# Solution for Multilingual Publishing by Unicode and XSL

**January 2, 2004**
**Antenna House, Inc.**

## Problems in making multilingual literature

Let us first go over potential challenges in multilingual computer formatting. Each of these items is already difficult enough on its own, and rapid progress in technology is making our mastery and utilization of such formatting even harder.

In this document, we are going to compile the issue of multilingual computer formatting at first. Then the current state of the multilingual formatting are discussed in terms of Unicode, XML and XSL (Extensible Stylesheet Language). Finally, the examples of formatting are going to be listed.[1]

## How to create the source data for formatting?

Information needs to be prepared as coded data for computers to process it. From this perspective, the creation of multilingual data is far more difficult than doing so monolingually.

1. Selection of character encoding
    - Character encoding has been standardized basically for each country. But representation of data using a local character code set would not enable the handling of documents with a mix of multiple languages. Editing or formatting of a document which contains more than one language would inevitably require Unicode.
    - To what extent can Unicode support language diversity? What is the latest status of Unicode standardization? What products with Unicode capability are available? Since Unicode is evolving at a very fast pace, it is necessary to keep abreast with the latest news on Unicode and digest it accurately.
    - What kinds of problems does Unicode have?
    - The numbers of character codes usable have been limited in conventional ASCII, JIS, or ISO-8859 series encoding. In contrast, the Unicode Standard provides a variety of new character codes, for instance, the 16 character codes listed below. What significance do these codes have for formatting? How can we use them effectively?

**16 characters starting from U+2000**

```
2000;N # EN QUAD
2001;N # EM QUAD
2002;N # EN SPACE
2003;N # EM SPACE
2004;N # THREE-PER-EM SPACE
2005;N # FOUR-PER-EM SPACE
2006;N # SIX-PER-EM SPACE
2007;N # FIGURE SPACE
2008;N # PUNCTUATION SPACE
2009;N # THIN SPACE
200A;N # HAIR SPACE
200B;N # ZERO WIDTH SPACE
200C;N # ZERO WIDTH NON-JOINER
200D;N # ZERO WIDTH JOINER
200E;N # LEFT-TO-RIGHT MARK
200F;N # RIGHT-TO-LEFT MARK
```

2. Selection of the computer. How do we choose the hardware and OS?
    - What type of environment do we choose: Macintosh, Windows 2000/XP, UNIX such as Solaris, etc., Linux, or JAVA?

[1] This document is written as XML document conforming to SimpleDoc.dtd, which is the in-house standard document type definition, then formatted by XSL Formatter V2.5 and converted to PDF.

- In Windows, multilingual processing that includes Asian languages is made possible by the provision of a library called Uniscribe. It seems that Internet Explorer and Microsoft Word use Uniscribe to allow the processing of a wide range of Asian languages.
- Windows seems to be the most advanced in multilingual processing capability. How much processing capability can you obtain in JAVA for Asian languages? What is the current situation of the multilingual formatting by Linux or UNIX?

3. How do we enter data into the computer?
    - What types of software are available for data entry?
    - What kind of keyboard should be prepared? Keyboards have been standardized in each country; personal computers sold in a specific country come with keyboards in its national standard.
    - Do we need IME? What should be the selection criteria of IME? As is generally known, romaji (roman character) input and kana-kanji conversion is the main approach for input method of the Japanese language. But it seems too demanding for foreigners who are not familiar with Japanese to enter kanji by pronunciation using roman characters. By the same token, it will be very difficult for Japanese to input Chinese characters using pinyin, although it must be the natural choice for Chinese.
4. Method of representing data
    - Should the data be application dependent binary or should it be application independent XML?
    - XML could be the best for achieving multilingual processing. On the other hand, it is true that XML poses a higher hurdle for users to clear. Tagging as XML is not too difficult but, generally speaking, people tend to be overly intimidated by tags. How can we lower the hurdle for XML?
    - With XML, the data structure (Schema) has to be designed.
    - Instead of defining new data structures, can we use existing DTD/Schema?
    - Is it possible to propose a new standard DTD/ Schema definition? Will any new Schema appear?
5. Selection of editor software
    - As familiar editing software improves the productivity of document creation, determining the editing software is very important. From this perspective, Microsoft Word will be the first choice. Is Microsoft Word usable as multilingual editing software?
    - There is a number of editing software claiming to be multilingual. However, there is not much all-around software that is capable of editing English, other Western languages, Japanese, Chinese, Korean, Arabic, Hebrew, and Thai in a single version. If we have to switch editing software by language, no document with multiple languages can be generated. In addition, as changing software from language to language would involve learning of new operations and raise problems of data compatibility, it should be avoided.
    - In order to create data with XML, it is necessary to have tools which support Schema-driven data input and editing. Is there such multilingual software?
    - If experts create a document, they can use a type of XML editing software that displays the tag. Experts are knowledgeable enough to understand the meaning of XML tags. Is there any XML editing tool that shows XML tags while it edits a multilingual document? If so, which software is the best for that tool?

### Method of formatting

1. If we change layout of document frequently, we will need a WYSIWYG formatting software. It there any XML formatting software available that allows frequent layout changes, WYSIWYG editing, and reflection of editing results to the XML source data?
2. Fonts are essential in the visualization of character images on screen, to paper, or to PDF. What types of Unicode compatible fonts are available?
3. When a PDF file is created and then distributed or printed, it is necessary to embed the outline of fonts in PDF. Therefore, the fonts to be used in multilingual formatting have to allow outline-embedding. What fonts are available for multilingual formatting?

4. How much can we utilize XSL-FO (XSL)? To what extent can we specify complex layouts?
    - What is the characteristic of the XSL Formatter, which is the XML multilingual formatting software that complies with XSL specification?
    - Does it work when formatting rules are different from one language to another?
    - Does it work when languages with different formatting rules coexist in a single text?
    - Does it work when one language runs from right to left and another runs from left to right in one document?

### Printing and PDF creation methods

1. How to print multilingual documents
2. Distinctions between PDF for printing and PDF for the Web

### Others

1. Preparation of a Table of Contents and Back-of-the-Book Indexes.
2. Sorting order of indexes, sorting rules by language, and sorting rules for mixed language documents

## Preliminary knowledge of multilingual formatting

### Character and language

A language is written with one or more scripts, digits, signs and marks. A coded character set defines the aggregate of letters, characters, digits, signs and marks. There are many local coded character sets for each country and language. The following table shows a list of character sets for major languages.

