LINGUISTIC STYLE GUIDELINES

Writing for Sun

Sun Microsystems Ireland Ltd.,
European Localisation Centre,
Hamilton House,
East Point Business Park,
Dublin 3, Ireland.

October 1998, Version 1

Contents

Chapter 3 Help 29

Chapter 4 Documentation 33

Chapter 1 The Basics

Introduction

In this guide Sun tries to give their translation vendors standard linguistic guidelines, which highlight the conventions and linguistic issues related to the translation and localization of software, help and documentation of Sun products. Sun invites feedback from their vendors. Suggestions and criticisms will be incorporated in future updates of this guide.

Most 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
- write clearly and concisely in 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 your language (i.e. consistency of terminology, correct application of grammatical, syntactical and punctuation rules, avoiding of anglicisms, colloquialisms, and jargon).

Why Style is Important

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.

Write for your reader

As a technical writer and translator, you communicate information on behalf of the reader—who depends on you. In this sense your main tasks is to enable the reader to use the product in question. It is part of the translation process to adapt the source text to the requirements of the target audience.

Work with an Editor

Technical translation does not occur in a vacuum. Other translators can provide invaluable help, and you should always work with an editor on your translation. An editor relates to a document as an advocate for the reader, and as a professional who can critique your work. The editor is often the “first customer” to read your document.

The following are some general rules which can help translators during the translation process to achieve a high-quality technical translation.

Stylistic Guidelines

Hardware and software personification

Rule for your language:

Example:

English	**Language**
The program will delete the document.	

Interrogatives, exclamations and rethorical questions

Rule for your language:

Example:

English	Language
How do you use an applet?	
Let’s get to know the program!	

Colloquialisms

English technical manuals often use colloquialisms which must be avoided in the translation.

Example:

English	Language
By doing this, you can see the Welcome to JavaViews screen.	
Now you know what to do if you want to proceed with the installation.	

Grammatical Guidelines

Infinitive versus personal style

Grammar rule for your language:

Example:

English	**Language**
Use the script to save the existing configuration to a diskette.	

Future tense versus present tense

Grammar rule for your language:

Example:

English	**Language**
This chapter will describe the procedures to install a client/server system.	

Modal verbs versus imperative

Grammar rule for your language:

Example:

English	**Language**
You should save data before proceeding to the next step.	

2nd person singular versus impersonal form

Grammar rule for your language:

Example:

English	Language
Make sure you specify a valid name for the device, and that you have network access rights.	

Passive voice versus active voice

Grammar / Stylistic rule for your language:

Example:

English	Language
The adapter card is used as follows:	

Translation of the -ing form

Grammar rule for language (depending on context: 1st level heading, 2nd level heading, etc.):

Example:

English	Language
Adding and saving documents to folders	

Capitalization

Grammar rule for your language:

Example:

English	Language
Building a Simple Applet	

Character sets

The correct character set for your language must be used. The spelling checker you use must recognize the extended characters.

Acronyms

Use the corresponding acronym appropriate to your language.

Example:

English	Language
URL (Uniform Resource Locator)	

Use of numerals

Rule for your language:

Example:

English	Language
5 directories and 12 files	

Abbreviations

Generally avoid abbreviations where possible and only use standard ones. Where abbreviations are necessary due to space reasons, follow your country grammar rules.

Examples:

MB
KB
b
pt
dpi
ppi
lpi
n/a

Punctuation

Comma, period, colon, semicolon

Rule for your language:

Commas, periods, colons and semicolons follow immediately the character preceding them and they do not require a space before.

For more detailed information about country specific rules related to the use of punctuation, refer to your language guidelines.

Brackets

There should be no space between brackets and the text inside them. In English the bracket only contains the period if it is a full sentence.

Hyphenation

Translators should avoid inserting manual hyphens. If hyphenation is necessary, adhere to the rules for your own language. Do not split product names and trademarks (except where necessary due to space restrictions).

