

EPUB3 Japanese Basic Standard

Association of Media in Digital

Draft (14 October 2011)

Voyager Japan, Inc.

INFCITY Inc.

Table of Contents

Objective	3
Introduction	3
Scope of this document.....	3
About EPUB3 Specifications	4
HTML5 in EPUB3	5
Other considerations.....	5
How to prepare EPUB Contents	5
Prepare Folders.....	5
Prepare "minetype" file.....	6
Prepare "META-INF/container.xml"	6
Prepare Package Document	7
Prepare TOC files.....	8
Create HTML5 Content Document.....	9
Create CSS file	10
Compress files with ZIP and change file extension to ".epub"	10
Detailed explanation about making of Japanese ebooks	10
Concerning representation that cannot be done with current EPUB3 specification.....	10
Restriction on various file formats	12
Vertical / horizontal writing and vertical / horizontal page progression.....	12
Package Document.....	13
TOC files.....	18
CSS.....	19
HTML5 Content Document	26
Representing comic books.....	38

Objective

Introduction

How do you produce Japanese ebooks in EPUB3 format?

There are many ways to represent Japanese text.

Does EPUB3 contain sufficient features to produce a Japanese ebook?

Yes, the standard is very supportive of Japanese texts.

In EPUB3, the developers can utilize a wide range of HTML5 and/or CSS3 features. Also, developers MAY use JavaScript, although this feature is EPUB Reading System dependent (In EPUB, "viewer" is referred to as "EPUB Reading System" or "user agent"). By combining HTML5/CSS/JavaScript, the publishers can produce ebooks with a variety of rich content. Nearly any content or interaction can be represented using EPUB3. However, dependence on EPUB Reading Systems means that EPUB3 files may lose some universality. This document is a set of instructions and standards, which attempts to explain how to reproduce Japanese books in EPUB3 format in a manner that avoids losing universality by minimizing the amount of non-content-related statements, and reusing statements where appropriate.

To achieve maximum compatability, some legitimate EPUB3 features are stated as "MUST NOT" in some section of this standard. The followings are the objectives of this document.

- To set standards for the production of universally accessible data.
- To minimize the number of statements in order to avoid complexity.
- To impose no restrictions on the reproduction of Japanese books in ebook forms (In EPUB, "viewer" is referred to as "EPUB Reading System" or "user agent")

Scope of this document

This document is based on 'EPUB3 Proposed Specification 6 September 2011' and contains necessary information to produce Japanese ebooks in the EPUB3 format, and to convert legacy Japanese ebooks to EPUB3.

There is no change between the final specification: 'Recommended Specification 11 October 2011' and the 'EPUB3 Proposed Specification 6 September 2011.'

The following features are included in the scope of this document.

- MathML (to represent mathematical formulas)
- Media Overlays (through Synchronized Multimedia Integration Language (SMIL)), and the replay of multimedia
- Encryption of contents

The most up-to-date EPUB Specification can be found at:

<http://idpf.org/epub/30>

About EPUB3 Specifications

The EPUB3 specifications (Proposed Specifications) include the following:

EPUB3 Overview

This document provides an overview of EPUB3. This part should be read first. However, it contains only a limited amount of information. Please refer to the following specification documents for detail.

EPUB Publications 3.0

This section explains mainly the Package Document. Package Document is also referred to as "EPUB rootfile" in Open Container Format Specification. In this section, the term Package Document will be used.

EPUB Content Document 3.0

This section defines which documents (HTML5, SVG, CSS) can be used as EPUB content. It also includes some explanation about the Navigation Document (TOC File).

Open Container Format 3.0

These specifications define the process to bundle files that compose EPUB ebooks into a single file (.zip). The specifications also include syntax of definition files (i.e.: container.xml).

Media Overlays 3.0

This section explains audio synchronization in EPUB3. Media Overlay 3.0 is defined as a subset of SMIL.

Others

EPUB3 Changes from EPUB2.0.1

This document clarifies which changes made from EPUB2.0.1.

Working Group Charter

States the objectives for the development of EPUB3 (at the time of draft, EPUB3 was regarded as EPUB2.1).

