SUNWhdft</td>
<td>Traditional Chinese Localizations for CDE Fonts</td>
</tr>
<tr>
<td>SUNWhdhe</td>
<td>Traditional Chinese Localizations for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNWhdhev</td>
<td>Traditional Chinese CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWhdhez</td>
<td>Traditional Chinese (Common) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWhdicn</td>
<td>Traditional Chinese Localizations for CDE Icons</td>
</tr>
<tr>
<td>SUNWhdim</td>
<td>Traditional Chinese Localizations for CDE Imagetool</td>
</tr>
<tr>
<td>SUNWhdwm</td>
<td>Traditional Chinese Localizations for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNWhepmw</td>
<td>Traditional Chinese (EUC) Localization for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWhervl</td>
<td>Traditional Chinese (EUC) SunVideo Runtime Support Software</td>
</tr>
<tr>
<td>SUNWhexir</td>
<td>Traditional Chinese (EUC) XIL Runtime Environment</td>
</tr>
<tr>
<td>SUNWhj2p</td>
<td>Traditional Chinese localization of Java Plug-in 1.2.2</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUNWhj2rt</td>
<td>Java virtual machine and core class libraries (Traditional Chinese supplement)</td>
</tr>
<tr>
<td>SUNWhjvdev</td>
<td>Traditional Chinese Localizations for JavaVM developers package</td>
</tr>
<tr>
<td>SUNWhjvrt</td>
<td>Traditional Chinese Localizations for JavaVM run time environment</td>
</tr>
<tr>
<td>SUNWhksr</td>
<td>Traditional Chinese (EUC) KCMS Runtime Environment</td>
</tr>
<tr>
<td>SUNWhl</td>
<td>This package contains the stream modules for Traditional Chinese Language Environment. It is a required package to run Traditional Chinese Language Environment.</td>
</tr>
<tr>
<td>SUNWhlerx</td>
<td>Stream modules for Traditional Chinese Language Environment. It is a required package to run Traditional Chinese Language Environment (64-bit).</td>
</tr>
<tr>
<td>SUNWhleue</td>
<td>This package contains Traditional Chinese Language Environment specific files. It is a required package to run Traditional Chinese Language Environment.</td>
</tr>
<tr>
<td>SUNWhoaud</td>
<td>Traditional Chinese OPENLOOK Audio Applications Package</td>
</tr>
<tr>
<td>SUNWhodc</td>
<td>Traditional Chinese OPENLOOK Document and Help Viewer Applications Package</td>
</tr>
<tr>
<td>SUNWhodem</td>
<td>Traditional Chinese OPENLOOK Demo Programs Package</td>
</tr>
<tr>
<td>SUNWhodst</td>
<td>Traditional Chinese OPENLOOK Deskset Tools Package</td>
</tr>
<tr>
<td>SUNWhodte</td>
<td>Traditional Chinese Core OPENLOOK Desktop Package</td>
</tr>
<tr>
<td>SUNWhoimt</td>
<td>Traditional Chinese OPENLOOK Imagetool Package</td>
</tr>
<tr>
<td>SUNWhoman</td>
<td>Traditional Chinese OPENLOOK Toolkit/Desktop Users Man Pages Package</td>
</tr>
<tr>
<td>SUNWhorte</td>
<td>Traditional Chinese OPENLOOK Toolkits Runtime Environment Package</td>
</tr>
<tr>
<td>SUNWhrdm</td>
<td>Taiwanese (EUC) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWhreg</td>
<td>Traditional Chinese Localizations for Solaris User Registration</td>