In Documentation translators should NOT insert hyphenation, as Sun uses Adept and the page displayed is not WYSIWYG. Therefore, you cannot see where a line break will be after postscripting or Answerbook creation.

Country Specific Guidelines

Date

The correct format for dates is:

Example:

English	**Language**
24 June 1998	

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

English	**Language**
06/24/98	

Time

Time format for your country:

Example:

English	**Language**
2:00 pm	

Measurements

English measurements are replaced with the metric ones, with a few exceptions (3.5” disk and display measurements). If in doubt, ask Sun for instructions.

Example:

English	Language
The monitor weighs 74 lb	

Units of measurement are usually not followed by a period. *Example*: cm, MB, MHz, kg. Please check if this rule applies to your language.

Separators

Rule for your language:

Example:

English	Language
1.5 mm	
230,000,000,000	

Keyboard

Please refer to the Sun standard keyboard for your language.

Localization of examples

For all the examples which are culturally bound, such as names of persons, places, events and so on, localize them according to the culture of your country. Avoid references to real persons, places or events.

Example:

English	**Language**
John Smith 23, Malborough Ave San Francisco USA	

Standard phrases

The following phrases are translated as follows in all Sun documents:

English	**Language**
Note Important Warning Caution See also Part number Revision	

Terminology

Glossary

During the translation process, consistency in terminology within a product and with other Sun products is required. This will be maintained by the user of a Glossary. Translators will be provided with a Sun master glossary. 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. This glossary is intended for use by anyone who performs localization of Sun's software or hardware products.

Localization must be done in accordance with the standard Sun 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, where 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 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.

Format of Glossary

The Sun glossary is an Applix spreadsheet tab-delimited file, which can be easily exported and imported into/out of Excel 4. Excel 4 is the highest compatible version between Applix and Excel.

Hierarchic - which product line to use

Sun has various products which are part of a product line. Within Sun glossaries they are categorised under the heading of Java, Solaris and Hardware.

Localization Process

1. Translate adhering to Sun's glossary.
2. For 3rd party products, refer to product or reference material. Request from Sun if necessary).
3. New terms should be translated to vendors best ability, and sent to Sun for validation. Please refer to Sun's *Glossary Maintenance Procedure* for further information.
4. Approval of the new terms will be decided by Sun internal linguistic testers. Please refer to Sun's *Glossary Maintenance Procedure* for instructions on how to proceed after approval or rejection of a new term.
5. Please attempt to send queries concerning terminology issues during the life cycle of the project.

Trademarks

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

Some names may be used as trade names of the company and also as trademarks for the Sun line of product. 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:

Example:

English	**Language**
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...	

The two symbols adopted by Sun are ™ for trademarks and SM for service marks. 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™ 2.7 would appear in text mode as:

```
<trademark>Solaris</trademark> 2.7
```

To protect Sun trademarks designate them on books spines, covers, self-covers and the first time they are mentioned in the text. After the first time you mention it, you can use the trademark as an adjective without the symbol.

Here are some examples of Sun trademarked products and services:

- HotJava™
- HotJava™ Views™
- Java™
- JavaBeans™
- JavaChip™
- Java™ Compiler Compiler™
- Java Cup™ International
- Java℠ Developer Conference℠
- Java℠ Developer Connection℠
- JavaEngine™
- Java Financial Object Xchange™
- JavaJoint™
- JavaOne℠
- JavaOS™
- Java™ Portability Kit™
- Java™ Reel™
- JavaScript™
- Java™ Select Program™
- JavaServer™
- JavaSoft™
- JavaSpaces™
- JavaStar™
- JavaStation™
- Java Store℠
- JavaStudio™
- Java™ Telecom Object NEtwork℠
- JavaTutor™
- JavaWorld™
- Java™ WorkShop™

- Java™ Financial Object Xchange^{SM}
- 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.

Don’t translate....