HTML5 in EPUB3

In principle, HTML5 in EPUB3 can be summarized as below (see EPUB Content Documents 3.0 for detail). HTML5 is sometimes referred as XHTML5. In this document, the term HTML5 refers to the following:

- Utilization of XHTML syntax, not HTML.
- Using the extension “.xhtml”.
- DOCTYPE declaration is <!DOCTYPE html>
- Media type is "application/xhtml+xml"

Other considerations

File names and folder names

For the file and folder names, developers SHOULD use commonly used web fonts, even when it is not explicitly stated. The restriction on characters set is as follow:

- English letters and numbers (ASCII). Symbols are limited to LOW LINE(_) and HYPHEN-MINUS(-).
- Files and folder names are case sensitive. However, to avoid potential problems, developers MUST NOT use hyphen and underscore in the following manner (developer should avoid the use of hyphen and underscore as prefix/suffix of file/folder names).
- Do not create two files with names that only differ in its case (i.e.: sample.jpg, Sample.jpg). Such files are distinctively recognized under UNIX filesystem, but will be read as if they were the same file in Microsoft Windows.
- Use same case in both file name and its path (i.e.: "sample.jpg" should not be specified as src="Sample.jpg"). Some operating systems can find the correct file in this situation, however, UNIX and its derivatives will not.

How to prepare EPUB Contents

The following section will provide an overview on how to create EPUB3 ebooks.

Prepare Folders

First, prepare the following folder structure.

Contents: Root folder. In practice, its folder name should not be "contents". Names that are associated with the publication being published should be used instead, i.e.: Rights management ID, name of contents being published (title).

contents/META-INF: This folder name, META-INF, cannot be changed.

contents/OEBPS: Folder contains EPUB contents.

contents/OEBPS/text: Folder contains HTML5 file.

contents/OEBPS/styles: Folder contains CSS files.

contents/OEBPS/images: Folder contains image files.

The root folder, "contents", can have any name (spaces and Japanese characters are not allowed). Hence, root folder should be named with the content being published in mind (title or rights management ID). The root folder name can be changed in the middle of production since only the contents of this folder would be packaged in the actual EPUB3 file, and assigned folder names would not be recorded. This document refers to the "contents" folder as "root folder".

"OEBPS" folder and its sub-folder can be named at the developer's discretion.

However, if there are no appropriate reasons to do otherwise, standard guidelines recommend the use of default name, "OEBPS".

Prepare "minetype" file

Create text file with content below under the root folder. The folder MUST be named "minetype". The folder name and its content are universally useable for any EPUB publications.

application/epub+zip

Prepare "META-INF/container.xml"

Create text file with content below under the "META-INF" folder. The folder name MUST be "container.xml" and it MUST NOT be encrypted.

```
<?xml version="1.0"?>
<container version="1.0"
  xmlns="urn:oasis:names:tc:opendocument:xmlns:container">
<rootfiles>
  <rootfile full-path="OEBPS/xxxxxx.opf"
```

```
media-type="application/oebps-package+xml"/>
  </rootfiles>
</container>
```

"xxxxxxx.opf"(Developer MAY change name of Package Document. In consideration of rights management, the file name SHOULD be unique to the file)

Also, developer MAY place the following files under "META-INF" folder.

```
signatures.xml
encryption.xml
metadata.xml
rights.xml
manifest.xml
```

Prepare Package Document

Package Document is an XML file that includes all metadata and composed files. Extension is ".opf". It MUST be placed under the path "META-INF/container.xml"

Basic Structure

```
<?xml version="1.0"?>
<package xmlns="http://www.idpf.org/2007/opf" version="3.0" unique-
identifier="pub- id" xml:lang="ja">
  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title>(title)</dc:title>
    <dc:creator id="creator">(author)</dc:creator>
    <meta refines="#creator" property="role" scheme="marc:relators" id="role">aut</
meta>
    <dc:language>ja</dc:language>
    <dc:identifier id="pub-id">urn:uuid:290ceb3e-c346-4bc6-b41d-000B215E83</
dc:identifier>
    <meta property="dcterms:modified">2011-09-22T08:30:00Z</meta>
  </metadata>
  <manifest>
    <item id="toc" href="toc.xhtml" properties="nav" media-type="application/xhtmll
+xml" />
    <item id="cover" href="images/cover.png" properties="cover-image" media-
type="image/png" />
    <item id="content000" href="text/content000.xhtml" media-type="application/xhtmll
+xml" />
```

```

    <item id="content001" href="text/content001.xhtml" media-type="application/xhtml+xml" />
    <item id="style001" href="styles/ebook_style.css" media-type="text/css" />
</manifest>
<spine page-progression-direction="rtl">
    <itemref idref="content000" />
    <itemref idref="content001" />
</spine>
</package>

```

Elements

Metadata

States information about the publication

Manifest

States composed files

Spine

Specify the order of replay

Prepare TOC files

TOC file is a navigation file. NCX format was used in EPUB2, but it was replaced with an HTML5 file in EPUB3.

