



# Localization Style Guide for Korean

---

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

부품번호: 806-6238-01  
January 2001

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. 모든 권리는 저작권자의 소유입니다. This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

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

Sun, Sun Microsystems, the Sun logo, SunDocs, Java, the Java Coffee Cup logo, AnswerBook, AnswerBook2, HotJava and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. Netscape Navigator is a trademark of Netscape Communications Corporation.

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

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

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

---

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, Californie 94303-4900 Etats-Unis. Tous droits réservés.

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

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

Sun, Sun Microsystems, le logo Sun, SunDocs, Java, le logo Java Coffee Cup, AnswerBook, AnswerBook2, HotJava et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. Netscape Navigator est une marque de Netscape Communications Corporation.

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

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



# 목차

---

## **Preface 9**

### **1. Translation Guidelines 13**

#### 1.1 Introduction 13

#### 1.2 Some Basic Rules 14

#### 1.3 Stylistic Guidelines 15

##### 1.3.1 Preferred terms for operational verbs 16

#### 1.4 Linguistic Guidelines 17

##### 1.4.1 Grammatical Issues 17

##### 1.4.2 Capitalization 19

##### 1.4.3 Acronyms and abbreviations 19

##### 1.4.4 Use of numerals 20

##### 1.4.5 Punctuation 20

#### 1.5 Country Specific Guidelines 22

##### 1.5.1 Date 22

##### 1.5.2 Time 22

##### 1.5.3 Measurements 22

##### 1.5.4 Numbers and Numerical Separators 23

##### 1.5.5 Addresses 23

1.5.6	Standard phrases	24
1.6	Terminology	24
1.6.1	Glossary	24
1.6.2	Trademark	25
1.7	Don't translate....	26
<b>2.</b>	<b>Messages</b>	<b>27</b>
2.1	Introduction	27
2.2	General Guidelines	28
2.2.1	File Naming and Delivery	28
2.2.2	File Content	29
2.3	C/C++ Program Messages	30
2.3.1	Conversion Specifications	31
2.3.2	Order of Variables	32
2.3.3	Escaped characters	33
2.3.4	Formatting Variables	33
2.4	POSIX message files (.po)	35
2.5	XPG4 Message Files (.msg)	36
2.6	Java program messages (.java)	37
2.7	Java Properties files (.properties)	39
2.8	CDE Resource Files (.tmsg)	42
2.9	Other file types	42
2.9.1	DOS Batch files (.bat files)	42
2.10	Other Issues	43
2.10.1	Character Sets	43
2.10.2	Filenames and extensions	44
2.10.3	Checklist	44
<b>3.</b>	<b>Graphics</b>	<b>45</b>
3.1	Introduction	45

3.2	Bitmapped Images	46
3.2.1	File Formats	46
3.2.2	Guidelines	47
3.3	Scalable Vector Descriptions	49
3.3.1	File Formats	49
3.3.2	Scalable Fonts	50
3.3.3	Guidelines	50
4.	<b>CDE Help (HelpTag Format)</b>	<b>53</b>
4.1	Introduction	53
4.2	Translating CDE Help	54
4.2.1	Overview	54
4.2.2	What should be translated?	54
4.2.3	Formatting tags	56
4.2.4	Filenames and extensions	56
4.3	Components of Sun Help Files	56
4.3.1	Topic titles	56
4.3.2	First and second level headings	56
4.3.3	Step titles	56
4.3.4	Numbered and bulleted lists	57
4.3.5	Related topics	57
4.3.6	Glossary	57
4.3.7	Formatting of software references	57
5.	<b>HTML Files</b>	<b>59</b>
5.1	Introduction	59
5.2	Localized HTML Files	59
5.2.1	HTML Header Section	59
5.2.2	META Tags	61
5.2.3	Charset Encodings	61

5.2.4 HTML Body 61

**6. SolBook SGML 63**

6.1 Introduction 63

6.2 Components of Sun Documentation 63

6.2.1 Title page and cover page 63

6.2.2 Part numbers 64

6.2.3 Release dates 64

6.2.4 Phone and fax numbers 64

6.2.5 Copyright page 64

6.2.6 Table of contents 65

6.2.7 Extended characters 65

6.2.8 Chapter titles 65

6.2.9 Headings 65

6.2.10 Step titles 66

6.2.11 Numbered and bulleted lists 66

6.2.12 Formatting 66

6.3 Localizing Sun Documentation 66

6.3.1 Overview 66

6.3.2 Directory/File structure 67

6.3.3 Translating the .book file 68

6.3.4 Remove MS-DOS end-of-line characters 69

6.3.5 Formatting and layout 69

6.3.6 Typographic Conventions 69

6.3.7 Verify SGML code 70

6.3.8 Updating Entities 71

6.3.9 Filenames and extensions 71

6.3.10 Graphics 71

6.3.11 Checklist 72

6.4	Cross References	72
6.4.1	Adept	72
6.4.2	FrameMaker	73
6.4.3	References to other applications, manuals, and operating systems	73
6.5	Glossary	73
6.6	Index	73
A.	<b>Standard Terminology</b>	<b>77</b>
A.1	Book Organization	77
B.	<b>Sun Trademarks</b>	<b>79</b>
B.1	Book Organization	79
C.	<b>Localized Book Titles</b>	<b>81</b>
C.1	Solaris 8 Installation Collection	81
C.2	Solaris 8 User Collection	82
	용어해설	<b>83</b>
	색인	<b>85</b>





# Preface

---

*Localization Style Guide for Korean* covers standard linguistic guidelines and highlights the conventions and linguistic issues related to the translation and localization of software. It also helps facilities and documentation of Sun products.

---

## Who Should Use This Book

This book is intended for translation and localization partners of Sun Microsystems, including actual translators, localization engineers, and product coordinators. It is also intended for similar personnel within Sun itself.

Linguistic reviewers will find guidelines to help them apply consistent standards in the review process. Desktop publishing personnel will also find information on typographical conventions and the structure of files.

The focus of the book is the delivery of "correct" files – correct in format as well as in content. This has required the collaboration of many individuals. As work has proceeded on it, it has become clear that it spans many disciplines, including: linguistics, journalism, programming languages, graphics, Internet application development, et cetera. As a result, the body knowledge continues to grow.

There are sections of this book which are noticeably brief. These include the discussions of: CDE Helptag and HTML. These sections will be expanded as demand for more information and experience dictate. The editors of this book are also interested in hearing feedback on issues which readers of this book feel are important. This feedback will also play a role in future development of the book.

---

주: Distribution of this book is limited to Sun Microsystems and its localization and translation partners. Frequent project updates and special project requirements dictate that partners stay in close communication with their Sun localization project contacts. Revisions of this book will be made available through these contacts.

---

---

## Before You Read This Book

If you are a localization engineer, you are encouraged to become familiar with the following books:

- Tuthill, Bill and David Smallberg. *Creating Worldwide Software*, Mountain View, CA: Sun Microsystems, Inc., 1997.
- *International Language Environments Guide*, Part # 806-0169-10, Palo Alto, CA: Sun Microsystems, Inc., 2000.

Readers should be familiar the fundamentals of editing text files on a UNIX system.

---

## How This Book is Organized

The book is divided up into 6 main chapters and multiple appendices that give vendors guidelines when translating. The books is composed of the following:

Chapter 1, "Translation Guidelines," deals with stylistic and linguistic elements and considerations when translating to Korean.

Chapter 2, "Messages", deals with specific guidelines for translating software components and message formats in Sun products.

Chapter 3, "Graphics", displays the different types of graphics files that translators must deal with, including pixel-based, TIFF, GIF, and EPS images.

Chapter 4, "HelpTag" is to provide translators aid when translating the Help components of Sun products. HelpTag is a markup language, much like SGML, that is used by the CDE (Common Desktop Environment) to help the on-line viewer.

Chapter 5, "HTML Files" is to help translators when dealing with hypertext on the world wide web. It includes headers, META tags, and charset encodings.

Chapter 6, "SolBook SGML" gives specific guidelines to translators about Sun products' documentation. The SolBook deals with the SGML files of Sun's SolBook DTD.

Glossary is a list of words and phrases found in this book and their definitions.

---

## Related Books

These books are related to the tasks that are being described in this book and may be useful resources.

- *A Style Guide for the Computer Industry*, Mountain View, CA: Sun Microsystems, Inc., 1996.
- *CDE Style Guide*, Part # 806-1360-10, Palo Alto, CA: Sun Microsystems, Inc., 2000.
- *International Language Environments Guide*, Part # 806-0169-10, Palo Alto, CA: Sun Microsystems, Inc., 2000.

---

## 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<sup>SM</sup> web site enables you to have access to 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>.

---

## Typographic Conventions

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

## ☞ P-1 Typographic Conventions

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

## Shell Prompts in Command Examples

A *shell* is a command interpreter that reads user-typed commands, and provides an environment in which to execute UNIX programs. A number of different UNIX shells are available on Solaris.

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

## ☞ P-2 Shell Prompts

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

# Translation Guidelines

---

---

## 1.1 Introduction

This guide provides Sun's translation vendors with standard linguistic guidelines which highlight the conventions and linguistic issues related to the translation and localization of software.

The majority of Sun documents are highly technical and are aimed at readers with a high level of technical knowledge and competency. Therefore the translator's task involves a number of processes:

- Understand and convey the information contained in the source text accurately and non-ambiguously.
- Translate clearly and concisely into the target language.
- Respect the readers' technical proficiency.
- Avoid style or terms that could offend the reader.

Follow the basic rules for technical writing in Korean (i.e. consistency of terminology, correct application of grammatical, syntactical and punctuation rules, avoiding of anglicisms, colloquialisms, and jargon).

Good style is synonymous with effective communication. Documents that communicate effectively, reduce costs, and increase customer satisfaction. Documents written in a style that responds to the requirements of the readers result in fewer revisions, fewer calls to customer support, and reduced training needs. A good translation makes readers feel as if the document was written in their native language. Customer satisfaction increases when accurate and functional documentation enables customers to use the product quickly and efficiently.

The overall guidelines ruling any technical translation are that they are correct, clear, concise and consistent. It is important that any platform specific requirements and

standards are adhered to. It is also vital that the functionality of Sun software is not jeopardized and that the look and feel of the user interface is retained.

Sun requires that all of their vendors make quality the highest priority. In order to achieve this, it is important that good communication is established between Sun and their vendors. Any issues are dealt with promptly and measures are put in place to ensure a smooth localization process.

Vendors should check that the material received is adequate for completing the job. If any files or reference materials seem to be missing, they should be queried with Sun. Every translator involved in a project must have all relevant information.

Specific project guidelines, as well as glossaries, must be adhered to. If any queries arise, they should be addressed before the completion of the project.

---

## 1.2 Some Basic Rules

- Keywords and Tags - Don't translate keywords or tags. Both SGML files and HTML files use certain sets of markup tags.

Example 1:

```
<highlights><para>This appendix lists Solaris  
books which are commonly referred to by users and by  
other documentation.</para></highlights>
```

The tags `<highlights>` and `<para>` should not be translated.

Example 2:

```
{ "Supported_Locales", "zh, zh_TW, ko, DE_AT, el_GR, ja"},
```

The string on the right double quotes is indicators of locales; do not translate it.

- Consistency in translation. The same terms should have same translation across all the files and documents.
- Grammar, syntax, punctuation - Make sure localized grammatical, syntactical and punctuation rules are used correctly.
- File and directory names and structures - Don't change file names or directory names and document structures. Tools which automatically and reliably build and test documentation or software frequently rely on correct naming.
- Format - Keep the file format as close to the original document as possible. For graphics, make sure not to change the file format.
- Update - When receiving a new update, don't retranslate the whole document; only change the differences and additions.

## 1.3 Stylistic Guidelines

Here is a list of stylistic guidelines.

### 1. Hardware and software personification

Personification gives living—or human—traits to an inanimate object.

For Korean, when hardware or software is personified, you should omit it and translate in the form of passive voice, making the object the subject.

English	Korean
The program will delete the document.	문서가 삭제됩니다.
The 250 MHz UltraSPARC(TM)-II CPU processor delivers record-breaking database performance.	250 MHz UltraSPARC(TM)-II CPU 프로세서로 인해 데이터베이스 성능이 비약적으로 향상됩니다.