</tr>
<tr>
<td>SUNWhsadl</td>
<td>Traditional Chinese (EUC) Localizations for Solstice Admintool launcher and associated libraries.</td>
</tr>
<tr>
<td>SUNWhltlk</td>
<td>Traditional Chinese ToolTalk Runtime Package Package</td>
</tr>
<tr>
<td>SUNWhttfe</td>
<td>Traditional Chinese True Type optional Fonts Package Extension</td>
</tr>
<tr>
<td>SUNWhuada</td>
<td>Traditional Chinese (UTF-8) Localizations for Software used to perform system administration tasks. Admintool requires both this and SUNW5adi packages for Localization.</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUNWhuadi</td>
<td>Traditional Chinese (UTF-8) Localizations for admintool and GUI install.</td>
</tr>
<tr>
<td>SUNWhubas</td>
<td>Traditional Chinese (UTF-8) Localizations for CDE Base functionality</td>
</tr>
<tr>
<td>SUNWhuccd</td>
<td>This package contains Traditional Chinese Console Display Environment specific files. It is a required package to run Traditional Chinese Console Display Environment</td>
</tr>
<tr>
<td>SUNWhudab</td>
<td>Traditional Chinese (UTF-8) Localizations for CDE Desktop Application Builder</td>
</tr>
<tr>
<td>SUNWhudc</td>
<td>Traditional Chinese (EUC) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNWhudez</td>
<td>Traditional Chinese (UTF-8) Localizations for Desktop Power Pack Applications</td>
</tr>
<tr>
<td>SUNWhudft</td>
<td>Traditional Chinese (UTF-8) Localizations for CDE Fonts</td>
</tr>
<tr>
<td>SUNWhudhe</td>
<td>Traditional Chinese (UTF-8) Localizations for CDE Help Runtime environment</td>
</tr>
<tr>
<td>SUNWhudhv</td>
<td>Traditional Chinese (UTF-8) CDE Help Volumes</td>
</tr>
<tr>
<td>SUNWhudhz</td>
<td>Traditional Chinese (Common UTF-8) Localizations for Desktop Power Pack Help Volumes</td>
</tr>
<tr>
<td>SUNWhudic</td>
<td>Traditional Chinese (UTF-8) Localizations for CDE Icons</td>
</tr>
<tr>
<td>SUNWhudim</td>
<td>Traditional Chinese (UTF-8) Localizations for CDE Imagetool</td>
</tr>
<tr>
<td>SUNWhudst</td>
<td>Traditional Chinese (UTF-8) Localizations for CDE Desktop Applications</td>
</tr>
<tr>
<td>SUNWhudte</td>
<td>Traditional Chinese (UTF-8) Localizations for CDE Desktop Login Environment</td>
</tr>
<tr>
<td>SUNWhudwm</td>
<td>Traditional Chinese (UTF-8) Localizations for CDE Desktop Window Manager</td>
</tr>
<tr>
<td>SUNWhulee</td>
<td>This package contains Traditional Chinese (UTF-8) Language Environment specific files. It is a required package to run Traditional Chinese UTF-8 Language Environment.</td>
</tr>
<tr>
<td>SUNWhuodt</td>
<td>Traditional Chinese UTF-8 Core OPENLOOK Desktop Package</td>
</tr>
<tr>
<td>SUNWhupmw</td>
<td>Traditional Chinese UTF-8 Localization for Power Management OW Utilities</td>
</tr>
<tr>
<td>SUNWhurdm</td>
<td>Taiwanese (UTF-8) OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWhusad</td>
<td>Traditional Chinese (UTF-8) Localizations for Solstice Admintool launcher and associated libraries.</td>
</tr>
</tbody>
</table>
### TABLE C–9  Traditional Chinese (continued)