Basic Structure

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/2007/ops" xml:lang="ja">
  <head>
    <link rel="stylesheet" href="styles/nav_style.css" type="text/css" />
    <title>(title)</title>
  </head>
  <body>
    <nav epub:type="toc">
      <h1>TOC</h1>
      <ol>
        <li><a href="text/000.xhtml#mark001">TOC entry</a></li>

```



```
.....  
</ol>  
</nav>  
</body>  
</html>
```

Create HTML5 Content Document

Content Documents are stated using HTML5 syntax. Separate files **MUST** be created if explicit page break is necessary. Structure and style are defined separately, HTML5 for structure and CSS for style.

Basic Structure

```
<?xml version="1.0" encoding="utf-8"?>  
<!DOCTYPE html>  
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://  
www.idpf.org/2007/ops" xml:lang="ja">  
  <head>  
    <meta charset="UTF-8" />  
    <link rel="stylesheet" href="../styles/ebook_common.css" type="text/css" />  
    <link rel="stylesheet" href="../styles/ebook.css" type="text/css" />  
    <link rel="stylesheet" href="../styles/ebook_v.css" type="text/css" class="vertical"  
title="縦組" />  
    <link rel="alternate stylesheet" href="../styles/ebook_h.css" type="text/css"  
class="horizontal" title="横組" /> <title>(title)</title>  
  </head>  
  <body class="top">  
    ...  
  </body>  
</html>
```

For those elements that do not included in XHTML, namespace **MUST** be specified.

- prefix namespace URI
- epub <http://www.idpf.org/2007/ops>
- m <http://www.w3.org/1998/Math/MathML>
- pls <http://www.w3.org/2005/01/pronunciation-lexicon>
- ssm1 <http://www.w3.org/2001/10/synthesis>
- svg <http://www.w3.org/2000/svg>

epub and svg tags are **REQUIRED** for Japanese ebooks. Other tags **MAY** be used.

Create CSS file

CSS defines styles for structures defined in HTML5 statements. The CSS file **MUST** be placed under "OEBPS/styles". Legitimate CSS that can be applied to EPUB content are CSS2.1 and the following modules from CSS3.

- Speech
- Fonts
- Text
- Writing Modes
- Media Queries
- Multi-Column Layout
- Ruby Positioning (MUST be used with -epub- prefix i.e. -epub-writing-mode)

Compress files with ZIP and change file extension to ".epub"

Compress files prepared with ZIP program. 'minetype' file **MUST** be the first file in compressed zip file. To complete creation of EPUB file, change the ZIP file's extension to ".epub".

Detailed explanation about making of Japanese ebooks

Concerning representation that cannot be done with current EPUB3 specification.

Representations that needs alternate methods due to the format of reflowablebooks.

Citation (Wari-cyu)

It is difficult to specify the area of Wari-cyu when it traverses across more than one lines or pages. Therefore, to represent Wari-cyu, developers **MUST** employ alternate means of representation such as reducing font size and displaying them within parentheses, or by creating a separately linked page.

Free layout within page

In principal, developers cannot represent book content in fixed layout (specification does not ensure correct rendering).

Desired features in the next version of EPUB

This section lists representation methods that are currently difficult to archive in EPUB (specification does not ensure correct rendering), but important, and therefore desired. Some are already included in the scope of ‘Advanced Adaptive Layout Charter’.

Spread-page representation

Currently, specifications do not define the way to place the content on the center of two-page or single-page spreads. This document defines alternate methods; however, native support of such a feature is desired.

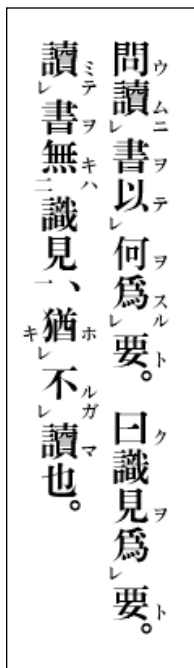
Clickable map

Clickable maps function appropriately when an image is displayed in its original size. This is due to the fact that a region specified as clickable in its original map image shifts as image sizes are reduced by the EPUB Reading System or to fit inside of single page of comic books. This is a frequent occurrence and therefore we expect EPUB to comply with "image that is clickable even when image is not displayed in its original size."

Illustration display method

When one side of two-page spreads is dedicated to illustration, developers currently use the float element to flow text. However, if there are methods to display illustrations in a way that is independent from text flow, and to display them in predefined pages (i.e. 3rd page of every chapter), they can be utilized in production of "light" novels.

Return marker for classical Chinese text



漢文の返り点 (Kanbun-no-kaeriten) or return marker for classical Chinese text, i.e., "marks placed alongside characters indicating their Japanese ordering is to be 'returned' (read reverse)" Wikipedia <http://en.wikipedia.org/wiki/Kanbun>

Currently, developers may specify ue-tsuki (upper placement) and shita-tsuki (lower placement). However, return marker and furigana cannot be specified at the same time, as in Kanbun. This feature may not materialize in the next version of EPUB, but a feature to display Kanbun in a future edition of EPUB is desired.

Display of vertical / horizontal mixed writing mode without use of Alternate Style.
Because EPUB3 does not currently support a logic property, vertical writing in Windows system MUST use the font name with "@" mark; developers are REQUIRED to create vertical and horizontal stylesheets and use them with Alternate Style. A solution to this inconvenience is desirable.

Restriction on various file formats

This section states restriction on files that are raw materials of EPUB contents. These restrictions are not necessarily stated in EPUB specifications, and certain EPUB Reading System may not be able to display large image/video files. Thus, this document specifies the practical upper limit on file sizes. These numbers may change in future edition of this standard.

JPEG

MUST be RGB or gray-scale JPG (MUST NOT be CMYK). Progressive format is not allowed. The upper limit on file size is 2048x2048.

PNG

MUST be still image and MUST NOT contain animation. The upper limit on file size is 2048x2048. If the images (JPG, PNG) are created using Photoshop, it SHOULD be saved for "Web use."

SVG

SVG 1.1 available.

The upper limit on file size is 2048x2048px.

Vertical / horizontal writing and vertical / horizontal page progression

There is no commonly used terminology for vertical / horizontal writing mode and

vertical / horizontal page progressions. To avoid confusion, this standard defines terms above as follows:

Vertical / horizontal writing mode

It represents order of letters, top to bottom (vertical writing), or left to right (horizontal writing). It corresponds to CSS property --epub-writing-mode.

Vertical / horizontal page progression

It represents the direction of page progression, either right-to-left or left-to-right. Normally when page progression is right-to-left, vertical writing mode is utilized. If page progression is left-to-right, then horizontal writing mode is utilized. This is a book wide setting and therefore defined inside of the spine page-progression-direction, which can be found in Package Document. In some case, direction of page progression are called "right binding" or "left binding" as in a physical book.

Package Document

Root element package

Example