Exceptions to this include:

- confusing subjects
- subjects without action
- explaining the status of subjects

### 2. Interrogatives, exclamations and rhetorical questions

Interrogatives are usually questions that ask questions using “who”, “which”, or “what”. A rhetorical question is one that does not expect an answer. Exclamations are statements that carry a feeling of excitement or alarm.

For Korean, exclamations should be translated into declaratives. Avoid calling users “you”. In many cases, you can simply use “user” or “network administrator” instead.

English	Korean
How do you use an Applet?	애플릿을 어떻게 사용하시겠습니까?
Let's get to know the program!	이 프로그램에 대하여 알아보겠습니다.

### 3. Colloquialisms

Colloquialisms are a type of familiar and informal conversational language. English technical manuals often use colloquialisms which must be avoided in the translation.

For Korean, colloquialisms should be translated into literary styles.

English	Korean
By doing this, you can see the Welcome to JavaViews screen.	그렇게 하면 <b>JavaViews</b> 시작 화면(Welcome to JavaViews 화면)이 나타납니다.
Now you know what to do if you want to proceed with the installation.	설치를 계속하려면 지금 위의 사항을 수행하십시오.

### 1.3.1 Preferred terms for operational verbs

Word usage	Rule for usage	Translation
Click	Use the word "click" when users are clicking on objects with a mouse.	클릭
Press	Use the word "press" when the user is describing a keyboard action.	누르기
Select	This term is interchangeable with the words "Click" and "Press". There are many situations where the word "select" is applicable, such as: select by clicking; select by using arrows; or select by using tab keys. When the situation is obvious, using just the word "select" is permissible.	선택
Point	This word denotes the usage of a mouse pointer on an object. It is used when submenus or sub-submenus are to be opened.  ex. Point [Tool] and click [Terminal] menu command.  ex. Click twice on [View] menu and point to [Code Sets], then click on [Korean].	포인트
Submit	This term is often used in a form and displayed as a button.  ex. After filling out the form, Click the [Submit] button.	제출



## 1.4Linguistic Guidelines

### 1.4.1Grammatical Issues

#### 1. Infinitive style

An infinitive is a verb, in English, that is often preceded by “to” and may be followed by an object.

For Korean, there are no infinitives in the language. Translate them into the style of declaratives or intentional sentences. Do not translate “you” or “the user” into Korean.

English	Korean
Use the script to save the existing configuration to a diskette.	스크립트를 사용하여 기존 구성을 디스켓에 저장합니다.

#### 2. Future tense versus present tense

For Korean, present tense is preferred. When present tense translation creates confusion, future tense translation may be used.

English	Korean
This chapter will describe the procedures to install a client/server system.	이 장에서는 클라이언트와 서버 시스템 설치 절차를 설명합니다.

#### 3. Modal verbs versus imperative

A modal verb is a type of auxiliary verb, such as “can”, “may”, “must”, “might”) whereas the imperative is a noun or verb that expresses command or order.

English	Korean
The data must be saved before proceeding to the next step.	데이터는 다음 단계로 진행하기 전에 반드시 저장되어야 합니다.
Save the data before proceeding to the next step.	다음 단계로 진행하기 전에 데이터를 저장하십시오.

#### 4. 2nd person singular versus impersonal form

In 2nd person singular forms, the sentence involves a reference to the speaker using “you”, “your”, or “yours”. The impersonal form has no reference to a distinct person and no expressed subject.

For Korean, In case of an impersonal subject sentence, leave out the subject. Do not use “you”. Avoid using “you” or “the user” in translation.

English	Korean
Make sure you specify a valid name for the device, and that you have network access rights.	장치의 올바른 이름을 지정하고 네트워크 액세스 권한이 있는지 확인합니다.
Make sure it specifies a valid name for the device, and that it has network access rights.	장치의 올바른 이름을 지정하고 네트워크 액세스 권한이 있는지 확인합니다.

#### 5. Passive voice versus active voice

A passive voice occurs when the subject of the sentence being acted upon by another agent; however with an active voice, the subject and verb relationship is very straightforward.

For Korean, when people are the main focus of an activity, the sentence should be translated into active voice. When explaining an action by the user, use an active voice, without using “you”. When translating an active voice into a passive voice, use the result as the subject.

English	Korean
The adapter card is used as follows:	어댑터 카드는 다음과 같이 사용됩니다.
Use the adapter card as follows.	다음과 같이 어댑터 카드를 사용하십시오.

#### 6. Translation of the -ing form

The -ing form is the present participle of a verb (or also called the “progressive tense”) and it indicates that the action is continuing.

For Korean (depending on context: 1st level heading, 2nd level heading, etc.), the -ing form can be an adjective or noun.

English	Korean
Adding and saving documents to folders.	폴더에 문서 추가 및 저장[need text entity]
Add and save documents to folders.	[need text entity]
Getting started.	Geettiing staarteed.[need text entity]

## 1.4.2 Capitalization

In English, capitalize the first-person singular pronoun “I”, titles, and proper nouns.

In Korean, there are no capitalized words. However, for English trademarks, you should follow the original English style and keep the word as it is in English.

English	Korean
Building a Simple Applet	Buuildiing aa Siimplee Appleet

## 1.4.3 Acronyms and abbreviations

An acronym is a word formed by the initial letters it represents.

Use the corresponding acronym appropriate to Korean. If the meaning of the word is not apparent to the user, write out the entire acronym and its meaning and translation only at the place where it is first mentioned in the book or chapter. If the same acronym is mentioned again, it does not need to be written out.

Avoid abbreviations, but if absolutely needed, only use standard ones. Where abbreviations are necessary due to space reasons, follow your country grammar rules.

You should translate abbreviations that are widely familiar (cm, km, m, kg, g, etc.) into Korean. However, some abbreviations that are not familiar (in., lb., pitch, pt., byte, etc.) should remain as they are.

English	Korean	Meaning
URL	URL (일관된 자원 위치)	Meaning
dpi	dpi	dots per inch
ppi	dpi	pixels per inch
lpi	lpi	lines per inch
WYSIWYG	WYSIWYG (위지윅)	what you see is what you get
MB	MB	megabytes
KB	KB	kilobytes
pt	pt	point
n/a	n/a	not available, or not applicable

#### 1.4.4 Use of numerals

For Korean, always use Arabic numerals. However, 1 or 2 may be translated in the Korean language. Put the object first and then the corresponding number.

English	Korean
5 directories and 12 files	디렉토리 5개와 파일 12개

#### 1.4.5 Punctuation

Punctuation may be determined by the stylesheet used. A stylesheet is a set of decisions made by the writers and editors of a book or document about the usage of product names, numbers, abbreviations, and acronyms, hyphenation, and capitalization. In this way, consistency is maintained amongst the various books written.

##### 1.4.5.1 Comma, period, colon, semicolon

In English, the punctuation marks comma, period, colon, and semicolon are separators.

Rule for Korean:

Colon(:)-at the end of a sentence (not word), it should be replaced with a period (.)

Follow the English convention for all punctuation, except when encountering a colon.

##### 1.4.5.2 Brackets and Parenthesis

There should be no space between parenthesis and the text inside them. In English, the parenthesis only contains the period if it is a full sentence, where the first letter in the sentence is capitalized. Follow the English convention for parenthesis. You should avoid using sentences in brackets whenever possible. It is recommended to use a phrase in a bracket. Comma (,) should come right after the bracket. However, when a sentence comes inside the bracket, comma (,) comes inside them.

Use different brackets for different types of objects. There should be no spacing between characters in Korean.

Symbol	Rule	Example
	Use for window, button, and box titles in dialog boxes.	Netscape Composer 창. 다른 이름으로 저장하기 대화 상자. 확인 버튼을 클릭. IP 주소 상자에 IP 주소를 입력.
“ ”	Use for option entry titles in lists.	설치레벨 드롭다운 목록의 “고급 사용자”를 클릭.
	When bold characters are not available for the document, use these for keyboard key names.	Esc 키를 누름.
	Use this for titles of books or manuals.	Solaris 공통 데스크탑 환경: 사용 설명서 참조
	Use for titles of chapters or sections.	제1장 인쇄 참조
“ ”	Use for messages that appear in the display.	화면에 "ESC키를 누르십시오"가 표시됩니다.

### 1.4.5.3Hyphenation

Translators should avoid inserting manual hyphens. Do not split product names and trademarks (unless necessary due to space restrictions).

In documentation, translators should not insert hyphenation since Sun uses Adept and the page displayed is not WYSIWYG.

Rule for Korean: Do not manually hyphenate localized and English text. The text should wrap automatically.

### 1.4.5.4Spacing

For Korean, characters represent syllables. There should be no spacing between syllables of a word, but there should be spacing between Korean words. English words and Arabic numerals may be following by Korean words (articles); there should be no space between them.

## 1.5Country Specific Guidelines

### 1.5.1Date

Dates may be given in a long (in English, spelled out) form or a short, abbreviated form. The English examples below use U.S. English, which puts the month first, then numeric day of month next, and year last. British English puts the numeric day first, followed by the month. In the long form, it is possible for the two forms to be intermixed in either locale.

In Korean, the year comes first, followed by month, then day.

English	Korean
24 July 2000	2000년 7월 24일
July 24, 2000	2000년 7월 24일

When dates occur in short form, the format order is:

yyyy-mm-dd

English	Korean
07/24/00 (U.S. style)	2000-7-24

### 1.5.2Time

Time format for Korean:

English	Korean
7:35 am	오전 7시 35분
2 pm	오후 2시

### 1.5.3Measurements

English measurements are replaced with metric ones, with a few exceptions (3.5 inch disk and display measurements). The original English units are put in parenthesis.

For Korean, there is no space between the number and the unit of measure.

English	Korean
The monitor weighs 74 lb.	모니터의 무게는 33.6kg입니다.
3.5 inch floppy	3.5 인치 플로피

Units of measurement are usually not followed by a period. Example: cm, MB, MHz, kg.

### 1.5.4 Numbers and Numerical Separators

A decimal point is used to distinguish the ones digit from the one-tenths. Similarly, a comma is used between intervals of a thousand.

For Korean, a comma (,) should be used as a thousand separator and a period (.) as a decimal point. When translating the numbers into Korean, write words separated by spaces in blocks of ten thousand.

English	Korean
1.5 mm	1.5mm
2 million	2백 만
230,000,000,000	230,000,000,000

### 1.5.5 Addresses

If given as a mailing address, do not translate. If the address applies to the language of the published book, then contact main Sun authority for information.

English	Korean
John Smith 23 Malborough Ave San Francisco, CA 98765 USA	Joohn Smiith 23 Maalboorooouugh Aavee Saan Fraanciiscoo, CAA 98765 USA

## 1.5.6 Standard phrases

The following phrases are translated as follows in Sun documents:

English	Korean
Note	주
Important	중요
Caution	주의
See also	관련 항목
Part number	문서 번호
Revision	수정

Older versions of Sun documentation distinguished between “Caution” and “Warning”. Newer document simply uses “Caution.” If “Warning” is encountered, it can be translated the same way as “Caution.” In practice, the translations of these standard phrases are often built into a stylesheet for a particular document type. In any case, the tags of a document, e.g., <warning>, should not be translated.

## 1.6 Terminology

### 1.6.1 Glossary

During the translation process, consistency of terminology within a product and with other Sun products is required. Translators are provided with a Sun master translation glossary for each locale requiring translation. This glossary covers only those terms that are specific or unique to Sun’s business, not every conceivable technical term used in the computer industry. It is intended for use by anyone who performs localization of Sun’s software or hardware products.

Most terms in the translation glossaries are derived from Sun’s own English terminology *Global Glossary*. This glossary covers multiple disciplines pertaining to Sun products, and contains over 2000 terms and definitions. It can be found at <http://docs.sun.com>.

Localization must be done in accordance with the standard Sun translation glossaries. If product specific terminology is not covered by these glossaries or there are additional terminology requirements, then please request the necessary reference



material or glossaries. Product-specific glossaries should be delivered, if possible, in plain text format.