The following is a list of terms not to be translated. This list will be updated in future releases, and in relation to specific projects. If you are in doubt as to whether to translate a term / phrase, please do not hesitate to contact your Project Manager at Sun.

- Sun Microsystems Computer Company
- Computer Systems
- A Sun Microsystems, Inc. Business
- Solaris 2.6 Hardware: 5/98
- Sun’s address on title pages.
- USA
- Trademark registered words
-
-
-
-
-

Chapter 2 Software

Introduction

This section provides translators with specific guidelines to translate the different software 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 require all of their vendors to 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.

Translating Sun Software Files

Overview

On a UNIX system such as Sun Solaris, the GUI based software uses the X-Window system, plus some library that provides widgets (window gadgets) such as buttons, menus, sliders, etc. The most common library that is used is called Motif (the CDE Windowing Environment is mainly based on Motif).

While at at the lowest level, all GUI applications that run on Solaris use the X-Window system, they can use libraries other than Motif on top of this to give a different look and feel. Examples would be the XView library, or the Swing library - part of the Java Foundation Classes. Some applications, such as Netra, which is HTML based, rely on the underlying web-browser to provide widgets. Finally, some Sun software is designed not to use a GUI at all.

These different ways of presenting software to the end user mean that there are several different file formats that contain localisable text which is pulled into the application running in the locale in question.

Here is a list of the different file formats Sun has used:

- .msg files
- .po files
- .tmsg files
- .java files
- .properties files
- .rc files
- .html files (sometimes with embedded java applets)
- app-defaults files
- plain text files (application specific format)

The exact format of each file type along with information on how to compile them is available from your Project Manager at Sun. The following are some points on dealing with software files that apply to all file formats.

Structure

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 localisable message text itself. Please ensure that key values do not get translated or changed in any way - otherwise the software will not be able to locate the correct message key, and the text will appear in the application in english, or may cause unexpected behaviour in the application.

Resizing

In most cases, this happens automatically, and there is no need to specify the size of buttons or dialogs.

DOS vs. UNIX

All plain text (non-compiled) files in DOS format should be processed using 'dos2unix -ascii' -- which strips `<carriage return>` characters from the end of lines; since DOS uses the character sequence `<carriage return><line feed>`, whereas UNIX just uses `<line feed>` to end lines.

Placeholders & Formatting Information

In nearly every file type, there is the notion of placeholders. These are symbols that will get substituted for text when the application is running. For example:

```
"Could not open file : {0} - this file does not exist"
```

This would appear in the application as:

```
"Could not open file : test.txt - this file does not exist"
```

These placeholders can be moved according to the syntax of your language. Sometimes placeholders can represent the format that the substituted text is to be printed in. For example in one file format, the placeholder %X makes the substituted text be printed in hexadecimal format, and the string "\n" prints a new line. So, the following might appear within a localisable text file:

```
"The hexadecimal \n number is 0x%X \n."
```

This would get printed as :

```
The hexadecimal
number is 0x30a
.
```

Character Sets

Here is a short list of the common character sets that Sun uses for it's localised files. Ensure that you use the correct character set when returning source files to Sun.

German	iso-8859-1 (Latin 1)
Spanish	iso-8859-1 (Latin 1)
French	iso-8859-1 (Latin 1)
Italian	iso-8859-1 (Latin 1)
Swedish	iso-8859-1 (Latin 1)
Portugese	iso-8859-1 (Latin 1)
Catalan	iso-8859-1 (Latin 1)
Polish	iso-8859-2
Russian	iso-8859-5
Hungarian	iso-8859-2
Czech	iso-8859-2

Since the 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 localised.

.msg and .tmsg files

Msgs files, message text source files, are text files containing the text that will appear in dialog boxes, menus, etc. For extended characters, use the Latin1 character set.

Note: The fields are separated by a single space or tab character. Any other spaces or tabs are considered as part of the subsequent field. Therefore, you must make sure that you don’t add or remove any leading spaces or tabs.