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWhuudc</td>
<td>Traditional Chinese (UTF-8) Localizations for User Defined Character tool for Solaris CDE environment</td>
</tr>
<tr>
<td>SUNWhwbcp</td>
<td>Traditional Chinese OpenWindows Binary Compatibility Package</td>
</tr>
<tr>
<td>SUNWhwsr</td>
<td>Traditional Chinese prodreg 2.0 localizable text resources</td>
</tr>
<tr>
<td>SUNWhxe</td>
<td>Traditional Chinese X Windows Platform Software Package</td>
</tr>
<tr>
<td>SUNWhxman</td>
<td>Traditional Chinese X Windows Online User Man Pages Package</td>
</tr>
</tbody>
</table>

### TABLE C–10  Shared

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWabcp</td>
<td>Asian common files for SunOS 4.x Binary Compatibility</td>
</tr>
<tr>
<td>SUNWerdm</td>
<td>OILBN ReadMe Directory</td>
</tr>
<tr>
<td>SUNWudct</td>
<td>User Defined Character tool for Solaris CDE environment</td>
</tr>
</tbody>
</table>
Index

16-bit Unicode 3.0 codeset 150
32-bit STREAMS 108
64-bit STREAMS 109

A
alphabets 28
APIs 41, 47
  using to develop applications 37
applications
  FontSet/XmFontList definitions 117
  internationalizing 117
  linking to system libraries 38–39
  XPG4 41
Asian
  printing support 159
AttrObject 132

B
base language 21
Bi-directionality 129
Big-5
  codeset 33
bin/stty 112
/bin/stty directory 112
books@sun.com 16
bopomofo in Chinese 30
breve 62

C
caron 62
catgets() 47
CDE 123
  en_US.UTF-8 locale support of 86
  input methods 87
Central European languages, character
  support 86
character classification macros 40–41
character shaping 129
character support 86
character transformation macros 40–41
characters
  number 28
Chinese text
  bopomofo 30
  linguistic introduction 29
  pinyin 30
  zhuyin 30
code conversion STREAMS modules 109
code conversions 113
codeset
  Big-5 33
  character support 86
  Extended UNIX Code (EUC) 33
  Shift-JIS 33
Codeset Independence 34
commands
  CSI-capable 34
Common Desktop Environment
  Internationalization
    Programmer’s Guide 123
complex language shaping 129
Complex Text Layout (CTL)
CTL 129
Compose Key 30
Compose Sequence 97–98
Compose Sequences
  Latin-1 90, 94
  Latin-2 94
  Latin-4 97
  Latin-5 97, 111
Compose Sequences, for locales 62
Context 133
conversion
  multibyte and wide character process code 41
conversions 113
country of use 21
creating
  message catalogs 47
Creating Worldwide Software 16, 31
.csfrc 112
CSI, Codeset Independence
CSI-capable commands 34
CSI-enabled libraries 36
CSText 134
CTL architecture 130
type
  macros 40
currency 21
  presentation order of 26
  sizes of 27
  units of 26
currency symbol 62
Cyrillic input mode 97
Czech
  character support 86
  keyboards 62
D
date formats 25
Daylight Savings Time (DST) 24
decimal places 25
degree symbol 62
delimiters
  numeric 26
  thousands 25
  word 28
desktop environments 123
desktop layers 123
deutsche mark 26
developer’s cluster 86
diacritical marks 62
  in English input mode 89
diacritics 129
diaeresis 62
dollar 26
doubleacute 62
DST (Daylight Savings Time) 24
Dt Apps 130
dtmail 127
dtterm 110
dynamic linking 38–39
dynamic text widgets 129
E
Editing behavior 133
en_US.UTF-8
  code conversions 113
  fontset definitions 117, 119
  overview 53
English
  character support 86
  input mode 89
Euro currency 20
European printing support 157
Extended UNIX Code (EUC) 33
F
file code 34
fonts
  across different platforms 124
  adding or removing 66
  formats 66
  location 66
FontSet definitions 117, 119
FontSet/XmFontList definitions 117
formats
  currency 26
  dates 25
  numeric 25
  time 24
franc 26
Full Solaris locale 22
**G**
- genmsg utility 47–48
- German character support 86
- GMT offset 24
- Greek character support 86
- input mode 98
- Greenwich Mean Time offset 24

**H**
- Hangul in Korean 29
- Hanja in Korean 29
- Hanzi in Chinese 29
- Hiragana in Japanese 28
- Horizontal Tabs 152
- Hungarian character support 86
- keyboards 62