Where there is more than one definition for a term, the discipline is identified in the definition. Even where a term may be unique to a particular discipline, the discipline may be identified to assist in understanding.

The entries in this glossary usually begin with a lowercase letter. Acronyms are all capitalized, unless they traditionally appear in lowercase. Acronyms are usually referenced to the spelled-out definitions.

### 1.6.1.1 Using the glossary

- Translate adhering to Sun's glossary.
- For third party products, refer to product or reference material. (Request from Sun if necessary.)
- Vendors should translate the new terms to their best ability and send them to Sun for validation.
- Approval of the new terms will be decided by Sun internal linguistic testers.
- Please attempt to send queries concerning terminology issues during the life cycle of the project.

### 1.6.2 Trademark

A trademark is a word, phrase, name, symbol, logo or slogan adopted and used by a company to identify and distinguish its products from those of other companies.

Registered trademarks are also not to be translated.

Some names may be used as trade names of the company and also as trademarks for the Sun line of products. For example "Sun" is the trademark of the company and, at the same time, it is used for Sun line of products. Never trademark a trade name. Here follows an example from the Sun copyright page:

English	Korean
The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees.... Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface...	OPEN LOOK 및 Sun™ Graphical User Interface는 Sun Microsystems, Inc.에서 그 사용자 및 사용 허가를 받는 자들을 위해 개발하였으며...Sun은 Xerox로부터 Xerox Graphical User Interface에 대한 비독점적 사용 허가를 받았으며...

The two symbols adopted by Sun are <sup>™</sup> for trademarks and <sup>SM</sup> for service marks. Service marks are typically used in sale or advertising of services whereas trademarks are for manufacturers or intellectual property owners to distinguish their products. These marks are usually registered and protected by law. As a rule, place the symbol after the trademark and if superscript symbols are not available, enclose them in brackets. In SGML/Adept please use the mark-up "trademark" to insert the trademark. For example, the text "Solaris<sup>™</sup> 7" would appear in an SGML file as:

```
<trademark>Solaris</trademark> 7
```

To protect Sun trademarks, they are designated on book spines, covers, self-covers and when they are first mentioned in the text. After the first time a trademark is mentioned, the trademark can be used as an adjective without the symbol.

Some examples of Sun trademarked products and services are given in Appendix B. A list of trademarked products specific to a project can be provided by Sun.

---

## 1.7 Don't translate....

Some terms and text should not to be translated. This includes the classes of terminology listed below. If you are in doubt as to whether to translate a term/phrase, please do not hesitate to contact your project manager at Sun.

- Computer Systems
- Programming source code
- Directory and file names
- Trademarks and service marks

In addition to the above terminology, different types of software files have rules about what not to translate.

## Messages

---

This chapter provides translators with specific guidelines to translate the various message formats for Sun products.

- 절 2.1 “Introduction” (27페이지)
- 절 2.2 “General Guidelines” (28페이지)
- 절 2.3 “C/C++ Program Messages” (30페이지)
- 절 2.4 “POSIX message files (.po)” (35페이지)
- 절 2.5 “XPG4 Message Files (.msg)” (36페이지)
- 절 2.6 “Java program messages (.java)” (37페이지)
- 절 2.7 “Java Properties files (.properties)” (39페이지)
- 절 2.8 “CDE Resource Files (.tmsg)” (42페이지)
- 절 2.9 “Other file types” (42페이지)
- 절 2.10 “Other Issues” (43페이지)

---

### 2.1 Introduction

On a UNIX system such as Sun Solaris, the GUI (*Graphical User Interface*) uses the X Window System (also known as X11), which includes a library or toolkit of widgets (graphical objects visible on the window system) such as buttons, menus, sliders, etc. Instead of using a command line style, GUIs use graphical icons to represent commands. One common widget library is Motif. (The CDE windowing environment is mainly based on Motif). The Motif toolkit and programs are compatible with OpenWindows, so that they have a backward compatibility.

At the lowest level, all GUI applications on Solaris use the X Window System. They can also use libraries other than Motif on top of this to give a different look and feel. An example would be the Swing library, which is a part of the Java Foundation Classes. Some applications, such as Netra, are HTML-based and rely on the underlying web-browser to provide widgets. Finally, some of Sun software is not designed to use GUI at all.

These different ways of presenting software use different file formats for messaging, but they all contain localizable text.

---

## 2.2 General Guidelines

1. Format — Keep the format as close to the original document as possible. Line breaks, spaces, etc. should follow the original document.
2. Translation consistency — The same terms should have the same translation across all the files. Sun maintains a translation glossary for each locale.
3. Updating — When there is a new update, do not retranslate the entire document, only change the differences and/or additions. In such cases, you may receive files that are already partially translated, which leverage work from the previous release.
4. Comments — Do not translate comments. These are used by the original developers and maintainers, or for translators to understand the programming source code. They are invisible to the tools that build software products.

Nearly all message files take the format of: `<key> <value>`. The `<key>` is the piece of text that the application uses to identify individual message strings. The `<value>` is the localizable message text itself. Make sure that key values do not get translated or changed in any way. If changed, the software will not be able to locate the correct message key and the text will appear in English for the application. Or it may cause unexpected behavior in the application.

### 2.2.1 File Naming and Delivery

File names and directory names should not be changed, unless specifically directed to do so. When files return from translation, they are integrated by automatic tools which assume that the file names are correct. The following are some important facts about filenames and delivery:

1. *Case-sensitivity*. UNIX file names are case-sensitive (i.e., upper-case versus lower-case).. The file name `ReadMe` is not the same as `README` or `readme`. Some Windows tools are case sensitive; others are not. Care should be taken not to change case as the file is copied or edited.

2. *Name suffix.* Some Windows tools recognize a name suffix, delimited by a dot (".") and limited to three letters. In such cases, `.html` is truncated to `.htm`, `.tiff` becomes `.tif`, et cetera. The UNIX file system does not recognize the concept of filename suffix. The dot (period) is an ordinary character. Thus, both `.tiff` and `.tif` are acceptable suffixes. So is `.tiff.old.2`. Make sure file names are not be truncated on account of suffix length.

Files are often delivered with suffix `.tar.gz`. The contents of such a file is actually a directory tree of files which has been archived with the `tar` command and compressed with `gzip`.

On a UNIX machine, the directory may be restored by

```
gzip -d -c filename.tar.gz | tar xvf -
```

The directory could be archived and compressed again by

```
tar cvf - ./relative-dirname | gzip -9 > filename.tar.gz
```

The *relative-dirname* is the name of the directory with respect to the current working directory, where the command is being run.

## 2.2.2File Content

The content of files can be classified as either binary or text. Binary files include executable programs, the libraries they work with, compressed files, etc. Text files include program source code, README files, and other files which are operated on by a text editor.

Different operating systems have different conventions for distinguishing a line of text. Some systems prepend a length field before each line, giving how many bytes are in the line. However, the systems of principal interest for graphics and translation (i.e., UNIX, Mac, Windows) use end-of-line terminators.

- UNIX systems, including Solaris, terminate a line with *newline* (hexadecimal 0A).
- MacOS (Apple Macintosh) systems terminate a line with *carriage return* (hexadecimal 0D).
- DOS and Windows systems terminate a line with *carriage return* then *newline* (hexadecimal 0A, 0D).

Text files delivered for use in Solaris should terminate lines with *newline*. In general, converting from DOS to UNIX text on Solaris can be accomplished by:

```
dos2unix -ascii dosfile unixfile
```

Conversely, converting from UNIX to DOS on Solaris can be accomplished by

```
unix2dos -ascii unixfile dosfile
```

Do not forget the `-ascii` option. The command will otherwise convert the file to ISO format, which is not usable.

File *encoding* should conform the codeset specified for a project. The Solaris `iconv` command is able to make conversions between various codesets. Encoding for Java programs and properties files are dealt with separately. (See 절2.6“Java program messages (. java)” (37페이지).)



---

주의: The `dos2unix` command should not be used on binary files. These include GIF, JPEG, etc.

---

## 2.3C/C++ Program Messages

By far the most common language for writing Solaris components is the C programming language. Most of the kernel (the lowest level software layer) of Solaris is written in C; so are most of the libraries and commands. As a result, most Solaris messages originate from C programs. After a program has been internationalized, the message strings are extracted into message files which can then be translated and combined back with the programs for localized operation.

In general, C programming language facilities are also available to C++. Thus, you may see messages which originate from C++, but their origin is actually of little consequence.

To simply print a message like

```
Hello, world!
```

a C program could execute a statement like the following:

```
printf ("Hello, world!\n");
```

To print someone's name instead of `world`, the statement could be generalized to

```
printf ("Hello, %s!\n", name);
```

where `%s` corresponds to the variable `name`. If the value of `name` is "George", then the output appears as

```
Hello, George!
```

The first argument to `printf()` is always a format string, in this case `"Hello, %s!\n"`. The format string may contain conversion specifications (such as `%s`) which correspond to the remaining arguments to `printf()`. It may also contain escaped characters (such as `"\n"`) for special purposes. (In this case, `"\n"` places a newline character at the end of the string.)

Format string (such as `%s`), escaped characters (such as `"\n"`) and variables (name) won't be changed, in this example, only `"Hello,"` should be translated.

## 2.3.1 Conversion Specifications

Conversion specifications determine how a variable is going to be interpreted for formatting purposes. The specification starts with `%` and ends with a single character *op* indicating the type of conversion operation; between them there may be various optional specification elements. The general form of the conversion specification is:

`%[flag][minwidth][.prec][lh]op`

The optional parts of the conversion specification are as follows:

<i>flag</i>	(optional) modifies meaning of main conversion operation. ("-" to left-justify, "0" for zero padding instead of space char, etc.)
<i>minwidth</i>	(optional) is the minimum width of the formatted result
<i>prec</i>	(optional) is the precision for numeric conversions, e.g., digits to right of decimal point
<i>lh</i>	(optional) indicates the size of the argument; for example, "l" for a long integer, "h" for a short one

The conversion operators are described below:

operator	description
d	signed decimal number (integer), e.g., -12345
u	unsigned decimal integer (integer), e.g., 12345
o	unsigned octal number (integer), e.g., 0123
x, X	unsigned hexadecimal number (integer), e.g., 0x12ab or 0X123AB

operator	description
c	single character (converted from an integer), e.g., "c"
s	string of characters (bytes), e.g., "String of stuff"
f	decimal floating-point, fixed fractional precision, e.g., -123.456
e, E	decimal floating-point, fixed fractional precision, e.g., -1.23456e+2 or -1.23456E+2
g, G	uses f or e/E conversion, depending on value
%	single percent character; none of the optional parts of a conversion specification can be used.

## 2.3.2 Order of Variables

Generally, program variables are evaluated in the same order that they occur in the format string. However, when a program is localized to another locale, this may not make sense grammatically. In this case, the place holders must explicitly denote which variables they are associated with.

Consider the following example where `amount` is set to 35, and `kind` is the string `free`.

```
printf("The temperature this %s is %d degrees.", timeofday, degree);
```

This will display the output:

```
The temperature this morning is 35 degrees.
```

For a different locale, this grammatical order of the format string may not be correct. The desired result may be

```
35 degrees is the temperature this morning.
```

But the order of variables is predetermined by `printf()` statement in the compiled C program, i.e., first `amount`, then `kind`. By re-specifying the format string as

```
"%2$d degrees is the temperature this %1$s."
```

the desired result can be achieved. The place holder `%2$s` tells the UNIX messaging system to print the second variable (`kind`) first, and treat it as a string. Then, `%1$d` tells it to print the first variable (`amount`) next as a number.

The locale-specific format string is introduced into the locale by the POSIX or XPG4 messaging scheme.



### 2.3.3 Escaped characters

As mentioned earlier, `printf()` statements can contain escaped characters. The frequently used `\n` denotes a new line. For example, the program segment

```
printf("Hello World")
printf("hello world\n")
printf("Hello mom ..")
```

produces the output

```
Hello Worldhello world
Hello mom ..
```

You can see that the first two `printf()` statements display on the same line, but the third displays on the next line. This is because the second line contains the escaped character `\n`, which forces a new line, which placing the cursor in the first column of the next line.