```
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="pub-id"
version="3.0" xml:lang="ja">
```

"root element" is a top most element in XML documents tree structure.

Required Property

version:

In EPUB version 3, property value is "3.0".

unique-identifier:

ID to define book unique ID. This is not same as Unique ID. Assign character string such as "pub-id" to property value. This value **MUST** be unique within the scope of Package Document, however, it **MAY** be reused in other publications. In other words, "pub-id" can be a fixed value. The unique ID of the ebook is specified in **metadata dc:identifier** property.

Other Property

xml:lang:

for Japanese ebooks, the value is "ja".

xmlns:

specify namespace, value is <http://www.idpf.org/2007/opf>

Order of child elements of package element

In the order of: metadata, manifest, spine.

Publication information is stated in metadata.

Example

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>(title)</dc:title>
  <dc:creator id="creator">(author)</dc:creator>
  <meta refines="#creator" property="role" scheme="marc:relators">aut</meta>
  <dc:publisher>(name of publisher)</dc:publisher>
  <dc:language>ja</dc:language>
  <dc:identifier id="pub-id">urn:uuid:290ceb3e-c346-4bc6-b41d-000B215E83</
dc:identifier>
  <meta property="dcterms:modified">2011-09-22T08:30:00Z</meta>
</metadata>
```

metadata elements does not have property, but to specify namespace, developer MUST state xmlns:dc="http://purl.org/dc/elements/1.1/

Required child element**dc:identifier**

Specify unique ID. The value of unique ID can be UUID, DOI, ISBN, or ISSN. For academic articles, if DOI is already assigned, use DOI. If DOI has not been assigned, then this standard RECOMMENDS assigning UUID.