**I**
- IA keyboards 63
- IBM DOS 437 64
- iconv
  - command 113
  - Japanese character code conversion 77
- input modes
  - Cyrillic 97
  - English 89
  - Greek 98
- internationalization
  - ISO Latin-1 21
  - Java 34
- Internationalization 20
- internationalization APIs 41, 47
- internationalizing applications 117
- ISO 8859-n character support 86
- ISO Latin-1 21
- ISO-10646 86

**J**
- Japanese text
  - Hiragana 28
  - Kanji 28
- Katakana 28
- linguistic introduction 28
- Java internationalization 34

**K**
- Kanji in Japanese 28
- Katakana in Japanese 28
- key Compose Sequences 62
- Keyboard Selection 154
- keyboards 30, 62
  - Changing keyboards on IA 63
  - Changing on SPARC 63
  - Czech 62
  - Hungarian 62
  - Latvian 62
  - Lithuanian 62
  - Polish 62
  - Turkish 62
- Korean text
  - Hangul 29
  - Hanja 29
  - linguistic introduction 29
  - krona 26
  - krone 26
  - kroner 26
  - KSC-5700 80

**L**
- LANG 108
- LANG environment variable 108, 124
- language 21
- Language Conversion Library 127
- language engine 129
- language-dependent rendering 24
- Latin-1 Compose Sequences 94
- Latin-2 Compose Sequences 94
- Latin-4 Compose Sequences 97
- Latin-5 Compose Sequences 97, 111
- Latin-n terminals 111
- Latvian keyboards 62
- Layout behavior 133
- Layout Direction 148
- Layout Modifier Orientation 133
- Layout Services 132
- layoutDirection 148
- LayoutObject 131–132
LC_ALL 21
LC_COLLATE 23
LC_CTYPE 23
LC_MESSAGES 23–24
LC_MONETARY 23
LC_NUMERIC 23
LC_TIME 23
LCL 127
left-character() 136
libc 38–39, 41
libraries, linking applications to 38–39
Ligation 129
ligatures 129
linking applications 38–39
lira 26
list separators 26
Lithuanian keyboards 62
LO_LTYPE 24
loading
  STREAMS modules 109–110
locale 21
locale utility 108
locale(1) 108
locales 20–22
categories of 23
  Compose Sequences 62
database 33, 37
  environment variables 108, 124
  what is... 21
localization 20
localization resource category 122
@ls numerals=:national 133
@ls numerals=nominal:national 133

M
m_create_layout() 132
macros
cctype 40–41
mail interchange 126
markka 26
mbtowcs 41
mbtowc 41
message catalogs, creating 47
modinfo command 110
modload command 110
Mouse Selection 153

mp(1) 157
multi-byte Unicode representation 53
multibyte file code 41
mystreams file 112

N
NULL (0x00) 34
number of characters 28
Numbers 25
numeral shaping 130
Numerals 149
NUMERALS_CONTEXTUAL 149
NUMERALS_NATIONAL 133
NUMERALS_NOMINAL 133
numeric conventions 25

O
ogonek 62
OpenWindows
  changes 127
order for sorting 28
Orientation 133
OSF/Motif Programmer’s Guide 17
OSF/Motif Programmer’s Reference 17
OutToInp 135

P
Page Description Language (PDL)
  interpreters 122
page sizes 30
paper sizes 30
PDL interpreters 122
People’s Republic of China 30
peseta 26
pinyin in Chinese 30
PLS 129
Polish
  character support 86
  keyboards 62
Portable Layout Services (PLS)
  PLS 129
Porting Instructions 155
positional variation 130
POSIX 123
postscript(1) 157
PostScript 121
  support under Solaris 157
PostScript Language Reference Manual 17, 121
  Supplement 17, 121
pound 26
printing support
  Asian 159
  European 157
Programming the Display PostScript System
  with X 17, 121
Property 135
pseudo-XOC 131

R
radix 25
  radix characters 25
region 21
Render Table 151
Rendition 149
Russian
  character support 86