| ISO Language code | ISO Language | Type of letter | code classified by area |
|---|---|---|---|
| ar | Arabic | Arabic | ASMO 449, Latin/Arabic Alphabet |
| bg | Bulgarian | Cyrillic | Latin/Cyrillic Alphabet |
| km | Cambodian | Khmer | (First registered from Unicode V3.0) |
| zh-CN | Chinese (Simplified) | Simplified Chinese | GB2312, GB18030 |
| zh-TW | Chinese (Traditional) | Traditinal Chinese | BIG5 |
| hr | Croatian | Latin | Latin Alphabet No.2, 10 |
| cs | Czech | Latin | Latin Alphabet No.2 |
| da | Danish | Latin | Latin Alphabet No.1, 4, 5, 6, 8, 9 |
| nl | Dutch | Latin | Latin Alphabet No.1, 5, 9 |
| en | English | Latin | Latin Alphabet No.1..10 |
| et | Estonian | Latin | Latin Alphabet No.4, 6, 7, 9 |
| fi | Finnish | Latin | Latin Alphabet No.4, 6, 7, 9, 10 |
| fr | French | Latin | Latin Alphabet No.9, 10 |
| de | German | Latin | Latin Alphabet No.1..10 (Excluding 7) |
| el | Greek | Greek | Latin/Greek Alphabet |
| he | Hebrew | Hebrew | Latin/Hebrew Alphabet |
| hi | Hindi | Devanagari | IS 13194 (ISCII), etc. |
| hu | Hungarian | Latin | Latin Alphabet No.2, 10 |
| is | Icelandic | Latin | Latin Alphabet No.1, 6, 9 |
| id | Indonesian | Latin | Latin Characters |
| it | Italian | Latin | Latin Alphabet No.1, 3, 5, 8, 9, 10 |
| ja | Japanese | Latin, Lanji, Kana, Katakana | JISX0201, JIS X0208, JIS X0212 |
| kk | Kazakh | Cyrillic | Extended Latin/Cyrillic Alphabet (Cyrillic Asean) |

| ISO Language code | ISO Language | Type of letter | code classified by area |
|---|---|---|---|
| ko | Korean | Hangeul, Kanji | KS C5601, KS X1001, Johab |
| lv | Latvian | Latin | Latin Alphabet No.4, 7 |
| ms | Malay | Latin orArabic | Latin Alphabet, Arabic Extended |
| lt | Lithuanian | Latin | Latin Alphabet No.4, 6, 7 |
| no | Norwegian | Latin | Latin Alphabet No.1, 4..9 |
| fa | Persian (Farsi) | Arabic | Extended Latin/Arabic Alphabet (Arabic Character 28+ Original 4 Characters) |
| pl | Polish | Latin | Latin Alphabet No.2, 7, 10 |
| pt | Portuguese | Latin | Latin Alphabet No.1, 3, 5, 8, 9 |
| ro | Romanian | Latin | Latin Alphabet No.10 |
| ru | Russian | Cyrillic | koi8-r, Latin/Cyrillic Alphabet 32 Chars (not compatible with Ukrainian) |
| sr | Serbian | Cyrillic | Latin/Cyrillic Alphabet (Serbian) |
| sk | Slovak | Latin | Latin Alphabet No.2 |
| sl | Slovenian | Latin | Latin Alphabet No.2, 4, 6, 10 |
| es | Spanish | Latin | Latin Alphabet No.1, 5, 8, 9 |
| sv | Swedish | Latin | Latin Alphabet No.1, 4, 5, 6, 8, 9 |
| sw | Swahili | Latin | |
| tl | Tagalog/Takalog | Latin | |
| th | Thai | Thai | TIS 620, Latin/Thai Alphabet |
| tr | Turkish | Latin | Latin Alphabet No.5 |
| uk | Ukrainian | Cyrillic | koi8-u, Latin/Cyrillic Alphabet 33 Chars |
| ur | Urdu | Arabic Extended | |
| vi | Vietnamese | Latin | Extended Latin Characters |
| xh | Xhosa | Latin | |
| zu | Zulu | Latin | |

## Unicode

At present, the Unicode Standard provides the coded character set of scripts, digits, signs, and marks for almost any languages around the world.

### History of Unicode

```
Oct. 1991 Unicode 1.0.0 issued
Jul. 1996 Unicode 2.0.0 issued
Sep. 1999 Unicode 3.0.0 issued
Mar. 2002 Unicode 3.2.0 issued
Apr. 2003 Unicode 4.0.0 issued
```

Unicode not only defines coded character set, but also provides other specifications as follows:

- "The Unicode Character Database" which indicates writing direction of each character and other information on characters
- "The Line Breaking Properties". The standard describes the property of each character that allows or prevents a break opportunity before or after the character.
- "The Bidirectional Algorithm" which rules algorithm for determining the writing direction of ambiguous characters between text strings with different writing direction. These problems are encountered when a document contains both characters that are described from left to right (such as Latin alphabets or Japanese characters), and from right to left (such as Arabic or Hebrew alphabets).

These specifications have become a foundation for the development of software to process multilingual documents.

### Internal character code of OS and application

During the eighties to the nineties, the personal computer OS was based on national standard of character codes. The application programs that run on the OS were restricted by the OS and had limitations on handling of character codes. For example, Japanese Windows Me internally manipulates Japanese characters that are encoded by Shift-JIS (JIS X0201 plus JIS X0208). Application software that runs on Windows Me cannot easily process special Latin letters such as A with diaeresis : Ä, O with diaeresis : Ö, U with diaeresis : Ü and so on. These codes are assigned for half-width katakana in JIS X0201, and the codes conflict with special Latin letters.

As for Microsoft Windows 2000/XP, the processing inside OS is based on Unicode and multilingual processing functions are strengthened sufficiently. Windows 2000/XP should be selected for multilingual processing.

Some application software manipulates internal data encoded by Unicode, and the other manipulates internal data encoded by local standard. To process multilingual documents, it is necessary to select the application software which processes Unicode inside. For example, XSL Formatter and Microsoft Word 2000/XP are Unicode application, but FrameMaker is not a Unicode application.

### Role of application

Multilingual processing is not complete even if application software is able to process Unicode. There are some problems between Unicode and multilingual processing. The followings are examples:

**Glyph substitution**

When we write Japanese or Traditional Chinese text, both vertical and horizontal writing can be used for the same string of text. For some kind of character codes such as punctuation marks, parentheses, and quotations, it is necessary to use different glyphs in vertical or horizontal writing. Formatting engine should change glyphs automatically.

日本語、『縦書』。

日本語、『縦書』。

The Arabic script is cursive. Each letter has four glyphs and changes glyph depending on the letter appears by itself or at starting, intermediate, or ending position in a word. Software for Arabic also should change glyphs automatically.

**Syllable composition**

Southern East Asian languages, such as Thai, Cambodian, and Laotian, arrange syllables. A syllable consists of a consonant letter, vowel signs, and tone marks. Unicode defines character code points for each consonant letter, vowel sign, and tone mark. Consequently, application should be able to form a syllable with a consonant, vowel marks, and tone marks from a sequence of character codes.

### Font

When processing languages through computers, font technology is the next important infrastructure. In fact, without fonts, characters can be neither printed nor displayed. The following table contains a list of fonts that are usually supplied with Microsoft Windows 2000/XP, or can be downloaded free of charge from the Internet. Among these fonts, Arial Unicode MS is the only font that covers all range of Unicode.