For ISBN, developers MAY specify them separately in meta element.

Use the string specified in unique-identifier element within package (in above example, the value is "pub-id") to specify property ID (REQUIRED). Universal Unique Identifier (UUID) is a universally unique ID. In the UNIX operating system, UUID can be generated with uuidgen command. In Windows systems, UUID can be generated using tools, programs, and various Web services (do not use character string used in example, it is just an example). State generated UUID with "urn:uuid:" prefix. When using DOI, state them with "urn:dpi:" prefix (i.e. urn:doi:UUID). Once UUID is specified and EPUB has been distributed, UUID SHOULD NOT be change to other value. For publication update, do not change ID, but specify revised

date in UTC format as in example below,

i.e. :<meta property="dcterms:modified">2011-01-01T12:00:00Z</meta>.

dc:title

Specify the title.

dc:language

Specify the language. For publications written in Japanese, the value is "ja". The value is assigned to the child element, not property value, therefore use following syntax <dc:language>ja</dc:language>. This element MUST be specified only once. No multiple entries are allowed.

dcterms:modified

Specify the modified date.

Syntax for other information

Statements with meta element.

Example: specifying author's name

```
<dc:creator id="creator">夏目漱石</dc:creator>  
<meta refines="#creator" property="role" scheme="role">aut</meta>  
<meta refines="#creator" property="alternate-script" xml:lang="en">Soseki Natsume</  
meta>
```

There is no field to specify author's name, therefore specify author's name as creator then define its role as author (aut).

Example of specifying publisher's name :

```
<dc:publisher>(Publisher's name)</dc:publisher>
```

The developer may specify the type of publication using publication-type element as in following example.

```
<dc:type>comic</dc:type>.
```

Child element is a optional character, but fixed value in this document are;

comic:comic

dictionary: dictionary

When it is explicitly specified multiple titles such as sub-title, use display-seq element.

Reading for sort purpose: file-as

Series: group-position

Required entry of publication information in this standard

REQUIRED publication information for EPUB are dc:identifier, dc:title, dc:language, dcterms:modified. However, this standard strongly RECOMMENDS providing the following information as well.

- Read of title
- Read of title for sorting
- If it is a series publication, then volume number for the series

manifest element

Manifest element states content composed files. Manifest element has no property. Developer may state item element as child element of manifest element and place content composed files within. The number of item element MUST be the same as the number of content files.

Item element

Examples:

```
<item id="toc" href="toc.xhtml" properties="nav" media-type="application/xhtml+xml" />
```

```
<item id="content000" href="text/content000.xhtml" media-type="application/xhtml+xml" />
```

State following property.

id: Required property

ID of the file. Referred by spine itemref element. The value of ID MUST be unique within Package Document, but otherwise optional. For simplicity, this standard RECOMMENDS the use of file name without its extension (i.e. content000.xhtml to content000)

href: REQUIRED property: Specify composed files by relative path from Package Document

media-type: Required property. Specify media type of file. The following are commonly used media type.

- application/xhtml+xml: XHTML files (HTML5 file within EPUB3 Documents)
- image/jpeg: JPG file
- image/png: PNG file

- text/css: CSS file

properties: optional property.

Developer MAY state them if the file is meaningful within content.

Property value examples:

cover-image: Cover image

nav: Navigation files.

svg: if svg files are used in the XHTML files, then specify this value.

Order of content reply: spine

States the order of reply defined in manifest element.

Property of spine element

page-progression-direction: defines direction of page progression. Property values are below

ltr: left-to-right (horizontal page progression / left-binding). Normally specified in horizontal page progression (left-to-right) ebook.

rtl: right-to-left (vertical page progression, right-binding). Normally specified in vertical page progression (right-to-left) ebook.

default: EPUB EPUB Reading System dependent.

itemref element

Example:

```
<itemref idref="content000" />
```

```
<itemref idref="content001" />
```

Statements order is important. In the above case, the order of reply is "content000", then "content001". itemref element has the following properties.

idref: REQUIRED property

Specify item's ID defined in manifest item.

linear: OPTIONAL property

This property is normally omitted. If used for file that is independent from content flow and referred by it, such as footnote or endnote, then assign property value "no".

properties:OPTIONAL property. Specifies a page to display in a two-page spread as in case of comic or visually rich magazine. Property values are below.