S
saving
  STREAMS modules settings 112
sbin/sh 38
/sbin/sh command 38
Scandinavian and Baltic language character
  support 86
script selection 88
segment ordering 129
separators
  list 26
  thousands 25
  word 28
setenv command 108
setlocale man page 108
setting
  terminal options 112
setup
  TTY environment 108
ShapeCharset 133, 149
Shift-JIS codeset 33

Index-219

shortcuts. Compose Sequences
Simple Mail Transfer Protocol 127
single-display clients 124
Slash (0x2f) 34
Smallberg, David 16, 31
SMTP 127
Solaris
  Asian 67
  base product 86
  Chinese 69
  contents 49
  Japanese 75
  Japanese printing support 159
  Korean 79
sort order 28
Spanish
  character support 86
SPARC keyboards 63
standards
  interface 123
  internationalization 123
stateless file code encodings 34
static and dynamic text 129
static linking 38
strchg command 111
strconf command 111
STREAMS modules
  loading 109–110
  saving settings 112
String validation 129
String XmTextFieldGetLayoutModifier 144
stty command 112
stub entry points, in libw and libintl 39
su command 109
Symmetrical swapping 129
system libraries
  linking applications to 38–39

T
tabbing 129
terminal options, setting 112
terminal support for Latin-1, Latin-2, or
  KOI8-R 111
terminals
  Latin-n 111
  Latin-n terminals 111
X
X Display PostScript 121
X Window System 121
X/DPS 121
X/Open-Uniforum Joint Internationalization Working Group 86

X
Xetops 159
XFontStruct 125
Xlib dependencies 125
XmALIGNMENT_CENTER 134
XmALIGNMENT_END 134
XmCR_MOVING_INSERT_CURSOR 136–137
XmDEFAULT_DIRECTION 131
XmDirection 132, 147
XmEDIT_LOGICAL 134, 137–138, 153
XmEDIT_VISUAL 134, 137, 153
XmFont_IS_XO 132
XmFONT_IS_XOC 132, 152
XmFontSet 125
XmLabel 131, 149
XmLabelG 131
XmList 131
XmNalignment 134, 148
XmNAlignment 149
XmNeditPolicy 134, 137, 153
XmNfont 132
XmNfontName 132
XmNfontType 132–133
XmNgainPrimaryCallback 136–137
XmNlabelString 149
XmNlayoutAttrObject 132
XmNlayoutDirection 130–131, 133, 147–148
XmNlayoutModifier 131–133, 148–149
XmNmotionVerifyCallback 136–137
XmNrenderTable 134, 154
XmNrenditionTag 134
XmNrenditionTableAddRenditions 151
XmNrendition 131–134, 147–148
XmNrendition{Retrieve,Update} 133
XmString 131, 147
XmSTRING_COMPONENT_DIRECTION 131
XmSTRING_COMPONENT_LAYOUT_PUSH 131
XmSTRING_COMPONENT_LOCALE_TEXT 131
XmSTRING_COMPONENT_TEXT 131
XmSTRING_COMPONENT_WIDECHAR_TEXT 131
XmStringDirection 131, 147
XmStringDirectionCreate 147

X

U
u8lat1 STREAMS module 111
u8lat2 STREAMS module 111
UIL 147
Unicode 3.0 53
support 86
Universal Character Set Transformation Format for 8 bits encoding UTF-8 encoding
/usr/ucb/stty 112
/usr/ucb/stty directory 112
UTF-8 encoding 86
utilities
genmsg 47–48
locale 108

W
cwestombs 41
wctomb 41
Western European alphabets 28
Western European languages, character support 86
wide character expression 33
process code 41
words
   delimiters 28
   order of 64

International Language Environments Guide ♦ February 2000
XmText  131, 134, 148
XmTextField  131, 134, 145
XmTextFieldGetLayoutModifier  144
XmTextFieldSetLayoutModifier  146
XmTextGetLayoutModifier  145
XmTextSetLayoutModifier  146
XoJIG  86
XPG4 applications  41
xutops  159

XView toolkit  127

Y
yen  27

Z
zhuyin in Chinese  30