Listed below are the escaped characters and their meaning:

Character	Meaning
<code>\n</code>	Newline character
<code>\t</code>	Tab character (insert a tab space)
<code>\b</code>	Backspace (overwrite last char)
<code>\r</code>	Return (places cursor at the start of the current line to overwrite it)
<code>\f</code>	Formfeed, moves the cursor to the next line without resetting the position to column 1
<code>\'</code>	Display a single quote in the output
<code>\"</code>	Display a double quote in the output

### 2.3.4 Formatting Variables

#### ■ Minimum field width

A minimum field width is used to set up a column as shown below:

```
%15d
```

will print at least 15 digits, the number is right justified, and preceded with blanks (if necessary). Note that if the integer contains 18 digits then the full number is still displayed since 15 denotes the minimum field width.

#### ■ Precision

A precision (.num) figure limits the number of decimal places that are displayed, for example:

```
% .3f
```

will display a floating point number to three places decimal. Similarly you can use Minimum field width and precision together as follows:

```
%15 .3f
```

This displays the floating point number in a minimum field of 15 digits (including decimal point and three places decimal) right justified with preceeding blanks. Precision only works on types that allow decimal points. However, if applied to a variable type that does not allow decimal places (a string or integer) precision becomes the Maximum filed width instead:

```
% .15s
```

means print the string as normal. If the string contains more than 15 characters, then only display the first fifteen. As a result if you wanted all strings to print out in a column 15 characters wide you would use the following:

```
%15 .15s
```

The minimum field and maximum width is 15.

#### ■ Left justification

The defaults so far all use right justification. If you wish to left justify your output simply place a minus sign after the % symbol, i.e.:

```
%-15s
```

left justifies the string with a minimum length of 15 characters.

#### ■ Plus or Blank

Normally negative numbers are displayed with a leading minus sign (-). All positive numbers are displayed numbers (preceding blank). If you wish to precede a positive number with a plus sign then use the following formatting function call:

```
%+d
```

## 2.4 POSIX message files (.po)

POSIX portable message object (.po files) are human-readable and editable message files which are used with C programs. They contain copies of text strings from ANSI C programs, and can be used for input into the `msgfmt` utility – thus creating a binary form of the message file (.mo file) for the application to use during run-time.

A .po file is created by using the `xgettext` command. The file contains pairs of key and value strings, denoted respectively by `msgid` and `msgstr`. Comment may be added to file, and are denoted by a preceding `#` symbol. Only the string following `msgstr` should be translated.

Since the .po and .mo files are manipulated independently of the original C/C++ program, they have no way of knowing if a parameter should be represented by `%s`, `%d`, or other specification. This means that you could switch around specifications inside messages, compile into an .mo file, and get no errors. It is up to the human translator to ensure that the order of variables is retained.

So for a .po file, the human translator should make sure the order of variables is retained, and only the string following `msgstr` should be translated.

Here is an example of a .po file.

```
# File:main.c, line:2697, textdomain("STOK_OST_ADMIN");
msgid "NULL or garbled pathname"
msgstr "NULL or garbled pathname"
msgid "relative pathname %s not permitted"
msgstr "relative pathname %s not permitted"
msgid "invalid link specification %s"
msgstr "invalid link specification %s"
msgid "extra arguments in link specification %s"
msgstr "extra arguments in link specification %s"
msgid "ftype %c does not match link specification %s"
msgstr "ftype %c does not match link specification %s"
```

When translated, it appears as follows:

```
# File:main.c, line:2697, textdomain("STOK_OST_ADMIN");
msgid "NULL or garbled pathname"
msgstr "NULL 또는 잘못된 경로 이름"
msgid "relative pathname %s not permitted"
msgstr "상대 경로 이름 %s(이)가 허용되지 않음"
msgid "invalid link specification %s"
msgstr "연결 사양 %s 오류"
msgid "extra arguments in link specification %s"
msgstr "연결 사양 %s의 불필요한 인자"
msgid "ftype %c does not match link specification %s"
msgstr "ftype %c(이)가 연결 사양 %s(와)과 일치하지 않습니다."
```

Some additional translation guidelines:

- Don't put an extra SPACE after the `'\'` in the end of one line.

- Make sure to switch the two %d or %s in sentence such as "%d of %d".

For this file format .po, don't translate comments and anything starting with a pound sign (#).

## 2.5XPG4 Message Files (.msg)

Solaris CDE applications use XPG4 message files. The human-readable and editable files (.msg suffix) are converted into binary files (.cat files) by the `genccat` utility of Solaris.

In a .msg file, the line starting with '\$' is a comment line, so it does not need to be translated. The comment is implying that the following messages should not be translated. It is very crucial to follow the instructions in the comments, otherwise problems will occur. Console messages and mail forwarding message should not be translated.

Generally Solaris project has provided the new message indication in the comments line to make it more simple to figure out which need to be translated, for example:

```
$ NEW MESSAGES
1 ...
```

Here is an example.

```
$ error messages on pathnames
$quote "
$set 1
$ error msg when pathname is NULL or garbled
1 "NULL or garbled pathname"
$ error msg a given relative pathname is not permitted
2 "relative pathname %s not permitted"
$ error msg when link specification is invalid
3 "invalid link specification %s"
$ error msg when link specification has too many arguments
4 "extra arguments in link specification %s"
$ error msg when given ftype does not match link specification
5 "ftype %c does not match link specification %s"
```

When translated, it appears as follows:

```
$ error messages on pathnames
$quote "
$set 1
$ error msg when pathname is NULL or garbled
1 "NULL 또는 잘못된 경로 이름"
$ error msg a given relative pathname is not permitted
2 "상대 경로 이름 %s(이)가 허용되지 않음"
$ error msg when link specification is invalid
3 "연결 사양 %s 오류"
$ error msg when link specification has too many arguments
4 "연결 사양 %s의 불필요한 인자"
$ error msg when given ftype does not match link specification
```

5 "ftype %c(이)가 연결 사양 %s(와)과 일치하지 않습니다."

## 2.6 Java program messages (.java)

### *Java Language Keywords*

Java is a language that is widely used for embedded applications and web page applets. It is designed so that an executable Java program may run on any host operating system.

Localizing Java resource files differs from localizing other message files in that work is done on the actual Java programs. Although the emphasis is on key/value pairs, there is a limited amount of program logic and structure in them as well. A key/value pair in a Java resource bundle might look like the following:

```
{"info_copyright1", "Copyright 1999 Sun Microsystems, Inc."},
```

The first string is the key; it should not be changed. The second string is the value; this string should be translated.

Comments in a Java program should not be translated. They may take one of the following forms. (These are the same as for C++ programs.)

- Start with a double slash ("//") and continue to the end of the line. For example,

```
c = a + b; // This is a simple addition.
```

- Start with "/\*" and terminate with "\*/". For example,

```
c = a + b; /* This is a simple addition. */
```

This form of comment can span multiple lines.

An English original version of a Java resource file may look as follows.

```
package com.foo.linkage;
import java.util.*;

/**
 * Localizable resources for "com.foo.linkage" classes
 */

public class LinkageResources_en extends ListResourceBundle {
    static final Object[][] contents = {
        { "NULL or garbled pathname", "NULL or garbled pathname" },
        { "relative pathname {0} not permitted",
          "relative pathname {0} not permitted" },
        { "invalid link specification {0}",
          "invalid link specification {0}" },
    }
}
```

```

        "invalid link specification {0}" },
        { "extra arguments in link specification {0}",
          "extra arguments in link specification {0}" },
        // NEW MESSAGE
        { "ftype {0} does not match link specification {1}",
          "ftype {0} does not match link specification {1}" }
    };

    public Object[][] getContents() {
        return contents;
    }
}

```

The corresponding localized version of the resource file would look as follows.

```

package com.foo.linkage;
import java.util.*;

/**
 * Localizable resources for "com.foo.linkage" classes
 */

public class LinkageResources_ko extends ListResourceBundle {
    static final Object[][] contents = {
        { "NULL or garbled pathname", "NULL 또는 잘못된 경로 이름" },
        { "relative pathname {0} not permitted",
          "상대 경로 이름 {0}(이)가 허용되지 않음" },
        { "invalid link specification {0}", "연결 사양 {0} 오류" },
        { "extra arguments in link specification {0}",
          "연결 사양 {0}의 불필요한 인자" },
        // NEW MESSAGE
        { "ftype {0} does not match link specification {1}",
          "ftype {0}(이)가 연결 사양 {1}(화)과 일치하지 않습니다." }
    };

    public Object[][] getContents() {
        return contents;
    }
}

```

- Do not break long lines into multiple short ones, this will break when we build our product. (The long lines may look odd to you, but our program takes care of wrapping long lines into appropriate length.)
- Be careful with the doublequotes ("). Do not omit them. For example:

```
{ENTER_VALUE_MSG, "Enter a value for \"{0}\": "}
```

There are a total of 4 quotation marks, two of them have the escape character “\” in front of them. This indicates that the quotation characters will actually appear as part of the message. (Note that if the two internal quotes were replaced by locale-specific multi-byte quote characters, then the escape characters would not be necessary.)

- Do not translate files names that look like translatable strings. Example:

```
{BACK_BUTTON_IMAGE, "jdt/OK"}
```

- Do not convert native local language characters into Java Unicode notation (for example, \u1234). Please use native characters.

---

## 2.7 Java Properties files (.properties)

A typical .properties file will contain 'key=value' string pairs, like:

```
file_menu=File
edit_menu=Edit
view_menu=View
actions_menu=Service
help_menu=Help
```

Comments maybe added to the file, and are denoted by a preceding # symbol. Translate only the strings on the right hand side of the '=' sign.

Here is an example of a .properties file.

```
#
#ident "@(#)ResourceBundle.properties 1.2 99/05/12 SMI"
#
# Copyright (c) 1998-1999 by Sun Microsystems, Inc.
# All rights reserved.
#
# Resource bundle for ui package

file_menu=File
edit_menu=Edit
view_menu=View
actions_menu=Service
help_menu=Help

exit_item=Exit
create_item=Create
delete_item=Delete
duplicate_item=Duplicate
properties_item=Properties
filter_item=Filter...
update_item=Refresh

next_button=Next
ok_button=OK
cancel_button=Cancel
reset_button=Reset
help_button=Help
finish_button=Finish
previous_button=Previous

find_label=Find:
steps_label=Steps:

never=Never
```

```

add=Add
delete=Delete
move_up=Move Up
move_down=Move Down

```

When translated, it appears as follows:

```

#
#ident "@(#)ResourceBundle.properties 1.2 99/05/12 SMI"
#
# Copyright (c) 1998-1999 by Sun Microsystems, Inc.
# All rights reserved.
#
# Resource bundle for ui package

file_menu=파일
edit_menu=편집
view_menu=보기
actions_menu=서비스
help_menu=도움말

exit_item=종료
create_item=작성
delete_item=삭제
duplicate_item=복제
properties_item=등록 정보
filter_item=필터...
update_item=갱신

next_button=다음
ok_button=확인
cancel_button=취소
reset_button=재설정
help_button=도움말
finish_button=완료
previous_button=이전

find_label=찾기:
steps_label=단계:

never=사용 안함

add=추가
delete=삭제
move_up=위로 이동
move_down=아래로 이동

```

There are special forms designed for GUI menu items, such as:

```

buttonstyle.label.text=Navigation Buttons:
buttonstyle.items=top=At Top | bottom=At Bottom | left=At Left | right=At Right | none=None

```



Such items should still be view as 'key=value' pairs, except they start from the first '=' sign, and separated by '|', a vertical bar. Therefore, the above example has the following pairs:

```
top=At Top
bottom=At Bottom
left=At Left
right=At Right
none=None
```

The same rule applies —translate only the right hand side of the '=' sign, namely, the words 'At Top', 'At Bottom', 'At Left', 'At Right' and 'None'

If you are not familiar with properties files, please read the details. The properties files (designed by Javasoft) support a large amount of syntax, especially when considering the simplicity of their task. They provide many different ways for writing equivalent properties and have restricted it to three types of lines:

- Comment lines: Lines beginning with a #
- Property lines: Lines of the form "key=value"
- Blank lines: Lines beginning with blanks

Several tips about property files:

- If the value begins with a space, it must be prefixed by a backslash. For example:

```
this.is.wrong= no backslash
this.is.right=\ a backslash
```

Although not required, adding a comment about trailing whitespace preceeding the line makes it much easier to read:

```
this.is.hard.to.read=this line has 2 trailing spaces

# next line has 2 trailing spaces
this.is.better=this line has 2 trailing spaces
```

- If you have a format string and you want to get a single quote, use two single quotes:

```
this.is.wrong=I can't stand property files
this.is.right=I can''t stand property files
```

---

## 2.8CDE Resource Files (.tmsg)

Solaris CDE uses localized resource files of '.tmsg' suffixed files. This format will eventually make X resource files by combining with a '.nls' file, which is an Internationalized file, then generate a '.dt' file, dt actions read localized message string from the '.dt' file (a '.dt' file is not binary format.), for example:

`dtaction SDTimage`

For this format messages, what the translator need to do is translating the source (English) '.tmsg' file into target (Simplified Chinese) '.tmsg' file.