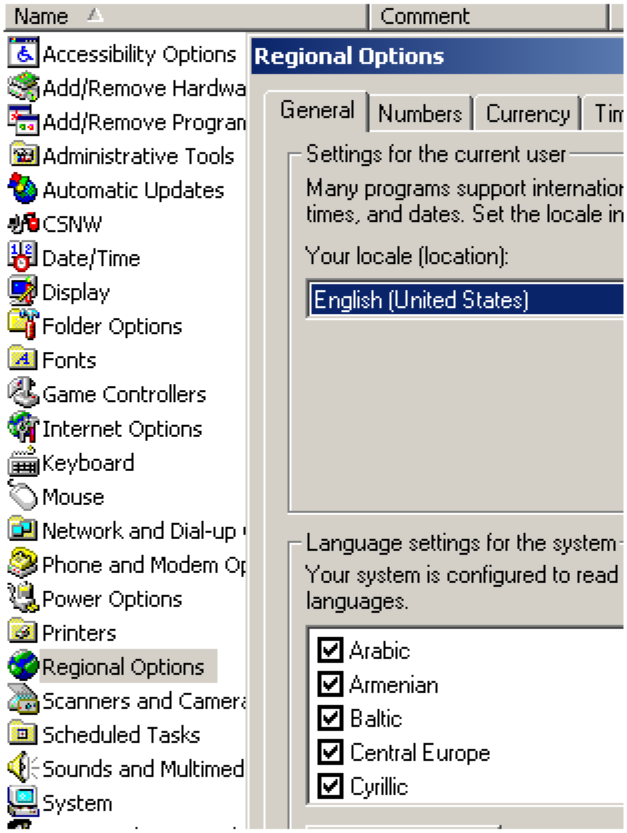
Arial Unicode MS has drawbacks that it does not include all of the characters of Unicode 4.0 yet, and its design of glyph is somewhat poor in quality.

For languages such as English, Western European, Slav, Japanese, Chinese (simplified and traditional), Korean, Arabic, Hebrew, and Thai, TrueType or OpenType (TrueType Format) fonts with enough quality can be prepared free of charge. Of course, these fonts alone are insufficient for designers who illustrate high quality print materials. However, for the purpose of IOM manuals, these fonts are practical.

Standard setting procedure of Windows 2000 does not necessarily install all fonts that are supplied with Windows 2000. Angsana (Thai font) or Mangal (Hindi font) is not installed with the standard installation of Windows 2000/XP. These languages are not installed unless you select the Regional Options of the Control Panel, go to the system language setting, choose the language (e.g., Thai, or Indic), and reset the system. (See next diagram)

| Font family | The principal character which it covers | Procurement manner | Sort |
|---|---|---|---|
| Arial Unicode MS | All characters of Unicode V2 | Office2000/XP etc. | Sans-serif |
| Arial | Latin, Greek, Cyrillic, Arabic, Hebrew | 2000/XP | Sans-serif |
| Courier New | Latin, Greek, Cyrillic, Arabic, Hebrew | 2000/XP | Monospace |
| Lucida Console | Latin, Greek, Cyrillic | 2000/XP | Monospace |
| Lucida Sans Unicode | Latin, Greek, Cyrillic, Hebrew, symbol | 2000/XP | Sans-serif |
| Microsoft Sans Serif | Latin, Greek, Cyrillic, Arabic, Hebrew, Thai | 2000/XP | Sans-serif |
| Tahoma | Latin, Greek, Cyrillic, Arabic, Hebrew, Thai | 2000/XP | Sans-serif |
| Times New Roman | Latin, Greek, Cyrillic | 2000/XP | Serif |
| Vernada | Latin, Greek, Cyrillic | 2000/XP | Sans-serif |
| Arabic Transparent | Arabic | 2000/XP | Sans-serif (Latin), Cursive (Arabic) |
| Traditional Arabic | Arabic | 2000/XP | Sans-serif (Latin), Cursive (Arabic) |
| Sylfaen | Latin, Greek, Cyrillic, Armenian, Georgian | XP | Serif |
| MS Hei | Simplified Chinese | IE5, Global IME5 | Monospace (Latin), Sans-serif (Chinese) |
| MS Song | Simplified Chinese | IE5, Global IME5 | Monospace (Latin), Serif (Chinese) |
| SimSun | Simplified Chinese | XP | Monospace (Latin), Serif (Chinese) |
| MingLiU | Traditional Chinese | 2000/XP | Monospace (Latin), Serif (Chinese) |
| PMingLiU | Traditional Chinese | Office2000 | Serif |
| Mangal | Devanagari | 2000/XP | |
| Palatino Linotype | Greek Poliytonic | 2000/XP | Serif |
| Shruti | Gujarati | XP | |
| Raavi | Gurmukhi | XP | |
| David | Hebrew | 2000/XP | Serif |
| David Transparent | Hebrew | 2000/XP | Serif |
| Fixed Miriam Transparent | Hebrew | 2000/XP | Monospace |
| Miriam | Hebrew | 2000/XP | Sans-serif |
| Miriam Fixed | Hebrew | 2000/XP | Monospace |
| Miriam Transparent | Hebrew | 2000/XP | Sans-serif |
| Rod | Hebrew | 2000/XP | Monospace |
| MS Gothic | Japanese | 2000/XP | Monospace (Latin), Sans-serif (Japanese) |
| MS Mincho | Japanese | 2000/XP | Monospace (Latin), Serif (Japanese) |
| Tunga | Kannada | XP | |
| Batang | Korean | 2000/XP | Serif |
| Gulim Che | Korean | IE5, Global IME5 | Monospace (Latin), Sans-serif (Korean) |
| Estrangelo Edessa | Syriac | XP | |
| Latha | Tamil | 2000/XP | |
| Gautami | Telugu | XP | |
| MV Boli | Thaana | XP | |

| Font family | The principal character which it covers | Procurement manner | Sort |
|---|---|---|---|
| Angsana New | Thai | 2000/XP | Serif |
| Cordina New | Thai | 2000/XP | Sans-serif |
| IrisUPC | Thai | 2000/XP | Sans-serif |



Many programs support internatio
times, and dates. Set the locale in

Your locale (location): English (United States)

Your system is configured to read
languages.

☑ Arabic
☑ Armenian
☑ Baltic
☑ Central Europe
☑ Cyrillic

**Setting of regional options**

### PDF technology

PDF technology is another promotional feature of a multilingual formatting. PDF is a medium that emulates paper digitally. Paper can not be transmitted as fast as its digital version via the Internet to anywhere across the world. The multilingual PDF could be circulated by electronic media such as CD-ROM or by Internet.

An important aspect is that the embedding of font outline data into PDF becomes possible.

If PDFs contain Arabic, Hebrew, or Thai scripts and they are created without embedded outline of font, they may not be circulated across the globe. The embedding of font outline in PDF is substantial for multilingual document.

### XML

XML is the most suitable technology to create multi-language documents.