- "page-spread-left": display content on left page.
- "page-spread-right": display content on right page.

TOC files

Navigation document. File name is OPTIONAL, but the name SHOULD resemble that it is a TOC file (i.e. toc.xhtml).

EPUB2 specification utilized NCX Document with namespace `http://www.daisy.org/z3986/2005/ncx/` (`media-type="application/x-dtbnex+xml"`), but in EPUB3, NCX Document was replaced by XHTML Document, which utilized `nav` element in HTML5 (`media-type="application/xhtml+xml"`). This change allowed developer to prepare a navigational file in HTML5 and eliminated the need to prepare a special file.

Below is the difference of HTML5 used in TOC file and content files.

Specify "nav" as property value of child element item of manifest element. State table of contents entry inside of nav element. Specify property in `epub:type` property.

Structure of TOC file

```
<?xml version="1.0" encoding="utf-8"?> <!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:epub="http://www.idpf.org/2007/ops" xml:lang="ja">
  <head>
    <link rel="stylesheet" href="styles/nav_style.css"
type="text/css" />
    <title>(title)</title>
  </head>
  <body>
    <nav epub:type="toc">
      <h1>toc</h1>
      <ol>
        <li><a href="text/000.xhtml">TOC entry</a></li>
        .....
      </ol>
    </nav>
```

```
</body>
</html>
```

Rules

- Only nav element can be the direct child element of body element.
- Specify heading (h1~h6) immediately following nav element.
- Only heading elements (h1~h6) and ol element can be direct child element of nav element. “a” element and span element is a child element of ol li elements.

About style

Considering it is used in EPUB EPUB Reading System's TOC feature, it is not necessary to define style.

In EPUB EPUB Reading System, ol li element MUST NOT contain numeric values (process as list-style:none)

nav element

Child element of body element.

Accept property epub:type. REQUIRED property is "toc". All other property value (page-list, landmark) is optional.

ol element

Child element of nav element.

The order of ol li must follow order that is defined in spine element of Package Document

Property value of epub:type property other an toc

page-list

<nav epub:type="page-list"> Shows order of pagination. MAY use hidden property.

landmark

<nav epub:type="landmark"> Corresponds to guide element (Not recommended in EPUB3) of Package Document.

CSS

When explaining HTML/CSS, normally HTML is explained first as it represents the structure of the document, followed by explanation of CSS, which defines style. However, in case of Japanese ebook, styles such as vertical writing, vertical / horizontal mixed writing, kenten (emphasis dot) has significant importance and therefore this document explains CSS (style) first.

CSS in EPUB3

This is redundant, however, CSS features that can be utilize in EPUB3 are CSS2.1 and following modules in CSS3.

- Speech
- Fonts
- Text
- Writing Modes
- Media Queries
- Multi-Column Layout
- Ruby Positioning

These modules are still in Working Draft stage and are not yet included in CSS3 proposal. Use -epub- prefix when using those modules.

Example: -epub-writing-mode

In this document, -epub prefix are added to the vocabulary as specified in EPUB specification, however, the current Google Chrome or Webkit Nightly Build does not support -epub prefix. Therefore, prefix -webkit must be added when testing with above-mentioned browsers.

Example:

```
-epub-text-combine:horizontal;  
-webkit-text-combine:horizontal;
```

How to define style

There are multiple ways to state CSS, but this standard limits itself to following.