Here is an example of a '.tmsg' file:

```
1 Personal Cards
2 Personal Aliases
3 System Cards
4 System Aliases
5 Hosts
6 Users
```

When translated, it appears as follows:

```
1 개인 카드
2 개인 별명
3 시스템 카드
4 시스템 별명
5 호스트
6 사용자
```

---

## 2.9Other file types

### 2.9.1DOS Batch files (.bat files)

\*.bat files are Windows shell scripts that can run on a DOS window. Some strings show at the console and therefore need translation. Translate in this format:

- Do translate echo lines, for example:

```
echo Creating self-installing executable...
```

but not the word echo.

- Do translate the comments at the top of the file. These are instructions to tell the use how to use this script.
- Do not translate any file names, variable names, etc...for example:

```
if exist %DATADIR%\lib\props\internal
rmdir/s/q %DATADIR%\lib\props\internal
```

---

## 2.10 Other Issues

### 2.10.1 Character Sets

Here is a short list of the common character sets that Sun uses for its localized files. Ensure that you use the correct character set when returning source files to Sun. (See also charset encodings in HTML chapter.)

---

Korean	KSC-5601
Simplified Chinese	GB-2312
Traditional Chinese	CNS-11643
Catalan	ISO-8859-1 (Latin 1)
Czech	ISO-8859-2
French	ISO-8859-1 (Latin 1)
German	ISO-8859-1 (Latin 1)
Hungarian	ISO-8895-2
Italian	ISO-8859-1 (Latin 1)
Polish	ISO-8859-2
Portuguese	ISO-8859-1 (Latin 1)
Russian	ISO-8859-5
Spanish	ISO-8859-1 (Latin 1)
Swedish	ISO-8859-1 (Latin 1)

---

Since Java uses unicode internally, any message files for Java application that do not use the iso-8859-1 character set need to be converted using the native2ascii utility before compiling and testing these in the application that is being localized.

## 2.10.2 Filenames and extensions

Files must be returned with the same filenames into the same directory structure.

## 2.10.3 Checklist

Please make sure that the following points have been verified:

- Date/Number formats are correctly localized.
- All placeholders (e.g., conversion specifications) have been moved as necessary to preserve syntax.
- All files have been proofread and, if applicable, spell-checked.
- All files can be compiled without errors. (This is often a good way to check file format compliance.)
- The directory structure is correct.
- The files have been verified in the running software.
- No keys (of 'key=value' string pairs) have been changed.

Avoid these common mistakes

- missing double quotes at end of string
- inconsistent translations
- wrong file name
- wrong document encoding (Traditional Chinese should use EUC, not Big-5, unless specifically directed.)
- carriage return (often appears in UNIX as "^M") at end of each line (DOS format)
- wrong translation
- corruption of original document format, e.g., multiple lines becomes 1 single long line.
- strings in double quotes not translated

## Graphics

---

---

### 3.1 Introduction

This section provides translators with guidelines for handling images and graphics in Sun products.

Graphics are integral assets to an international languages manual because users can identify more easily with illustrations than with text. The guidelines will increase the ability to understand the graphics and illustrations created in projects.

Although Sun expects its technical authors and illustrators to write and create with an international audience in mind, the resulting works may still exhibit deficiencies, particularly as additional locales are supported. Among the issues:

- Not every reader will read from left to right. A sequence of illustrations needs to have its order clearly marked.
- Avoid symbols using human hand gestures because there is a great chance that the position may be offensive in another culture.
- Avoid symbols using animals, alcohol or related material, historical references, and trendy objects.
- Color may represent different emotions and convey opposite meanings from one country to the next.

Graphics typically can be classified as either bitmapped images or scalable vector descriptions.

---

## 3.2 Bitmapped Images

Bitmapped images are rectangular arrangement of pixels, each usually having a size from 1 to 24 bits per pixel. A larger number of bits per pixel is possible, but is limited to very specialized applications. In the simplest case, each pixel has only one bit. Such graphics can only display black and white, with no shades of grey. A typical PC graphics card uses 8 bits per pixel, allowing it display 256 colors.

The images are sometimes referred to as bitmaps, pixmaps, or raster images. They are stored in a variety of formats, some of which have been adopted by web browsers. Similarly, these formats can be used to save snapshots of a window, other region, or a whole screen.

On Solaris, the principal tool for capture bitmapped images is the snapshot tool of the CDE Image Viewer (`/usr/dt/bin/sdtimage`). It is also possible to use the command `xwd`, an X window dump utility; this tool is more useful when images much be capture programmatically.

Images may also be composed or modified with tools such as Adobe Photoshop or the open source GIMP (GNU Image Manipulation Program).

### 3.2.1 File Formats

Several different file formats exist to represent bitmapped images. The format used for a specific file should correspond to that of the original untranslated material. In general, the format will depend on the publishing process or the target medium. On very rare occasions, the format will be changed if there are very compelling reasons.

Several of the formats incorporate LZW (Lempel-Ziv-Welch) compression, which is covered by U.S. Patent No. 4,558,302, owned by Unisys. Care must be exercised in the choice of tools for image creation and editing. In 1995, Unisys decided that vendors of commercial tools must license the technology from Unisys. Tools from major vendors are generally covered, and will cite on product packaging that the product uses LZW technology under license.

#### 3.2.1.1 TIFF Images (`.tiff`)

TIFF (Tag Image File Format) is a family of raster image formats with a large number of applications. Among them is TIFF Class F, which is used for sending and receiving faxes. The first TIFF specification was published by Aldus Corporation in 1986, after several meetings with scanner manufacturers and software developers.

The final specification, TIFF was published in 1992 by Adobe, and includes LZW compression and clarifications on CCITT Group 3 and Group 4 (fax) encoding.

Although it does not show up in final media, TIFF is the principal format used by Sun for preparing images for its technical publications. Editing is done on TIFF images. Various automated tools are then used to convert the images to their final formats.

### 3.2.1.2 GIF (GIF89a) Images (.gif)

The Graphics Interchange Format (GIF) was created by CompuServe as a means for defining generalized color raster images. The form which is popularly used for animation on the web is GIF89a, specified in 1989. Animation is achieved by allowing multiple images in a file. Each image has a lifetime specified in hundredths of a second. Animation was actually specified in GIF87a, specified in 1987. GIF exhibits fast compression and decompression of images, based on LZW compression.

### 3.2.1.3 JPEG Files

JPEG = Joint Picture Experts Group. More to come.

### 3.2.1.4 BMP Files

BMP files (suffixed .bmp) are raster image ("bitmap") files used on Microsoft platforms.

When porting to a non-Microsoft system, a different format is generally recommended since BMP have no compression support, and there dictates very large file sizes.

## 3.2.2 Guidelines

Here are some considerations concerning the manipulation of bitmapped images.

### 3.2.2.1 Changing Size

When an image appears not be the right size, there is a temptation to change the size within an image editing tool. This is almost invariably means changing resolution — the number of pixels in the image — and is the almost invariably the wrong solution. This is done by either removing pixels, or composing new ones to fill in where they didn't exist.

Whenever possible, size should be manipulated in the target medium. For print, this means rescaling so that pixels are larger or smaller than their default printing size. For web applications, image size and width can usually be specified within page markup, such as HTML.

For print, particularly of snapshots with tools with text labels, the snapshots should generally be rescaled so that the size of characters in the text labels is close to that of running text on the printed page.

### 3.2.2.2 Image Maps

Some applications, particularly interactive web-based ones, involve the user clicking on some part of an image. An image map defines the coordinates of sensitive areas of an image and the actions to be taken as a result of clicking on them.

In the case of localized applications, the coordinates of sensitive area may change and need to be remeasured relative to the origin of the image. For HTML, the origin is the top-left corner the image.

### 3.2.2.3 Callouts

Some images may be diagrammatic and may require explanatory labels (callouts) which point to parts of the image. Unless an image format like GIF or JPEG is the only final format, this is usually a mistake. If the image, along with callouts, are to be part of a printed publication, it should be converted to Encapsulated PostScript. Should this be necessary, notify your Sun project contact so that appropriate changes can be made to the publication in question. In some cases, the material for translation and localization need to be reissued.

### 3.2.2.4 Running Localized Tool

Localized versions of many CDE applications can be displayed either by logging into Solaris in the correct locale, or by starting the tools from a shell command with the correct locale settings. Thus, it is possible to display tools for many locales on the same desktop at the same time. This assumes that:

- the target locales are installed on the system.
- the input engine and fonts are active when the tool is invoked. (Contact the Asian Localization Center through your Sun project contact if you need assistance with this.)

For example, to invoke the Korean Calendar tool from the Bourne shell,

```
% LANG=ko dtcm &
```



---

## 3.3 Scalable Vector Descriptions

Artwork which can be rendered independent of the resolution of the target medium are typically captured as scalable vector graphics. Rather than being a set of pixels, an image is described by a set of lines and curves. Typical applications include typefaces and black-and-white line art. In fact, the application is typefaces is so common that they are codified into scalable fonts which can then be inked onto a printed page.

The de facto industry standard for professional printing is PostScript. The foundation for this is a raster image processor (RIP), which converts a scalable vector description into a raster image that can be output by the target device. These target devices may include high resolution imagesetters, low resolution laser printers, and even software RIPs such as Display PostScript (embedded in the CDE Image View) or the open source Ghostview.

The World Wide Web Consortium (W3C) is now working on SVG (Scalable Vector Graphics) format, an XML representation of two-dimensional graphics. In anticipation of it becoming an approved standard, commercial vendors have begun incorporating SVG support into their products.

### 3.3.1 File Formats

#### 3.3.1.1 PostScript

PostScript is an interpretive language designed for electronic printing. A PostScript file is often a program described as ASCII text. A PostScript file always starts with

```
%!PS-
```

It may also conform to PostScript Document Structuring Conventions (DSC), which use PostScript comments to relay other information to a software tool; such information might include: creation date, fonts used, number of pages, etc.

The interpreting engine is the PostScript RIP. In fact, to handle Asian character sets, the RIP should be able to handle PostScript Level 2 or higher.

#### 3.3.1.2 Portable Document Format

While PostScript is a full interpretive programming language, PDF (Portable Document Format) preserves the imaging model of PostScript, but has a briefer syntax and less programmatic capability. Its target is to allow on-line rendering of formatted documents. The most popular tool for PDF rendering is Adobe Acrobat.

### 3.3.1.3 Encapsulated PostScript

Illustrations are often created in a special form of PostScript, known as Encapsulated PostScript (EPS). EPS is designed for portability so that an illustration can be imported anywhere onto a page, resized, rotated, or manipulated in some other way without having to modify the contents of the EPS file itself. In this way, an EPS file is like a self-contained software function which is dropped into a larger program.

Not only is EPS used for artwork embedded into a book. It may also be used for book cover art, and multi-lingual instruction cards and licenses.

Although illustrations in EPS files can be resized in the target document, they should be created as if no resizing will be done.

## 3.3.2 Scalable Fonts

### 3.3.2.1 Asian Fonts

Unlike U.S. or western European locales, files with Asian characters cannot assume that the requested fonts are on the target device. Depending on the deliverable, it may be necessary to embed the font into the file, or create an outlined font (with fonts converted into a set of PostScript operators).