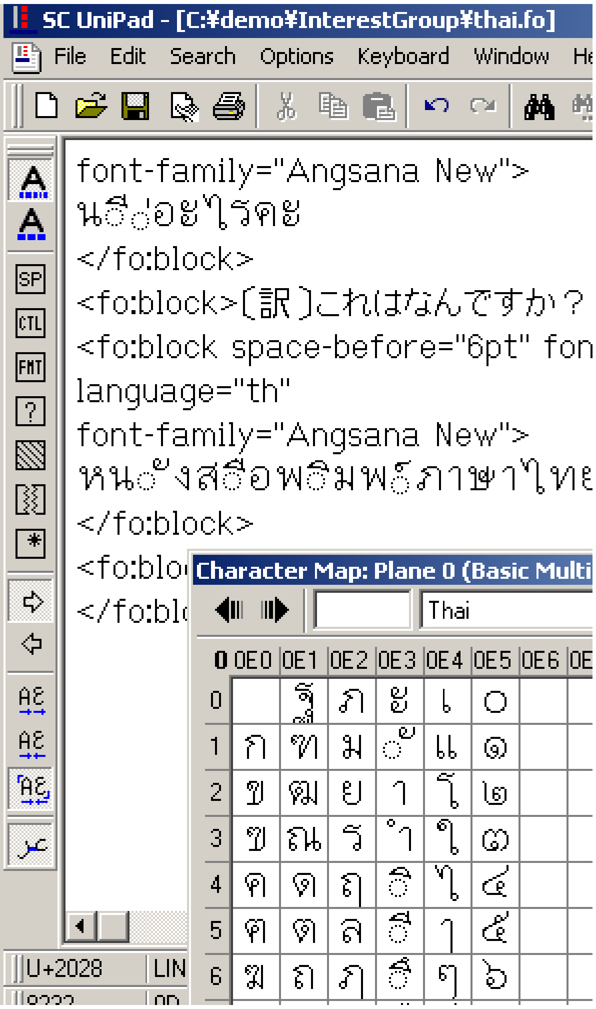
- XML adopts UTF-8 and UTF-16 of Unicode encoding as its default character encoding. XML data encoded as UTF-8 or UTF-16 are expected to be processed without any character code conversion by major XML tools. The local encoding of each country may also be specified with XML documents. In that case, XSL Formatter converts character encoding to UTF-16 when it reads XML document. The document adopting local encoding may not be converted correctly depending on tools.
- If word processors such as Microsoft Word are used, it is easy to type, edit, or print small amount of document that contain multilingual script. However, when we create enormous amount of documents, transform documents into different formats, or print documents with professional level-quality, it is necessary to interchange data between related applications. The foregoing data interchangeability is achieved by writing the information in XML.
- In XML, a document file can be divided into many partial files. Graphics are independent from main documents and may be linked with the main document as external files. Using this mechanism, when creating a document, one can store the body text portion of different languages into separate files. Graphics in all languages can be used as common files. Finally, all these parts may be integrated together to form a complete document.

### Creating and editing multilingual XML documents

There are three ways to create multilingual XML contents.

1. To use a text editor that is able to edit multilingual scripts
2. To use XML editor that can handle multilingual scripts
3. To use a word processor that is able to process multilingual scripts

Since XML is a text file, a text editor can be used to edit XML. There are text editors that accept multilingual scripts such as NotePad for Windows and UniPad. Especially UniPad is useful since it can display and edit each code point for such script as Thai by using code map of the Unicode 4.0.



```
font-family="Angsana New">
นี่อะไรคะ
</fo:block>
<fo:block>[訳]これはなんですか？
<fo:block space-before="6pt" fon
language="th"
font-family="Angsana New">
หนังสือพิมพ์ภาษาไทย
</fo:block>
<fo:blo
</fo:bl
```

Character Map: Plane 0 (Basic Multi

Thai

| 0 | 0E0 | 0E1 | 0E2 | 0E3 | 0E4 | 0E5 | 0E6 | 0E |
|---|---|---|---|---|---|---|---|---|
| 0 |  | ฐ | ภ | ะ | เ | ๐ |  |  |
| 1 | ก | ฑ | ม | ◌ั | แ | ๑ |  |  |
| 2 | ข | ฒ | ย | า | โ | ๒ |  |  |
| 3 | ฃ | ณ | ร | ำ | ใ | ๓ |  |  |
| 4 | ค | ด | ฤ | ◌ิ | ไ | ๔ |  |  |
| 5 | ฅ | ต | ล | ◌ี | ๅ | ๕ |  |  |
| 6 | ฆ | ถ | ฦ | ◌ึ | ๆ | ๖ |  |  |

**Input of character by UniPad**

Although some of new XML editors appeal us that they may edit multilingual scripts, there seems to be no advanced one. As stated above, an XML editor may not support multilingual completely if it only processes Unicode.

Microsoft Word is the word processor that enables us to edit the largest number of languages in one version. In order to create XML version of the document created by Microsoft Word, we had to save the document in RTF to use some kind of tools which transforms RTF to XML. In Microsoft Word 2003, however, it becomes possible to edit XML documents that are written with user defined XML Schema. It becomes also possible to save the documents without user Schema as WordprocessorML format. Since WordprocessorML is a kind of XML format, we can transform the document from WordprocessorML to any other XML format more easily than from RTF to XML format. WordprocessorML format will gradually replace RTF. We, Antenna House, introduced the world's first style sheet that transforms WordprocessorML into XSL-FO.

OpenOffice1.1/StarOffice7 that were released in Fall 2003 makes ability of editing multiple language enhanced and editing of Arabic and Thai possible. Since OpenOffice1.1/StarOffice7 save document as XML format, they can be considered as one of the options to create XML contents in multiple languages.

## Multilingual computer formatting with XSL

### What is XSL?

XSL is the specification that is designed in order to format and print XML onto the media which has the concept of paper. XSL is designed by taking the following multilingual computer formatting into account.

XSL defines a set of objects for formatting, such as page, header area, footer area, side bar, footnote area, footnote contents, before float, side float, block level, character level, inline level, a list or an itemized statement, table, or link.

By specifying the properties (attribute values) for each object, the layout or style of each object can be designated.

XSL Formatter is the multilingual formatting engine that enables us to format XML document in accordance with the layout that is specified by using XSL Formatting Objects (FOs). The XSL extension by Antenna House enhances the functions of multilingual formatting that are not even defined by XSL spefication.

### Font specification

The fonts for a script are specified by the "font-family" property of FO that contains the script. Even when data for the script has been created with correct character codes, characters may not be displayed or the character shape may be switched under the wrong specification of "font-family" property. It is very important to specify the property for multilingual formatting.

The value of "font-family" may be specified as the font names that appear on the Windows menu. Examples for FO are as follows:

- font-family="MS Mincho"
- font-family="MS Gothic"
- font-family="Arial"
- font-family="Times New Roman"

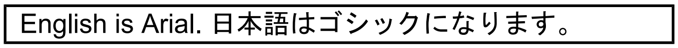
It is also possible to specify the font name using generic font family name. There are five generic font families available: serif, sans-serif, cursive, fantasy, and monospace. Once the value of the font family property is specified using a generic font family name, XSL Formatter takes up the font name actually installed in the operating Windows environment. The matching list of generic font family to actual font name by language can be set up selecting "Format Options" -> "Language-Fonts,i18n" tab. Select "Language" then specify generic font family setting for the language.