Filenames and extensions

Files must be returned with exactly the same filenames in exactly the same directory structure. If in doubt, please contact your Project Manager at Sun.

Conversion sequences

In the message text you may find conversion characters. They look like this:

```
Start character'%'
```

These sequences will be replaced with a string, digits, or a character when the program runs. In the following example:

```
10 Could not open: %s.\n
```

will be shown as:

```
"Could not open test.txt"
```

if this message is shown while trying to open the file "test.txt".

These sequences should not be changed, but may be moved to fit the translation.

Compiling

Before any file is compiled, you must remove MS-DOS end-of-line characters. Use the `eolfix` script to solve this:

```
eoifix *.msg
eolfix *.tmsg
eolfix *.po
```

Translating strings with placeholders

Format characters in the C language are always preceded by the `%` symbol. This symbol represents a placeholder for a variable type, it does not represent a variable. In C code output is usually generated by a `printf` command. The `printf` command works as follows:

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

when executed, the above lines produce the following output:

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

To get variables to print out we need to add both the variable and a place holder for that variable. Suppose that we introduced a variable called itm, which is a string. The variable itm is set to equal CCCCCCC. To print it out we could use the line as follows:

```
printf(" the item is %s .\n.\n.",itm)
```

Which when executed gives us the following output:

```
the item is CCCCCCC.
.
.
```

You should notice that the actual variable itm is placed outside the double quotes. To display two items itm and itm2, where itm2 is DDD, use the following:

```
printf(" the item is %s and the second is %s .\n.\n.",itm,itm2)
```

This produces the following output:

```
the item is CCCCCCC and the second is DDD.
```

In all the above examples we were using variables of type string (they represent a String of characters). C can handle many more variable types which are listed below and give their corresponding placeholders.

Variable Type	**PlaceHolder**
integer	%d
character	%c
string	%s
hexadecimal	%x %X
octal	%o
unsigned	%u
long	%ld
float	%f
exponential float	%e
%e or %f	%g

Variable Type	**PlaceHolder**
\...	escaped chars, see note at end
%h	short
%%	Displays % symbol

Now consider the following example where `var1` is an integer set at 25, and `var2` is a string set at Free.

```
printf(" there is %d %% %s",var1,var2)
```

This will display the following output:

```
there is 25 % Free
```

PO file format

PO files are obtained by running a utility on C source code which strips out anything between quotes in printf statements and placing this text in a portable object file (po). The C source code is then compiled in binary form. As the PO files have no sense of 'typing' it just assumes that text is text. This means that you could switch around `%s` and `%d` inside messages and compile into an MO file without any errors. When the binary is run and it calls that particular message Solaris checks the type of each variable and assigns the correct values to the variables, no matter which order you have put them in. This only works if each variable in a string is of a different type.

If the number 25 was held as a string then the above message would look as follows:

```
msgid " there is %s %% %s"
msgstr " there is %s %% %s"
```

Swapping the `%s` 's around will not change the display of the string, so if you need to re-arrange the order of the strings in your translation you can do as follows:

```
msgid " there is %s %% %s"
msgstr " %2$s space is approx %1$s %%"
```

displays this message:

```
Free space is approx 25 %
```

The `%2$s` tells Solaris to display the second string variable in this position.

Formatting variables

Place holders can be used for formatting the display of variables in the output. See the following examples of formatting variables and then the summary of these at the end of this section. This works on all variable types.