For Sun technical documentation, Asian CID fonts are typically embedded into PDF files. These PDF files can be read by Acrobat 4, and PostScript Level 2 code generated for printing.

### 3.3.2.2 Sun Font Family

Some deliverables, such as multi-lingual instruction cards, require the use of the *Sun font family*. This is a proprietary font family used exclusively by Sun for some of its material. The fonts are available to vendors to need to produce such work for Sun and who sign a non-disclosure agreement with Sun covering the use of the fonts. The fonts are only usable on the Apple Macintosh platform.

## 3.3.3 Guidelines

### 3.3.3.1 Callouts

Illustrations requiring callouts (explanatory labels) are generally created as EPS files. Text should be created at the correct size. In general, this is the size of running text in the enclosing target document.

### 3.3.3.2 Outlined Fonts

When returning EPS artwork to Sun which includes Asian fonts, both the original artwork and artwork with outlined fonts should be returned. This allows any last minute modification, and allows restarted from the translated EPS, should it be necessary.

(In Adobe Illustrator, outlined fonts can be created by the Type => Create Outlines function.)

### 3.3.3.3 Stray Points

EPS artwork should be checked for stray points. These are single points which do not show up in the visible artwork, but are registered in the EPS file. Stray points often occur when the user instructs the drawing application to start a text string at a particular point in the illustration, but never actually puts text there. Similarly, an invisible rectangle for bounding purposes may be placed in an illustration, but never really used.

Many software tools cannot distinguish stray points from points in actual use. When these points occur outside the normal boundaries of an illustration, these tools assume that the illustration is larger than it actually is, and that it needs to be automatically reduced in size to fit onto a page. This reduction often results in unused white space surrounding the illustration, and typefaces being reduced from what they should be.



# CDE Help (HelpTag Format)

---

---

## 4.1 Introduction

HelpTag is a markup language used for authoring CDE help volumes, and is based on SGML. This section provides translators with specific guidelines to translate the Help components of a Sun product.

The overall guidelines ruling any technical translations are that it is correct, clear, concise and consistent. It is important that any platform specific requirements and standards are adhered to. It is also vital that the functionality of Sun software is not jeopardized and that the look and feel of the user interface is retained.

Sun requires that all of their vendors make quality the highest priority. In order to achieve this it is important that good communication is established between Sun and their vendors, that any issues are dealt with promptly and that measures are put in place to ensure a smooth localization process.

Vendors should check that the material received is adequate to complete the job. If any files or reference material seem to be missing, these should be queried with Sun. All translators involved in a project must have all relevant information.

Specific project guidelines as well as glossaries must be adhered to. If any queries arise they should be addressed before the completion of the project.

---

## 4.2 Translating CDE Help

### 4.2.1 Overview

The on-line help used by Sun is written using HelpTag, a variation of SGML (Standard General Markup Language). The Tag source is compiled to generate a help volume. The help volume is then used by the CDE (Common Desktop Environment) help viewer to display help on-line.

Help volumes can include illustrations and each illustration is a separate graphic file. When the help volume is compiled, a reference to the graphic file is put into it.

### 4.2.2 What should be translated?

In general, all text inside a tag, i.e. within '<' and '>' should not be translated. All other text should be translated.

There are two exceptions to this rule, the `idx` tag and the `term` tag. The `idx` tag is equivalent to K footnotes in Win-helps. They look like this:

```
<idx|text to translate|
```

You may also find index with a primary and secondary level. They look like this:

```
<idx|primary text:secondary text|
```

In these tags the primary and secondary text should be translated. Do not remove any of the '|' or ':' characters.

The `term` tag looks like this:

```
<term nogloss|text to translate|
```

This tag provides a reference to another glossary item. The translation must be identical in both places.

Here is an example of a helptag source file:

```
<!-- Help volume: Image Viewer -->
<!-- File name: Messages -->

<chapter id=Messages>Image Viewer Messages

This section describes possible causes and solutions for
Image Viewer error messages.

    <idx> error messages <\idx>
    <idx> messages, error <\idx>

<sl id=ImageVMemoryErrs>Image Viewer Memory Error
<p>You may see the following message if your system doesn't have
```

enough local memory to run Image Viewer.

```
<list bullet>
<item>Unable to allocate sufficient memory. Image Viewer cannot
continue.
<location id=ImageVCantAllocErr> <\location>
<\list>
```

```
<procedure>Possible Solutions:
<list bullet>
<item>Close some applications to free up some local memory.
<item>Increase swap space.
<p>If you don't know how to increase swap space, contact your
system administrator.
<item>Add more local memory to your system.
<\list>
```

```
<s1 id=ImageVDNDErrs>Image Viewer Drag and Drop Errors
<p>You may see one or more of the following messages if you
encounter a Drag and Drop error.
```

When translated, it appears as follows:

```
<!-- Help volume: Image Viewer -->
<!-- File name: Messages -->
```

```
<chapter id=Messages>이미지 열람기 메시지
```

이 절에서는 이미지 열람기 오류 메시지의 발생 원인과 해결 방법에 대해 설명합니다.

```
<idx> 오류 메시지 <\idx>
<idx> 메시지, 오류 <\idx>
```

```
<s1 id=ImageVMemoryErrs>이미지 열람기 메모리 오류
<p>시스템의 로컬 메모리가 이미지 열람기를 실행하기에 부족하면 다음과 같은 메시지가 나타납니다.
```

```
<list bullet>
<item>충분한 메모리 할당이 불가능합니다. 이미지 열람기를 계속할 수 없습니다.
<location id=ImageVCantAllocErr> <\location>
<\list>
```

```
<procedure>해결 방법:
<list bullet>
<item>얼마간의 로컬 메모리를 해제하려면 일부 응용프로그램을 닫으십시오.
<item>스왑 영역을 증가시키십시오.
<p>스왑 영역을 증가시키는 방법을 모르면 시스템 관리자에게 문의하십시오.
<item>시스템에 보다 많은 로컬 메모리를 추가하십시오.
<\list>
```

```
<s1 id=ImageVDNDErrs>이미지 열람기 끌어 놓기 오류
<p>끌어 놓기 오류가 발생했을 때 하나 이상의 다음 메시지가 나타납니다.
```

### 4.2.3 Formatting tags

In the source files you may find formatting tags. These are `!!` and `%%`. Any text between such tags will be formatted as bold or italic. They look like this:

```
!!Bold!! other text %%italic%%
```

Make sure that these tags enclose the correct tagged word, even if the word order in the translated text changes. These tags are not to be confused with the placeholders mentioned in the Messages chapter.

### 4.2.4 Filenames and extensions

See rules as outlined in the Software section on 절2.1 “Introduction” (27페이지)

---

## 4.3 Components of Sun Help Files

### 4.3.1 Topic titles

When a user performs an index search, the results are shown as matching topic titles and the number of "hits" per topic. Each title occupies a single line. As a result, the may need to scroll the results horizontally to see the whole title. When possible, keep the title brief. As a general rule, the more important words should be closer the beginning of the title. This reduces the need to scroll the results to read the full title.

English	Korean

### 4.3.2 First and second level headings

English	Korean

### 4.3.3 Step titles

English	Korean



#### 4.3.4 Numbered and bulleted lists

Grammar rule for your language: punctuation and first word capitalization

English	Korean

#### 4.3.5 Related topics

English	Korean

#### 4.3.6 Glossary

If a Glossary is included as part of the files, please ensure that all entries should be re-sorted alphabetically after translation. Also ensure that any formatting applied to the English term is applied to the translated term, moving to a different word if necessary.

#### 4.3.7 Formatting of software references

Rules for Documentation apply here.



# HTML Files

---

---

## 5.1 Introduction

HTML (*HyperText Markup Language*) is a non-proprietary format that is modeled after SGML and is a type of hypertext that can be used for the World Wide Web. It is the presentation component of HTTP and there are ways to format it from scratch, or also in a WYSIWYG form. Do not change any HTML tags, especially the links.

---

## 5.2 Localized HTML Files

HTML files are often used as README files on a CD because they are the lowest common denominator in readable file formats. As such, they are often intended to be accessible before any software is installed and therefore exist outside of Solaris installable packages.

### 5.2.1 HTML Header Section

In an HTML document, all header information is wrapped in a pair of tags `<HEAD` and `</HEAD>`. It may contain information about the title, charset, meta information, cascading style sheet, script, etc.

To localize the header section, the following needs to happen:

- Translate the content between `<TITLE>` and `</TITLE>`.
- Change or add the charset in `META` tag.

- If there is a cascading style sheet in it, make sure all fonts are localized correctly.  
For example, my make be necessary to change

font-family: helvetica

to

font-family: *local-font-name*

Here is an example of HTML header section,

```
<HEAD>
  <META HTTP-EQUIV="Content-Type"
CONTENT="text/html; CHARSET=iso-8859-1">
  <TITLE>SOLARIS DOCUMENTATION CD IMPORTANT INFORMATION</TITLE>

  <META NAME="OWNER" CONTENT="Sun Microsystems">

  <SCRIPT language="JAVASCRIPT" src=showTime.js>
</SCRIPT>

  <STYLE TYPE="text/css">
    BODY { font-size: 12pt; font-family: helvetica;
background-color: #FFFFFF; color: #000000; }
    .localheader { font-family: helvetica; font-size: 11pt;
color: #FFFFFF; font-weight: bold; }
    .locallink { font-family: helvetica; font-size: 11pt;
color: #FFFFFF; font-weight: normal; }
    .categorylink { font-family: helvetica; font-size: 11pt;
color: #FFFFFF; font-weight: bold; }
  </STYLE>
</HEAD>
```

after translated, it should be like this:

```
<HEAD>
  <META HTTP-EQUIV="Content-Type"
CONTENT="text/html; CHARSET=euc-kr">
  <TITLE>SOLARIS DOCUMENTATION CD IMPORTANT INFORMATION</TITLE>

  <META NAME="OWNER" CONTENT="Sun Microsystems">

  <SCRIPT language="JAVASCRIPT" src=showTime.js>
</SCRIPT>

  <STYLE TYPE="text/css">
    BODY { font-size: 12pt; font-family: gothic;
background-color: #FFFFFF; color: #000000; }
    .localheader { font-family: gothic; font-size: 11pt;
color: #FFFFFF; font-weight: bold; }
    .locallink { font-family: gothic; font-size: 11pt;
color: #FFFFFF; font-weight: normal; }
    .categorylink { font-family: gothic; font-size: 11pt;
color: #FFFFFF; font-weight: bold; }
  </STYLE>
</HEAD>
```

## 5.2.2 META Tags

By using META tags, the site will bind the varname to the HTTP header field so that an HTTP server will use it to process the document. The META HTTP-EQUIV method should be used only if the charset parameter cannot be set on the server. For the different languages, this may be done so in such a manner:

For Western European languages:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

For Korean:

```
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
```

## 5.2.3 Charset Encodings

Charset encodings are parameters used to specify an HTML document's character encodings. The server sends a message to indicate that an HTML document uses a particular charset encoding (such as ISO-8859-1). The message would look like this:

```
Content-Type: text/html; charset=ISO-8859-1
```

Language	Charset Encoding
Korean	euc-kr
Simplified Chinese	gb2312
Traditional Chinese	x-euc-tw (or big5 for some messages)

## 5.2.4 HTML Body

The BODY contains all the content of an HTML document. In general, all content between <BODY> and </BODY> should be translated, with the exception of the following:

- tags and their attributes
- anything between <script> and </script>

- anything between `<applet>` and `</applet>`
- anything between `<object>` and `</object>`

## SolBook SGML

---

---

### 6.1 Introduction

This section provides translators with specific guidelines to translate the documentation components of a Sun product. The components are SGML files adhering to Sun's SolBook DTD.

Specific project guidelines as well as glossaries must be adhered to. If any queries arise they should be addressed before the completion of the project.

---

### 6.2 Components of Sun Documentation

Sun manuals are usually structured as follows: Cover and Title pages, Table of Contents, Chapters (with sections and subsections), Appendices, Glossary and Index. These parts are related to each other and are of utmost importance to keep consistency amongst all components.