To deal with the problem that a single font may not contain glyphs to display all the characters in an object, "font-family" property allows authors to specify a list of fonts. If fonts are specified in the list, then application of the font from the left is prioritized. By using this feature, we can specify at once the European and Japanese fonts when a document consists of a mixture of both European and Japanese scripts.

**Formatting sample of font-family**

```
<fo:block
font-family="Arial, MS Gothic, sans-
serif">
English is Arial. 日本語はゴシックになります。
</fo:block>
```

The following is the formatted result.


English is Arial. 日本語はゴシックになります。

### Formatting mixed document of Japanese and Chinese languages

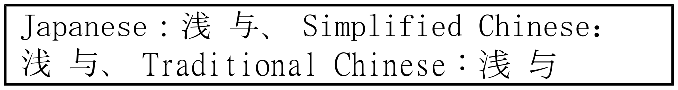
The Unicode Specification unifies Kanji of Japanese and Han of Traditional and Simplified Chinese with same shape and assigns it a single code point. But even the unified characters may have different glyphs between Japanese and Chinese languages.

Moreover, as design of popular font is also different between China and Japan, Chinese font families do not fit Japanese documents. Consequently, when we use Japanese, Traditional Chinese, and Simplified Chinese together, we should not use generic-font family but specify a definite family-name for each language.

**font-family setting for Japanese and Chinese**

```
<fo:block>
<fo:inline font-size="12pt" font-
family="MS Mincho">
Japanese：浅 与
</fo:inline>
、
<fo:inline font-size="12pt" font-
family="SimSun">
Simplified Chinese：浅 与
</fo:inline>
、
<fo:inline font-size="12pt" font-
family="MingLiU">
Traditional Chinese：浅 与
</fo:inline>
</fo:block>
```

This is formatted as follows.


Japanese：浅 与、 Simplified Chinese: 浅 与、 Traditional Chinese：浅 与

### Multilingual mixtured within a paragraph

There are difficult problems when we use many kinds of languages in one paragraph.

**Baseline adjustment**

One of the issues is how to align font baselines when there is a mixture of languages in the text. There are many fonts with the baseline at the bottom of the character (e.g. Latin characters), fonts with the baseline at the top (hanging baseline; e.g. Hindi characters), and fonts of which the lower edge becomes the baseline (kanji or Chinese characters). XSL specification defines properties for baseline adjustment.[(2)]

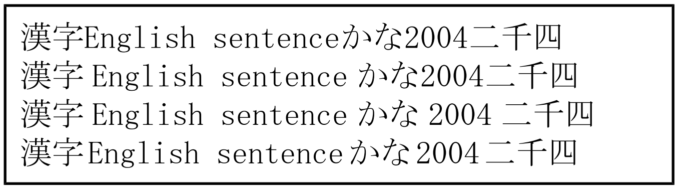
**Automated adjustment of spaces between different scripts**

In Japanese formatting, it is general to insert a narrow space between characters that belong to different scripts. This function of auto-spacing is prescribed not in XSL but in CSS3 Text Module. The XSL extension by Antenna House defines the "axf:text-autospace" and "axf:text-autospace-width" property to specify a space between ideographic and other characters. XSL Formatter can automatically adjust the space between ideographic and non-ideographic characters.

**Example of axf:text-autospace setting**

```
<fo:block font-size="12pt" padding="4pt"
xmlns:fo="http://www.w3.org/1999/XSL/For-
mat"
xmlns:axf="http://www.antennahouse.com/
names/XSL/Extensions">
<fo:block axf:text-autospace="none">
漢字English sentenceかな2004二千四
</fo:block>
<fo:block axf:text-autospace="ideograph-
alpha">
漢字English sentenceかな2004二千四
</fo:block>
<fo:block axf:text-autospace="ideograph-
numeric, ideograph-alpha">
漢字English sentenceかな2004二千四
</fo:block>
<fo:block axf:text-autospace="ideograph-
numeric, ideograph-alpha" axf:text-auto-
space-width="0.12em" >
漢字English sentenceかな2004二千四
</fo:block>
</fo:block>
```

This example is formatted as follows.



漢字English sentenceかな2004二千四
漢字 English sentence かな2004二千四
漢字 English sentence かな 2004 二千四
漢字 English sentence かな 2004 二千四

### Writing direction and XSL

In XSL, the default value of the line and character progression direction is the horizontal writing mode of English script, but other progression directions can be freely specified.

**Writing-mode**

The progression direction of characters and lines can be defined by specifying the "writing-mode" property for whole or parts of a document. However, the "writing-mode" can be specified only in the areas that are generated from the following FO. For example, as we cannot write from right to left by specifying the "writing-mode" for "fo:block," we have to place the "fo:block" into "fo:block-container."

- fo:simple-page-master
- fo:region-body
- fo:region-before
- fo:region-after
- fo:region-start
- fo:region-end
- fo:table
- fo:block-container
- fo:inline-container

Japanese and Traditional Chinese vertical writing modes can be specified as 'writing-mode="tb-rl".' Also, writing direction for scripts written from right to left such as Arabic or Hebrew can be specified as 'writing-mode="rl-tb".' If 'writing-mode="rl-tb"' is specified to a page, for example, the progression direction of a column in a multicolumn changes simultaneously. If 'writing-mode="rl-tb"' is specified to the table object, the rows are placed from right to left.

**UnicodeBIDI and "fo:bidi-override"**

Determining writing direction of characters in mixed multilingual scripts is a more complex task. As above mentioned, Unicode defines "The Bidirectional Algorithm" (UnicodeBIDI) specification to solve multilingual character mixing problems. UnicodeBIDI is adapted as "fo:bidi-override" in XSL. Details of UnicodeBIDI and "fo:bidi-override" will be explained later in the section of 'Using Arabic Language.'

[2] Refer to "Internationalized Text Formatting in CSS and XSL" by Steve Zilles for further details. The implementation of a baseline adjustment feature is not yet completed in XSL Formatter.

### Location of line breaking

The most important thing in formatting of text is to determine positions of the line breaking. The method for determining them is different depending on the language, especially script. Scripts are generally classified into two categories; Script with and without a space between words. Scripts without a space between words is further divided into two categories. One is the script which breaks lines between any characters and the other is the script which breaks lines at word boundary.

**Scripts with a space between words**

English, European languages, Arabic, Hangeul, and modern Indian languages

**Scripts without a space between words**

**Line breaks between any characters**

Japanese, Traditional Chinese, and Simplified Chinese

**Line breaks at word boundary**

Thai, Cambodian, and Laotian

Normally, line break of western languages occurs after sentence punctuation or at word space, word break by hyphenation is also admitted. In Japanese or Chinese ideographic scripts, line breaking can be located between any ideographic characters. In Thai, Cambodian, and Laotian, a kind of computer dictionary to find word boundary is necessary to decide the line breaking.

Multilingual formatting engine should be able to process line breaking differently for each script. XSL Formatter operates three ways of determining the position of line breakings depending on scripts. The computer dictionary can be used only for Thai at now.