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 if the integer contains 18 digits then the full number is still displayed, as 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"
```

Which 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 width is 15 and so is the maximum width.

Left justifying

The defaults so far all use right justify, 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 command:

```
%+d
```

Summary of format commands

The format of the printf command is summarised below:

```
%[pre][num][.num]vartype
```

Where:

pre indicates a preceding `+' if set to +

num indicates minimum field width of the variable

.num Indicates the number of decimal places for variables that allow decimal places, or the maximum field width for all other variables.

vartype represents the type of variable being displayed as outlined above.

Escaped characters

As mentioned earlier, printf statements can contain escaped characters. In earlier examples we used \n to denote a new line. In the first example of this section you saw the following three lines:

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

Which produced the following 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`. (\ denotes that the next character is escaped). `\n` means to print a new line character, which places the cursor in the first column of the next line.

Listed below are the escaped characters and their meaning:

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

Checklist

Please make sure that the following points have been verified:

- ✓ Date / Number formats are correctly localised.
- ✓ All placeholders have been moved to preserve syntax.
- ✓ All files have been proof read and spell-checked.
- ✓ All files can be compiled without errors. (this is a good way to check file format compliance)
- ✓ The directory structure is correct.
- ✓ The files have been verified in the running software.
- ✓ All files have been processed with 'dos2unix -ascii'.
- ✓ No key values have been changed.

Components of Sun Software

When localizing Sun's software translators must pay particular attention that code or anything that affects the functionality of the software remains unaltered. Specific project instructions provided by Sun will identify these components (i.e. file formats, directory structure, parameters, protocols etc.).

The software is the core part of the product and therefore special effort must be made to give the end user the feeling that the product has been developed for their target market. All other components (such as Help and Documentation) are based on the software in relation to terminology. It is therefore of the utmost importance that consistent, accurate and concise translations are used throughout.

The following guidelines deal with the different components in more detail.

Menu titles

See the following menus from the File Manager Help:

English	**Language**
File Edit Search Navigate	

Hot keys

Hot keys must be present in every menu title and its corresponding options. If these hot keys are missing in the English, please add them in your translation, but make sure you inform your Project Manager at Sun about this. Avoid using extended characters as hot keys. As part of the process of compiling/testing the software you should make sure that hot keys are not duplicated.

The use of hot keys is platform dependent. Ensure that there is no duplication of hot keys and also that they are used consistently in all software components.

See the following menus from the File Manager Help:

English	Language
File Edit Search Navigate	

Accelerator keys

Accelerator keys are a combination of the Control key plus a letter or a symbol that allow the user quick access to a menu or menu option. If the combination is made up of Ctrl+letter, these accelerator keys should be localized following the Solaris Operating System Standards. If you have not received the file that allows you to modify these control keys, inform your Project Manager at Sun and you will receive instructions accordingly.

Accelerator keys in the “File” menu of Solaris File Manager are:

English	Language
Go Home (Ctrl+H) Go Up (Ctrl+U) Find (Ctrl+F)	

Menu options

Grammar rule for your language:

See the following examples from the File menu in the File Manager:

English	Language
New Folder... New File... Go Home Go Up Go To...	

Dialog box titles

Ensure consistency between Dialog box titles and menu options where applicable. Where the command name has been shortened, the corresponding dialog box title should appear in full.

Grammar rule for your language:

Example:

English	Language
2 examples? Menu is “Save As” Dialog box is “Save As”	

Dialog boxes

Dialog boxes are composed of the following:

- Buttons
- Check boxes
- Radio buttons
- Fields and Field names
- List boxes
- Tabs

Points to watch out for:

- None of the dialog box fields must be truncated.
- If a button or other item can be activated by pressing a hot key, the same hot key must be used in other dialog boxes for the same item.
- There must not be any hot key duplication in any dialog box.

Grammar rule for each language:

Status line text

This is the explanatory text that appears in the status line of a window whenever you place the cursor over an item in a window.

Grammar rules for your language:

Error messages and other messages

Example:

English	**Language**

Chapter 3 Help

Introduction

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 require all of their vendors to 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.

Translating Sun .SDL/SGML Help

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. Each illustration is a separate graphic file. When the help volume is compiled, a reference to the graphic file is put into the help volume.

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.

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.

Filenames and extensions

See rules as outlined in the Software section on page 19.

Components of Sun Help Files

Topic titles

Grammar rule for your language: punctuation, formatting of menu options or dialog box titles in Topic titles

Example:

English	**Language**

First and second level headings

Grammar rule for your language: punctuation, formatting of menu options or dialog box titles in Topic titles

Example:

English	**Language**

Step titles

Grammar rule for your language:

Example:

English	**Language**

Numbered and bulleted lists

Grammar rule for your language: punctuation and first word capitalization

Example:

English	Language

Related topics

Example:

English	Language

Glossary

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

Formatting of software references

Rules for Documentation apply here.

Chapter 4 Documentation

Introduction

This section provides translators with specific guidelines to translate the documentation 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 require all of their vendors to 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.

Localizing Sun Documentation

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.

Directory/File structure

The file structure for most projects are as follows:

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

bookname.book

The `bookname.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. This postscript file 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.

Figures Directory

The `Figures` directory contains all the graphics for the book. Sun will provide a detailed list of all graphics in use in each book. This list is broken down into three categories, as follows:

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.

Meta Directory

The `Meta` directory contains several files, and must not be removed. The Pagination Sheet for the book is stored in this directory. The 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.

Translating the .book file

On the next page is an example of a book file, showing 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.