This section presents an overview of Sun's conventions and is to be adhered to when localizing the documentation.

#### 6.2.1 Title page and cover page

The Title page contains the manual title (including relevant trademark symbols when appropriate), a Sun logo, Sun address, part number, and release date.

## 6.2.2Part numbers

Each Sun document has an unique part number. A part number consists of nine digits, for example 805-1234-01. The first seven digits identify the manual and the last two digits identify the manual's revision level.

Sun will provide translators with the new part numbers; please replace the English part number with the correct one for your language.

Multilingual documents have the same part number as the original English.

## 6.2.3Release dates

English	Korean
Revision A, February 1998	1998년 2월 개정 A

The release date should match the one appearing on the English version of the document. However, as this sometimes changes vendors should query this point.

## 6.2.4Phone and fax numbers

Phone and Fax numbers should be localized.

English	Korean
650 960-1300	(650) 960-1300

## 6.2.5Copyright page

The copyright page contains the various copyright and trademark statements required in Sun manuals. The copyright page is a left page, printed on the back of the title page. Sun has standard Copyright text that will be supplied to vendors. Bear in mind that the copyright layout is different for each language, as follows:



Manual	Korean
English	English and French
French	French
German	German and French
Spanish	Spanish and French
Italian	Italian and French
Swedish	Swedish and French

## 6.2.6 Table of contents

The Table of Contents lists the first, second, and third level headings of a manual. The Table of Contents is generated automatically in Adept and FrameMaker. Translators should generate and check the Table of Contents so that it reflects the manual structure.

## 6.2.7 Extended characters

In HTML and SGML files the best way to enter Extended Characters, e.g. ä and ö is to use entities and edit the file in a text format, as opposed to in Adept. Therefore to enter ä you would type &auml; and to enter ö you would type &ouml;.

Please refer to the English files supplied for other examples.

## 6.2.8 Chapter titles

English	Korean
Solaris Advanced Installation Guide	Solaris 8 고급 설치 설명서
Solaris 2.6	Solaris 8

## 6.2.9 Headings

English	Korean
Overview of Installing Solaris	Solaris 설치 개요
Performing an Interactive Installation	대화식 설치

## 6.2.10 Step titles

English	Korean
How to Perform an Interactive Installation	대화식 설치 수행 방법
Ways to Upgrade a System	시스템 업그레이드 방법
What to do Before Upgrading	업그레이드 전에 해야 할 일

## 6.2.11 Numbered and bulleted lists

Grammar and punctuation rules:

English	Korean

## 6.2.12 Formatting

주: [NOTE: FIX PAGE #]

See the Section "Typographic Conventions" on page 55, for any formatting rules.

# 6.3 Localizing Sun Documentation

## 6.3.1 Overview

Sun uses SGML (Standard General Markup Language) for most documentation. From the SGML source printed documentation, on-line documentation (AnswerBook) and web documentation is generated. SGML files are similar to HTML, in the tags appearance. In fact, HTML is a subset of SGML.

The structure of SGML is defined by a DTD (Document Type Definition). The DTD is a file that defines what tags are used and how.

These files do not contain any information about the appearance of the text. Instead SGML uses a FOSI (Formatting Output Specification Instance) that defines the way the output should look like. The FOSI can be viewed as a type of style sheet. There is a separate FOSI for each language. It contains translations for prefixes such as

Chapter, Index etc. It also defines sort orders for Indexes but cannot be edited or modified by the translators. If there are problems or errors in the FOSI, they should be reported to Sun immediately.

All books have a number of SGML files with the extension `.sgm` and a book file with the extension `.book`.

A book may contain illustrations. Each illustration is a separate graphic file that is referenced into the book. The graphic files are created by taking screenshots (and editing these shots in some cases) in several different applications depending on which format is used.

### 6.3.2 Directory/File structure

The file structure for most projects are as follows:

```
Shortname/  
  figures/  
  Meta/  
  Bookname.book  
  Bookname.ps  
  .sgm-files
```

#### 6.3.2.1 shortname.book

The `shortname.book` is the file that defines what should be included in the book. There is some translation to be done in this file.

The `bookname.ps` is included for reference. This is a postscript file of the English book generated by Sun and should be printed and used as a comparison to complete your translated book. It can be useful for the translators to see how the text should be formatted and to see how screenshots are taken and their context.

#### 6.3.2.2 Figures Directory

The `Figures` directory contains all the graphics for the book and they are categorized into three major groups:

1. **UCO's - (Use Client Original), does not require any work, eg. icon, button.**
2. **Leverage - Screenshot has not changed from previous release. Pull in graphic from previous release.**
3. **New - Screenshot or graphic that is completely new.**

This directory name should not be changed and the contents should not be placed in any other directory. Please ensure this is so before delivering translated files back to Sun.

### 6.3.2.3 Meta Directory

The `Meta` directory contains several files and must not be removed. The `Pagination Sheet` for the book is stored in this directory and it contains how many pages are in the book and how they are numbered. In the `Master` page are text columns that act as a simple form for entering book titles and descriptions. The `META` directory also contains a `Readme` file (from the English writer) outlining changes if this book is an update from a previous version.

Most importantly in this directory there is another directory called `LastIds`. This contains information that is vital for the ID's in `Adept`. This directory does not require translation or editing but should be kept with the book at all times. Do not delete this directory.

### 6.3.3 Translating the .book file

On the next page is an example of a book file, which shows the end part of the file. The section not shown contains a long list of entity declarations. The entities that need to be localized should be edited in `Adept` only. In the example given the entity `&BookName` is something that should be translated in `Adept`.

There are a number of elements that need to be localized. They are preceded by the following tags:

- `<pubsnumber>` part number for the locale you are translating to
- `<pubdate>` name of the month
- `<publisher>` translate if needed.

Please refer to instructions provided by Sun in relation to project specific issues.

```
<book fpi="-//Sun::SunSoft//DOCUMENT CDETRANS Version 2.0//en
label="beta" id=CDETRANS" lang="en" userlevel="user"><title>&BookName;</title>
<bookinfo><bookbiblio><title>&BookName;</title>
<authorgroup><author><firstname>John</firstname><surname>Smith
</surname></author></authorgroup>
<isbn></isbn>
<pubsnumber>805-3903&ndash;10</pubsnumber>
<pubdate>September 1998</pubdate><publisher><publishername>Sun
Microsystems, Inc.<publishername><address><street>901 San Antonio
Road</street><city>Palo Alto</city><state>CA</state><postcode>
94303</postcode><country>U.S.A.</country> </address></publisher>
<copyright><year>1998</year><holder>Sun Microsystems</holder></copyright>
<abstract>&abstract;
</abstract>
</bookbiblio>
<legalnotice>&legal; &fr-legal; </legalnotice>
```

```

<subjectset>
<subject>
<subjectterm>Desktop & Window Systems</subjectterm>
<subjectterm>Introduction & Overview</subjectterm>
</subject>
</subjectset>
</bookinfo>&Preface;&OWtoCDE;
<index>
<indexentry><primaryie></primaryie></indexentry>
</index></book>

```

---

⚠: The sections inside the tags <legalnotice>, <subjectset> and <subjectterm> should NOT be translated.

For more details please refer to any project specific details given to you by your Sun Project Manager.

---

### 6.3.4 Remove MS-DOS end-of-line characters

MS-DOS uses CR-LF as end-of-line character while UNIX uses LF only. The eol characters need to be changed. This is done with the script `eolfix`:

```
eolfix *.sgm
```

Please ensure that all MS-DOS end-of-line characters are removed before delivering translated files to Sun.

### 6.3.5 Formatting and layout

Ensure that the book conforms to the Sun standards and that standard templates, FOSI and formats, are used.

### 6.3.6 Typographic Conventions

In Sun documentation, the use of Typographic Conventions is important and very common. Translators should ensure that the formatting applied to the English carries forward into the translation, moving the format to a different word or position in a sentence, if necessary. The following table is an example of the typographic conventions used in most Sun documentation:

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

### 6.3.7 Verify SGML code

There are two methods to verify SGML code.

#### 1. Using "Check Completeness" in Adept.

This is very straight-forward. Simply open the `Book` file only, go to Tools -> then Check Completeness. This will verify that your overall book is valid SGML. A dialog box will appear on the screen when it is complete. If there are any errors they will be displayed under the headings "Object in error" and "Nature of error". You can double click on the underlined error and it will bring you to the line that is causing the problem.

#### 2. Using a Sun tool called `doclint.sgml`.

This tool is much faster than using Adept to verify the code and must be run on all deliveries to Sun. At the command line, making sure that all files are closed in Adept, type the following: `doclint.sgml bookname.book`

A log file is created automatically, containing the results generated from `doclint`. It gives you a pass or fail rating. This log file will contain two types of problems:

- error messages - these will cause a fail rating,
- warnings messages - the warnings will allow a pass rating, unless there are error messages, in which case it will fail.

All errors must be fixed before delivery to Sun. The warnings are not so important, however, they contain information regarding the file sizes and table lengths which may be of some use.

---

⚠: Any errors relating to Olinks, Archive, and Cross-References may be ignored, as these can only be fixed by Sun.

---

## 6.3.8 Updating Entities

There are some entities that need to be translated, e.g. &BookName; etc. You must edit these entities in Adept only.

Go to Entities -&gt; Text to open the dialog box Text Entities. Here you should change the entities that contains text, such as BookName. Edit the text in the Context box and click on Change button for each entity.

Please refer to instructions provided by your Sun Project Manager for a project specific list of entities requiring translation.

## 6.3.9 Filenames and extensions

---

⚠: [NOTE: FIX PAGE #]

---

See rules as outlined in the Software section on page 32.

## 6.3.10 Graphics

Please ensure that all graphics are in the correct position and that all callouts are pointing to the correct items. Also ensure that figures and illustrations are numbered correctly.

The format of the graphics in the English source material should be followed at all times. Keep the exact same filenames, paying particular attention to the letter case used in the filenames. If the English has an .epsi graphic, then the localized version should be the same. Pay particular attention to the description of this file format in a terminal window. Open a terminal window, cd into the figures directory and type `file *`. The result you will get should resemble the following:

```
figures 1 % file *
desktop.epsi:      data
OWtoCDE.fig32.epsi: PostScript document
OWtoCDE.fig34.epsi: PostScript document
OWtoCDE.fig35.epsi: commands text
SMOnlineT.tiff:    TIFF file, big-endian
submnuHelp.tiff:   TIFF file, big-endian
```

The .epsi graphics that are not listed as "PostScript document" are in the incorrect format and will cause problems when the file goes to Answerbook creation. It will require resaving in the correct format by the vendor before delivery to Sun. Please see the graphic instructions for your project provided by Sun for more information.

### 6.3.11 Checklist

Please make sure that the following points have been verified:

- Ensure that the file names are the same as the originals.
- Ensure that no MS-DOS end-of-line characters are left.
- Ensure that no Translation Memory leftovers exist, if used.
- Check to be sure that you have an <abstract> element containing the entity Abstract.sgm.
- Open Abstract.sgm and verify that it is translated.
- Ensure that the tags <subjectset> and <subjectterm> are not translated.
- Ensure the filenames of graphics are exactly the same as the English.
- The size/resolution is similar to the original graphic (and it fits nicely on a page in the postscript).
- Print a new postscript file, ensuring that all art is in place and that no text runs off the edge of the page.

---

## 6.4 Cross References

Cross references are updated automatically and do not usually require translation. The use of Cross References changes depending on the software used.

### 6.4.1 Adept

In Adept there are 3 different types of cross references:

- 1. Internal cross references - to another section/chapter in that book**
- 2. Olinks/External cross references - to a section/chapter in another book**
- 3. Ulinks - links to web pages/URL's**



No translation is required since all other words, e.g. "on page", are already translated in the FOSI. In order to view the translated Internal cross references, do View -> Update Generated Text-once the book is translated.

## 6.4.2FrameMaker

In FrameMaker, there is just one type of cross reference. There may be some initial translation of the format required, e.g. the words "on page". Once translated in one chapter, this format can be imported into the other chapters.

It is the translator's responsibility to verify that all cross references are correctly updated.