In order to specify a candidate position for line breaking in a paragraph, you may insert a Unicode character U+200B (zero width space) at the position. XSL Formatter adds the position to candidates of points for line breaking.

### Hyphenation

If the scripts are the type of line breakings between words, the number of letters and characters in a line might decrease when a long word comes at the end of the line and the word is forwarded to the beginning of the next line. The length of line varies depending on the number of letters and characters in the line. Consequently, hyphenation function is necessary to average the length of lines by breaking words at the end of lines. XSL defines a few properties to specifie ON/OFF status of hyphenation function and to adjust the frequency of hyphenations.

XSL Formatter implements the hyphenation algorithm of TeX that was developed by Franklin Mark Liang as a default. Default hyphenation pattern dictionary included within distribution of XSL Formatter is Liang's original dictionary for English.

Hyphenation point in a word is determined by using a pattern dictionary for each language. By preparing a pattern dictionary written in XML, hyphenation for the language will be possible. You need to prepare the dictionary of the language except English by yourself. The format (DTD) of dictionary of XSL Formatter is the same as that of Apache FOP hyphenation dictionary. Therefore it's possible to use the hyphenation dictionary for FOP as it is.

Further, "Hyphenologist" by Computer Hyphenation Ltd. is available as an option for XSL Formatter. "Hyphenologist" provides you with the capability to hyphenate 40 or more languages.

In XSL, properties such as "country" or "language" (xml: lang may be used instead of country and language pair) can be specified in "fo:block," etc. Because hyphenation dictionary may be changed depending on these properties, you may use hyphenation for each language in whole document, each page, or each sentence.

### Justification

In XSL, "text-align" property applied to "fo:block" object may specify justification. Justification method shall be changed by languages. Although word spacing may change slightly in English, we should specify hyphenation property so as not to vary the space quantity.

Word spacing should not change in Arabic. For this reason, justification of Arabic script can be achieved by inserting a glyph called Kashida between characters to control the word length.

In Japanese and Chinese, justification is accomplished by adjusting the space between ideographic characters. However, if there is any European word in a line, the parts containing European words should follow the rule of Latin script.

In Thai, because line breakings occur at word boundary or at a sentence break, the length of a line easily varies. However, hyphenation is not used except for Sanskrit words. If we use justification for Thai, there is a risk that the result of justification might not become good-looking.

Although justification can be specified by XSL, the actual layout depends on the formatting engine that operates the justification.

### Line breaking between symbols, English characters, and numbers

The Unicode Standard publishes "The Line Breaking Properties" (UAX#14) that specifies the line breaking properties for every character. UAX#14 prescribes the normative line breaking properties for characters such as U+00A0 (No Break Space), U+200B (Zero Width Space), or U+2060 (Word Joiner). XSL Formatter is compatible with UAX#14 for these normative properties.

However, UAX#14 is loose for other characters and it should be customized not to create line breaking between symbols, English characters, and/or numbers. The XSL expansion "axf:line-break-at-punctuation-in word" by Antenna House can be used to define the frequency of the line breaking between symbols, English characters, and/or numbers.

### Japanese computer formatting

Japanese printing industry specifies a lot of original rules, such as treatment of punctuation or parenthesis. If we want to make them use computer formatting engine, we should create a formatting engine that implements these Japanese formatting rule.

Currently, these rules are not prescribed in XSL, but the effort to prescribe them in CSS3 is continued. Antenna House is trying to extend the XSL specification and implementing them in XSL Formatter.

## Using Thai language

Among the standard fonts in Windows 2000, both Tahoma and Microsoft Sans Serif support the range of Thai characters. If you go to Regional Options (Windows 2000) or Regional and Language Options (Windows XP) and add 'Thai,' the following Thai fonts are additionally installed.

- Angsana New
- AngsanaUPC
- Browallia New
- BrowalliaUPC
- Cordia New
- CordiaUPC
- DilleniaUPC
- EucrosiaUPC
- FreesiaUPC
- IriUPC
- JasmineUPC
- KodchiangUPC
- Lily UPC

Input Thai language and try formatting. Use SC Unipad, a Unicode text editor. In Unipad, the codes for the Thai language can be inputted by referring to the corresponding Unicode code chart. Angsana New (16pt) was specified for Thai for the example.

```
Angsana New font family, 16 point size is
specified to Thai language
นี่อะไรคะ
[Translation] What is this ?
หนังสือพิมพ์ภาษาไทยครับ
[Translation] Thai language newspaper.
```

There is no inter-word spacing in Thai. However, the line-break location is basically a word boundary. For this reason, check the word boundary by using a dictionary to determine the line-break location. In XSL Formatter V2.5, a feature that can automatically start a new line with a word boundary by using Window's Uniscribe is added. The following example shows the start of a new line by locating the break in the word "school."

```
Word [School]
โรงเรียน
โรงเรียนโรงเรียน
โรงเรียนโรงเรียนโรงเรียน
```

โรงเรียนโรงเรียนโรงเรียนโรงเรียน
โรงเรียนโรงเรียนโรงเรียนโรงเรียน
โรงเรียน

The following example shows that the start of a new line by locating the break in the word "school" is changed if the vowel in the word is miss-spelled.

โรงเรึยน
โรงเรึยนโรงเรึยน
โรงเรึยนโรงเรึยนโรงเรึยน
โรงเรึยนโรงเรึยนโรงเรึยนโรงเรึยน
โรงเรึยนโรงเรึยนโรงเรึยนโรง
เรึยนโรงเรึยน

The following shows the sample of the mixture of Japanese and Thai.

ศ、สの後のรはしばしば発音されません。
動詞の前にการ kaan やคความ khwaam を付けると、
動詞が名詞化されます。

## Using Arabic language

Let us now use Arabic. Among standard fonts in Windows 2000, the following five fonts support the range of Arabic characters:

- Arial
- Courier New
- Tahoma
- Microsoft Sans Serif
- Times New Roman

Note that Andalus, Arabic Transparent, Simplified Arabic, Simplified Arabic Fixed, and Traditional Arabic, which are added in Regional Options in Windows 2000, cannot be used as the embedding of fonts is prohibited.

First, we have an example of a document that includes the opening of the United Nations' Universal Declaration of Human Rights in only Arabic. Since Arabic characters run from right to left and this property is defined by the Unicode Database, the section in Arabic will be written from right to left by simply starting to write in Arabic.

**Sample of Arabic**

```
<fo:block
  font-family="Tahoma"
  language="ar">
Arabic (Omitted)
</fo:block>
```

This is then formatted as in the following. Since the progression direction of the text that includes this paragraph is set up for left-to-right writing, Arabic lines end up as left-justified. Also, the period is located at the right edge.

الإعلان العالمي لحقوق
الإنسان
الديباجة
لمّا كان الاعتراف بالكرامة المتأصلة في جميع
أعضاء الأسرة البشرية وبحقوقهم المتساوية
الثابتة هو أساس الحرية والعدل والسلام في
العالم.
ولما كان تناسي حقوق الإنسان وازدراؤها قد
أفضيا إلى أعمال همجية آذت الضمير الإنساني.
وكان غاية ما يرنو إليه عامة البشر انبثاق عالم
يتمتع فيه الفرد بحرية القول والعقيدة ويتحرر من
الفزع والفاقة.