```
<?Pub UDT _bookmark _target>
<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;<?Pub Caret></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 &amp; Window Systems
</subjectterm><subjectterm>Introduction &amp; Overview</
subjectterm></subject></subjectset>
</bookinfo>&Preface;&OWtoCDE;
<index>
<indexentry><primaryie></primaryie></indexentry>
</index></book->
<?Pub *0000008751 86>
```

Note – 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.

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.

Formatting and layout

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

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 `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`

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

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

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 -> 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 then using Adept to verify the code, and must be run on all deliveries to Sun. At the command line, (make 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, that may be of some use.

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

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 -> 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.

Filenames and extensions

See rules as outlined in the Software section on page 19.

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 localised version should be the same, paying 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 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.

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 exists, 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.

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 it is of utmost importance to keep consistency amongst all components.

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

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.

Part 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.

Release dates

Example:

English	**Language**
Revision A, February 1998	

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.

Phone and Fax numbers

Phone and Fax numbers should be localized.

Example:

English	**Language**
650 960-1300	+650 960-1300

Copyright 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 languages, as follows:

Manual	**Language**
English	English and French
French	French
German	German and French
Spanish	Spanish and French
Italian	Italian and French
Swedish	Swedish and French

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.

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.

Chapter titles

Example:

English	Language
Solaris Advanced Installation Guide Solaris 2.6	

Headings

Example:

English	Language
Overview of Installing Solaris Performing an Interactive Installation	

Step titles

Example:

English	Language
How to Perform an Interactive Installation Ways to Upgrade a System What to do Before Upgrading	

Numbered and bulleted lists

Grammar and punctuation rules:

Example:

English	Language

Formatting

See the Section “Typographic Conventions” on page 37, for any formatting rules.

Cross References

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

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 as 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.

FrameMaker

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 translators responsibility to verify that all cross references are correctly updated.

References 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.

Glossary

See the rules outlined in the section“Glossary” on page 32.

Index

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 in the document 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.

Example:

English	Language
`dump command, 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	Language
Example A Unnumbered chapter format **A** application architecture, 12 application gateway, **211**, 345-351 automounter facility, 49-54 *See also* mounting overview, 49 setup, 51 specifying subdirectories, 51 **B** backing up file systems, 58 `dump` command, 89-92	
Example B Unnumbered chapter format **A** application architecture, 1-2 application gateway, **4-18**,5-76 to 5-77 automounter facility, 2-12 to 2-18 *See also* mounting overview, 2-12 setup, 51 specifying subdirectories, 2-14 **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 as such cannot be edited manually at all. 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.

   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.

Note – 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.

Note – In an Adept generated Index if Extended Characters do not appear correctly in the 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.

Index