## 6.4.3References to other applications, manuals, and operating systems

If you find references to other products or software options from other products applications. Translate them as appropriate if the correct glossary is available or submit your queries to Sun. If the exact translation cannot be found, these references should remain in English. The same rule holds for third party trademarks.

---

## 6.5Glossary

---

⌘: [NOTE:FIX PAGE #]

---

See the rules outlined in the section"Glossary" on page 32.

---

## 6.6Index

An index is often the reader's primary information retrieval device. When readers search for a particular topic and find it referenced in an index, they are assured that the topic is covered and need look no further.

The following rules apply when translating an index:

1. **Translating an index may require that you create or delete entries, combine or split entries, and regroup or reword entries.**

The index is subject to the same typographical style conventions found in the text or the document itself. For example, if file names and commands appear in Courier font in your book, then they should appear that way in the index.

English	Korean
dump command, 44	dump 명령 44

## 2. Indentation:

The following table shows two examples of the indented indexing style used at Sun. Example A shows the index format for unnumbered chapters; Example B shows numbered chapters. The only differences between the two examples are the format of the page numbers and the use of "to" as a separator for page ranges in the book with numbered chapters. This setting is pre-defined in the English files, and should not be changed at translation stage.

English	Korean
Example A Unnumbered chapter format	Example A Unnumbered chapter format
A application architecture, 12 application gateway, 211, 345-351 automounter facility, 49-54 <i>See also</i> mounting overview, 49 setup, 51 specifying subdirectories, 51 B backing up file systems, 58 dump command, 89-92	A 응용프로그램 절 구조, 12 응용프로그램 게이트웨이, 211, 345-351 automounter facility, 49-54 <i>See also</i> mounting overview, 49 setup, 51 specifying subdirectories, 51 B backing up file systems, 58 dump command, 89-92

English	Korean
Example B Numbered chapter format	Example B Numbered chapter format
A application architecture, 1-2 application gateway, 4-18, 5-76 to 5-77 automounter facility, 2-12 to 2-18 <i>See also</i> mounting overview, 2-12 setup, 51 specifying subdirectories, 2-14	A 응용 프로그램 절 구조, 1-2 응용 프로그램 게이트웨이, 4-18, 5-76 to 5-77 automounter facility, 2-12 to 2-18 <i>See also</i> mounting overview, 2-12 setup, 51 specifying subdirectories, 2-14
B backing up file systems, 3-1 dump command, 3-34	B backing up file systems, 3-1 dump command, 3-34

### 3. Capitalization:

Do not capitalize any word in an index entry unless the word is a proper noun, an acronym, or an abbreviation that is supposed to be capitalized. Use the standard rules for capitalization.

### 4. Nesting levels:

The Sun style uses up to three levels of nested entries: primary entry, secondary entry and tertiary entry. Each entry level is indented from the previous level. See examples in previous table.

### 5. Double posting:

Double posting means identifying a topic in two different places in an index. For example, a topic that appears as "address switch" and "switch, address": is double-posted in the index. A topic that appears in three places is triple-posted, and so on.

Special consideration must be taken when translating these entries. Since each language has a different word order, the translator might find that some entries need to be "rearranged" so that they make sense in the translated version of the index. Sometimes the resulting translation might result in a duplication. The translator should make sure that any entry duplication is avoided. This might imply deleting one or more of the entries (if they are not applicable to your language) or rearranging them.

### 6. Automatic generation of index:

In Adept Indices are only generated when you either create a postscript file or when Print Preview is run. The Index will not appear as a separate file and cannot be edited manually. All index entries are part of the main body of the document, i.e. in the Chapters, and they can only be translated and edited there. Before starting translation of the index entries, you must check that the English index generates correctly. If this is not the case, report it immediately to Sun's Project Manager.

**7. In Framemaker the Index is generated from the book file.**

If hypertext links are created, they can be used to jump to an Index Marker if editing is required.

---

주: Sort Order, Headings such as "Numerics" and words such as "to" in page numbers are pre-defined and translated in each language specific FOSI. These cannot be changed by translation. If there are any errors, please report them to Sun immediately, so that a bug can be raised, and the issue fixed before the final delivery.

If Extended Characters do not appear correctly in the Adept generated Index, please close all files and Adept, change the locale to your language, open the files, and create the postscript file/run Print Preview again to regenerate your Index.

It is the translator's responsibility to generate the index and to check it for consistency and duplications.

---

## Standard Terminology

---

This appendix describes standard terminology which is applicable across books and projects.

---

### A.1 Book Organization

Rev 2000.2.29

This section gives translations for the common parts and headings of a book.

English	Korean
Contents	목차
Preface	머리말
Who Should Use This Book	이 책의 사용자
How This Book Is Organized	이 책의 구성 방법
Related Books	관련 서적
Related Documentation	관련 문서
Ordering Sun Documents	Sun 문서 주문 방법
Accessing Sun Documentation Online	Sun 문서 온라인 액세스

English	Korean
Typographic Conventions	활자체 규약
Shell Prompts in Command Examples	명령에서의 쉘 프롬프트 예

## Sun Trademarks

---

---

### B.1 Book Organization

Rev 2000.2.29

This section gives a representative list of Sun trademarks and servicemarks.

- HotJava™
- HotJava™ Views™
- Java™
- JavaBeans™
- JavaChip™
- Java™ Compiler Compiler™
- Java Cup™ International
- Java<sup>SM</sup> Developer Conference<sup>SM</sup>
- Java<sup>SM</sup> Developer Connection™
- JavaEngine™
- Java Financial Object Xchange™
- JavaJoint™
- JavaOne<sup>SM</sup>
- JavaOS™
- Java™ Portability Kit™
- Java™ Reel™
- JavaScript™

- Java™ Select Program™
- JavaServer™
- JavaSoft™
- JavaSpaces™
- JavaStar™
- JavaStation™
- Java Store<sup>SM</sup>
- JavaStudio™
- Java™ Telecom Object NEtwork<sup>SM</sup>
- JavaTutor™
- JavaWorld™
- Java™ WorkShop™
- Java™ Financial Object Xchange<sup>SM</sup>
- microJava™
- picoJAVA™
- Solaris™ Supplement for JavaStation™
- UltraJAVA™
- Visual Java™
- 100% Pure Java™

For further information please refer to your Sun Project Manager for project specific issues.



## Localized Book Titles

This appendix lists Solaris books which are commonly referred to by users and by other documentation.

The list covers the following Collections in solaris 8:

- Solaris 8 Installation Collection
- Solaris 8 User Collection

### C.1 Solaris 8 Installation Collection

These are the translations of titles found in the Solaris 8 Installation Collection:

English	Korean
What's New in the Solaris 8 Operating Environment	Solaris 8 운영 환경의 새로운 기능
Solaris 8 (Intel Platform Edition) Installation Guide	Solaris 8 (Intel 플랫폼판) 설치 설명서
Solaris 8 (SPARC Platform Edition) Installation Guide	Solaris 8 (SPARC 플랫폼판) 설치 설명서
Solaris 8 Advanced Installation Guide	This book is not localized.

English	Korean
Solaris 8 (Intel Platform Edition) Device Configuration Guide	Solaris 8 (Intel 플랫폼판) 장치 구성 설명서
Solaris 8 (Intel Platform Edition) Hardware Compatibility List	Solaris 8 (Intel 플랫폼판) 하드웨어 호환 목록

## C.2Solaris 8 User Collection

These are the translations of titles found in the Solaris 8 User Collection:

English	Korean
OpenWindows User's Guide	This book is not localized.
OpenWindows Advanced User's Guide	This book is not localized.
Solaris Common Desktop Environment Advanced User's and System Administrator's Guide	Solaris 공통 데스크탑 환경: 고급 사용자 및 시스템 관리자 안내서
Solaris Common Desktop Environment: User's Guide	Solaris 공통 데스크탑 환경: 사용 설명서
Solaris Common Desktop Environment: User's Transition Guide	Solaris 공통 데스크탑 환경: 사용자 변환 설명서
Using Power Management	전원 관리 사용법
Korean User's Guide	한국어 Solaris 사용 설명서
Korean System Administrator's Guide	한국어 Solaris 시스템 관리자 설명서
Korean Release Overview	한국어 Solaris 릴리즈 소개

## 용 어 해 설

---

**Common desktop  
environment (CDE)**

The Common Desktop Environment (CDE) was developed by several companies so that GUIs would be compatible with other platforms, such as UNIX Systems. With it, users could work on several computing platforms at once and this encouraged positive transfer, because one environment would enhance the performance over another. The CDE Desktop is under the Solaris user environment, a Motif interface, and powered by X11.

**Portable Operating  
System Interface  
(POSIX)**

POSIX is an IEEE standard (IEEE 1003.) for operating system interfaces that are based on UNIX operating systems. Programs that use the POSIX standard can be easily ported from one system to another. There are different parts of the POSIX standard. For example, POSIX.1 is a standard for the program interface in the C language whereas POSIX.2 is for standard shell and utilities interfaces.

**X/Open Portability  
Guide, Issue 4  
(XPG4)**

It is a specification that contains internationalized headers, interfaces, and utilities. X/Open (now known as the Open Group) is a group of computer manufacturers who promote UNIX-based portable applications. The XPG specifies locale mechanisms, documented in a book that published by the X/Open group called the X/Open Portability Guide. XPG3 contains all of POSIX and XPG4 is the newest release, which contains internationalization features.



# 색인

---

## A

acronyms, 19  
Adept, 7, 72

## B

.bat files, 42  
book organization, 7, 77  
brackets, 20

## C

C program messages, 30  
capitalization, 19  
CDE, 83  
character sets, 43  
character Sets  
참조 charset,  
charset encodings, 61  
checklist, 44  
components of Sun documentation  
cross references, 7, 72  
country specific guidelines, 22  
date, 22  
localization examples, 23  
measurements, 22  
separators, 23  
standard phrases, 24

## D

date, 22

## E

encoding  
META tags, 61  
encodings  
charset encodings, 61

## F

filenames and extensions, 44, 56  
formatting tags, 56

## G

GIF, 47  
glossary, 24  
grammatical issues, 17  
2nd person singular versus impersonal  
form, 17  
infinitive versus personal, 17  
modal verbs versus imperative, 17  
passive voice versus active voice, 18  
translation of the -ing form, 18

## H

help files, 56  
first and second level headings, 56  
formatting software references, 57  
glossary, 57  
numbered and bulleted lists, 57  
related topics, 57  
step titles, 56  
topic titles, 56

## HTML

- localized HTML files, 59
- HTML files, 5, 59
- HTML header section, 59
- hyphenation, 21

## J

- Java files, 37
  - Java language keywords, 37
- Java language keywords, 37

## L

- lists
  - numbered and bulleted, 57
- localizing Sun documentation, 6, 66
  - checklist, 6, 72
  - directory/file structure, 6, 67
    - bookname.book, 67
    - figures directory, 67
    - Meta directory, 68
  - filenames and extensions, 6, 71
  - formatting and layout, 6, 69
  - graphics, 6, 71
  - remove MS-DOS characters, 6, 69
  - translating .book file, 6, 68
  - typographic conventions, 6, 69
  - updating entities, 6, 71
  - verify SGML code, 6, 70

## M

- measurements, 22
- message translation guidelines, 28
- messages, 27
- META tags, 61
- .msg files, 36

## P

- .po files, 35
- punctuation, 20
  - brackets, 20
  - Comma, period, colon, semicolon, 20
  - hyphenation, 21

## R

- related books,
- resizing, 33
  - left justifying, 35
  - minimum field width, 35
  - plus or blank, 35
  - precision, 35

## S

- separators, 23
- shell prompts,
- Solbook SGML, 6,
- standard phrases, 24
- step titles, 56
- structure, 28
- stylistic guidelines, 15
  - colloquialisms, 15
  - hardware and software personification, 15
  - interrogatives, exclamations, and rhetorical questions, 15
- Sun documents
  - ordering,

## T

- tense
  - future versus present, 17
- terminology, 24
  - glossary, 24
  - using the glossary, 25
- TIFF, 46
- time, 22
- time, country specific guidelines
  - time, 22
- .tmsg files, 42
- topic titles, 56
- Trademarks, 25
- typographic conventions, , 69

## U

- use of numerals, 20