In XSL-FO, we can change the direction of writing in the middle of the region by specifying the "writing-mode." As the writing-mode can only be set up for regions that generate a reference area, the paragraph in Arabic is put into a "fo: block-container". If 'writing-mode= "rl-tb"' is specified for this "fo:block-container," then the entire region becomes set up as written from right to left, therefore the paragraph begins from the right. The period is also located at the left edge.

**Sample of Arabic written from right to left**

```
<fo:block-container
  writing-mode="rl-tb"
  font-family="Tahoma"
  language="ar">
<fo:block>
Arabic Arabic Arabic
```

```
</fo:block>
</fo:block-container>
```

It is formatted as follows.

الإعلان العالمي لحقوق
الإنسان
الديباجة
لمّا كان الاعتراف بالكرامة المتأصلة في جميع
أعضاء الأسرة البشرية وبحقوقهم المتساوية
الثابتة هو أساس الحرية والعدل والسلام في
العالم.
ولما كان تناسي حقوق الإنسان وازدراؤها قد
أفضيا إلى أعمال همجية آذت الضمير
الإنساني. وكان غاية ما يرنو إليه عامة البشر
انبثاق عالم يتمتع فيه الفرد بحرية القول
والعقيدة ويتحرر من الفزع والفاقة.

The following shows the mixture of Arabic and English.

اب ab means either *father* or *a father*, and باب bāb either *door* or *a door*.

### How to specify the progression direction in multilingual mixture document

BIDI (bi-directional) document consists of text strings that contain mixtures of multilingual characters that flow from right to left like Arabic and Hebrew and those that are composed from left to right like Japanese and English.

When characters of different progression directions are nested, ambiguity arises. In order to overcome this problem, Unicode defines BIDI processing algorithm of the character. This is mainly consists of an implicit rule based on character properties for writing direction and explicit control characters such as embedding or override-control characters.

XSL specifies "fo:bidi-override" function to be used for control BIDI problem. UnicodeBIDI and "fo:bidi-override" functions is properly implemented in XSL Formatter. The following example provides more details.

In this case, parentheses bind the Arabic text within a "fo: block."

<fo:block>ضصش (ضصش) ENGLISH</fo:block>

Parentheses are neutral characters, i.e. a character without directional properties. Generally, a neutral character is influenced by the directionality of the surrounding characters. For example, a parenthesis inserted between 'Left-to-Right' and 'Left-to-Right' characters will adopt the 'Left-to-Right' property, and a parenthesis inserted into 'Right-to-Left' and 'Right-to-Left' characters inherits the 'Right-to-Left' property. However, when a parenthesis is inserted between two other characters of opposite directional properties, the directional property of the higher or surrounding level, in this case, the "writing-mode" of "fo: block" is adopted.

Therefore, the example of "fo:block" is displayed as follows:

شصض) (شصض) ENGLISH

One of the methods which prevent this is by using the Unicode directional control characters (RLM, RLE). (3)

**Example using RLM**

```
<fo:block>ضصش (ضصش) &#x200F;ENGLISH</fo:
block>
```

**Example using RLE**

```
<fo:block>&#x202B;ضصش (ضصش) &#x202C;ENG-
LISH</fo:block>
```

The above two are displayed as follows.

(شصض) شصضENGLISH

Same results can be achieved by using "fo:bidi-override."

## Conclusion

There seems to be no product among current formatting software that can process almost all main languages of the world by only one version or one edition. Our objective remains to improve XSL Formatter to the point where it can achieve high-quality output available for publishing purpose of all global languages. Please teach us your informed ideas that make it possible to achieve this target.

Author, Tokushige Kobayashi (koba@antenna.co.jp), is appreciated your comments and requests.

(3) FO example uses Unicode LRO (U+202D) to describe character flow, in order of input from left to right. When applied to Arabic characters, which normally flow from right to left, these characters will be forced to flow from left to right and thus appear to be flowing from the wrong direction when output is displayed.

**Formatting examples in major languages**

# Japanese

## 海に沈む島

### ツバルは今

今、南太平洋に浮かぶ小さな島ツバルが、危機にさらされている。地球の温暖化で、最初に海に沈む島と想像されている。1997 年京都で環境に関する会議が開かれ 2008 年から 2012 年の間に先進国全体の温室効果ガスの排気量を、1990 年の排気量と比較して 5%以上減らすことを義務つけた。

### 温暖化防止対策

| チェック | 事項 | チェック | 事項 |
|---|---|---|---|
|  | エアコンの使用を減らす |  | ごみを減らす |
|  | テレビを付けっぱなしにしない |  | 水を出しっぱなしにしない |
|  | できるだけ車を使わず歩く |  | 紙を再利用する |

今、南太平洋に浮かぶ小さな島ツバルが、危機にさらされている。地球の温暖化で、最初に海に沈む島と想像されている。一九九七年京都で環境に関する会議が開かれ二〇〇八年から二〇一二年の間に先進国全体の温室効果ガスの排気量を、一九九〇年の排気量と比較して五％以上減らすことを義務つけた。

# Hebrew

## האי הטובע בים

### מה קורה ב"טובל"

בימים אלה, האי הקטן "טובל" אשר בדרום הפסיפיק, עומד בפני סכנה. בעקבות התחממות כדור הארץ, נראה שטובל הוא האי הקרוב ביותר לטבוע בים. בשנת 1997 נערכה בקיוטו ועידה שעסקה בנושאים הקשורים באיכות הסביבה, ובה נקבע כי בין השנים: 2008-2012 יש להוריד את שיעור פליטת הפחמן הדו- חמצני במדינות המתקדמות בלפחות חמישה אחוזים (בהשוואה לשיעור פליטת הפחמן הדו- חמצני בשנת 1990).

### כדי למנוע את התחממות כדור הארץ

| בדיקה | פריט | בדיקה | פריט |
|---|---|---|---|
|  | להפחית את השימוש במזגנים |  | לייצר פחות אשפה |
|  | לא להשאיר את הטלוויזיה דולקת כל הזמן |  | לחסוך במים |
|  | להשתדל ללכת יותר, ופחות להשתמש במכונית |  | למחזר נייר |

# Arabic

- Arabic is written from right to left. As for a character, its glyph changes according to the location of character in the word: start, middle, end.

## الغوص في البحر

ماذا يحصل في توفاليو الان؟

الان، تعتبر توفاليو من الجزر الصغيرة التي تتجه نحوها الانظار العالمية. من المعتقد بان توفاليو سوف تصبح البلد الاول الذي يغوص في البحر. في عام 1997 تم عقد مؤتمر في مدينة كيوتو حول مشاكل البيئة. وفي هذا المؤتمر تم اقرار تقليل كمية ثاني اوكسيد الكاربون في الجو بنسبة اكثر من 5% خلال الفترة من عام 2008 الى 2012، مقارنتنا بعام 1990.

لمنع ارتفاع حرارة العالم

| الفقرة | الفحص | الفقرة | الفحص |
|---|---|---|---|
| التقليل من القمامة. | | التقليل من استخدام مكيف الهواء. | |
| الاقتصاد بالماء | | عدم ترك التلفزيون مفتوح. | |
| اعادة استخدام الورق | | الاعتماد على السير بـدلا من السـيارة بقدر الامكان. | |

# Thai

- Phonogramic Thai language is displayed with 42 consonants of vowel and 32 voice pitch marks.

เกาะที่กำลังจะจม

เกาะตูวาลู...

เกาะเล็กๆที่อยู่ทางใต้ของทะเลแปซิฟิกกำลังอยู่ในภาวะอันตรายตามการคาดคะเนแล้ว เกาะตูวาลูจะเป็นประเทศแรกที่จมหายไปในทะเลจากสภาวะโลกร้อน(GlobalWarming)จากการประชุมระดับโลกในด้านปัญหาสิ่งแวดล้อมที่เกียวโตเมื่อปีค.ศ.1997 ที่ประชุมได้มีมติให้ประเทศพัฒนาแล้วทั้งหมดลดปริมาณการระบายสารคาบอนไดออกไซด์ออกสู่บรรยากาศให้ได้มากกว่า 5% ในระหว่างปีค.ศ.2008 ถึง ค.ศ.2012 เมื่อเทียบกับปริมาณของสารดังกล่าวที่ระบายออกในปีค.ศ.1990

การหลีกเลี่ยงสภาวะโลกร้อน (Global Warming)

| เครื่องหมาย | รายการ | เครื่องหมาย | รายการ |
|---|---|---|---|
| | ลดการใช้เครื่องปรับอากาศ | | ลดปริมาณขยะ |
| | ไม่เปิดโทรทัศน์ทิ้งไว้โดยไม่จำเป็น | | ไม่เปิดน้ำทิ้งไว้ |
| | พยายามเดินแทนการใช้รถยนต์ | | นำกระดาษมารีไซเคิลใช้ใหม่ |

## Traditional Chinese

# 沈下大海的島嶼

### 現在的圖華路(Tuvalu)島

現在、浮在南太平洋上的小島圖華路濱臨于極大的危機。由于地球溫暖化的影響、它可能會成爲第一個沈下大海的島嶼。１９９７年在日本京都召開的有關環境的會議上、就自２００８年至２０１２年之間所有先進國家的溫室效應氣體的排氣量、做出了履行與１９９０年排氣量相比至少減少５％義務的規定。

### 溫暖化防止措施

| 檢查 | 事項 | 檢查 | 事項 |
|---|---|---|---|
|  | 少用空調 |  | 減少垃圾 |
|  | 不要將電視機開　不管 |  | 不要發生長流水現象 |
|  | 儘量步行不用汽車 |  | 紙張再利用 |

現在、浮在南太平洋上的小島圖華路濱臨于極大的危機。由于地球溫暖化的影響、它可能會成爲第一個沈下大海的島嶼。１９９７年在日本京都召開的有關環境的會議上、就自２００８年至２０１２年之間所有先進國家的溫室效應氣體的排氣量、做出了履行與１９９０年排氣量相比至少減少５％義務的規定。

## Simplified Chinese

# 沉下大海的岛屿

### 现在的图华路(Tuvalu)岛

现在、浮在南太平洋上的小岛图华路滨临于极大的危机。由于地球温暖化的影响、它可能会成为第一个沉下大海的岛屿。１９９７年在日本京都召开的有关环境的会议上、就自２００８年至２０１２年之间所有先进国家的温室效应气体的排气量、做出了履行与１９９０年排气量相比至少减少５％义务的规定。

### 温暖化防止措施

| 检查 | 事项 | 检查 | 事项 |
|---|---|---|---|
|  | 少用空调 |  | 减少垃圾 |
|  | 不要将电视机开着不管 |  | 不要发生长流水现象 |
|  | 尽量步行不用汽车 |  | 纸张再利用 |

## Korean

# 바다 속으로 가라앉는 섬

## 투발루는 지금

남태평양의 조그만 섬나라인 투발루는 지금 바다에 잠길 위기에 처해 있다. 지구 온난 현상으로 인해 최초로 바다 속으로 사라질 것으로 보인다. 1997 년 교토에서 환경에 관한 회의가 열렸고, 이 회의에서 2008 년에서 2012 년 사이에 선진국 전체의 온실 효과를 일으키는 가스의 배기양을 1990 년의 배기양에 비해 5% 이상 감소시키는 것을 의무화 하였다.

## 온난 현상 방지 대책

| 체크 | 사항 | 체크 | 사항 |
|---|---|---|---|
|  | 에어콘 사용을 줄인다 |  | 쓰레기를 줄인다 |
|  | 텔레비를 오래 켜두지 않는다 |  | 물을 절약한다 |
|  | 가능한 한 자동차를 이용하지 않고 걷는다 |  | 종이를 재활용한다 |

## English (Quoted from "The Chicago Manual of Style")

```
<fo:block hyphenate="true"
language="en">
It enables hyphenation function.
```

### 13.2

This chapter will describe some of the common problems that arise in setting technical material and will suggest ways in which these problems can be solved or circumvented. It is intended for authors unfamiliar with techniques of typesetting and for copyeditors not blessed with a mathematical background. For more on typesetting and printing in general see chapter l9.

### 13.3

The advent of sophisticated phototypesetting systems, including both photomechanical and CRT systems, has revolutionized the setting of mathematical copy in recent years. Many expressions and arrangements of expressions that formerly were impossible or very difficult to set are now relatively easy to achieve. Not every manuscript involving mathematical expressions is composed by such an advanced system, however, and authors and editors should have some idea what to expect of the particular typesetting system employed for the manuscript in hand.

### 13.4

Typesetting systems can be thought of as existing on four levels of sophistication in mathematical capabilities.

## Reference

**Extensible Stylesheet Language (XSL) Version 1.0 W3C Recommendation 15 October 2001**

http://www.w3.org/TR/2001/REC-xsl-20011015/

**CSS3 Text Module W3C Candidate Recommendation 14 May 2003**

http://www.w3.org/TR/2003/CR-css3-text-20030514/

**XSL Extensions by Antenna House**

http://www.antennahouse.com/xslfo/axf-extension.htm

**Unicode**

http://www.unicode.org/

**Internationalized Text Formatting in CSS and XSL**

http://homepage.mac.com/thgewecke/.Public/SZillesPaper.pdf

**UniPad**

http://www.unipad.org

**UnicodeFonts**

http://www.alanwood.net/unicode/fonts.html

**Office 2003 XML Reference Schemas**

http://www.microsoft.com/office/xml/default.mspx

**FOP**

http://xml.apache.org/fop/index.html

**TeX hyphenation dictionary**

http://www.ctan.org/tex-archive/language/hyphenation/?action=/tex-archive/language/

**World Script**

http://www.omniglot.com

**Universal Declaration of Human Rights**

http://www.unhchr.ch/udhr/