- Page progression direction for a book is defined in the spine element in Package Document.
- Prepare stylesheets that can be used universally in any ebook, then link them first (refer to sample style sheet provided with this document).
- Prepare basic style sheets for individual ebooks.
- Prepare separate style sheets for both horizontal and vertical page progression. This is for when EPUB EPUB Reading System switch page progression direction. Minimize style statement within head element of HTML5 Content Document.
- Style SHOULD NOT be stated directly to tags inside of content of HTML5 Content Document.
- Define in class, not in ID.
- Situation dependent sector SHOULD NOT be used (i.e. `h1 em { color:red;}`、 `h1 > em { color:red;}`)
- MAY use multiple entry to class i.e. ` Bold Red `

Using Alternate Style

It is simpler to define styles for both vertical and horizontal page progression in a single style sheet. However, currently margin, padding direction, and format in Windows systems must be stated separately for each writing mode. If EPUB supports logic property, the need to use font name with "@" mark in Windows systems is eliminated, and style sheets can be combined into a single file. This document recommends the following practices: a) use contents style sheet to define classes and styles for selectors, and b) margin, padding, and other selectors/classes that specify document formats SHOULD be defined with common selector/class for vertical and horizontal page progression, then cascade them.

- Common Stylesheet: ebook_common.css
- Basic Stylesheet: contents_style.css
- Vertical page progression stylesheet: contents_style_v.css
- Horizontal page progression stylesheet: contents_style_h.css

For publication with vertical writing content, make link as below inside of HTML5 head element. Name of vertical page progression style sheet is "contents_style_h.css".

```
<link rel="stylesheet" href="../styles/ebook_common.css" type="text/css" />
<link rel="stylesheet" href="../styles/contents_style.css" type="text/css" />
<link rel="stylesheet" href="../styles/contents_style_v.css" type="text/css"
class="vertical" title="縦組"/>

<link rel="alternate stylesheet" href="../styles/contents_style_h.css" type="text/css"
class="horizontal" title="横組"/>
```

Common Style Sheet

List of universally usable classes. Developers might separate structure and style, but there is always common style.

Following can be used in any publication from any publishers.

```
/*justify*/
.left {
    text-align:left;
}
.center {
    text-align:center;
}
.right {
    text-align:right;
```

```

}
/*justify left/right*/
img.left {
  float:left;
}
img.left {
  float:right;
}
/*Color */
.blue {
  color:#0000ff;
}
.green {
  color:#008000;
}
.red {
  color:#ff0000;
}
/*vertical/horizontal mixed writing mode*/
.tcy {
  -epub-text-combine:horizontal;
  -webkit-text-combine:horizontal;
}
/*emphasis dot*/
.em_accent {
  -epub-text-emphasis-style: filled sesame;
}
.em_dot {
  -epub-text-emphasis-style: filled double-circle;
}
.em_circle {
  -epub-text-emphasis-style: open dot;
}
.em_disc {
  -epub-text-emphasis-style: filled dot;
}
}

```

Basic Stylesheet for individual ebooks

In Windows system, font-family has to be specified using Alternate Style since the font for vertical writing mode needs @ prefix. Developer SHOULD NOT specify font-family in Basic Stylesheet except when specifying western language font set. In principal, --epub-writing-mode SHOULD be specified in Alternate Style. If fixed writing direction (vertical/horizontal) is desired, regardless of its page progression

direction, class selectors SHOULD be included in Basic Stylesheet. Indentation is specified in different manner depending on writing direction and it SHOULD be included in Alternate Style.

Page margin

```
@page {  
  margin:8%;  
}
```

Specified page margin for paged media. Developer MAY specify margin-top, margin-bottom, margin-left, margin-right individually. Considering that an ebook may be viewed on variety of display sizes, this standard does NOT RECOMMEND the usage of px to specify page margin.

For paged media, developer SHOULD use @page to specify margin, however, if the target is continuous media (paged media), then developer SHOULD specify margin in body element.

Fitting images within a page

```
@media all and (orientation:portrait){  
  img.pagefit {  
    width:100%;  
  }  
}
```

If window is oblong, then specify image width as 100%. This style is for publication that is desirable to display image in largest size and still fit within the page(s) such as cover page, two-page or single-page spread, or comics.

```
@media all and (orientation:landscape){  
  img.pagefit {  
    height:100%;  
  }  
}
```

Type selector

Book wide style is applied using type selector. The style specified using type selector MAY be overwritten by class selector.

```
body {
```

```
font-size: 20px;
line-height: 175%;
letter-spacing: 0px;
text-align: justify;
}
```

Developers MAY change individual values. The font size specified in here sets standards for the rest of the book, therefore, specify them using px. Use of px to specify font size is only allowed in this section. Use this font size as standard and set the rest of font size or line spacing using % mark. Text-align is specified using left or justify except in publication like poems. Justify are likely to provide desired result than left in many cases.

```
h1,h2,h3,h4,h5,h6 {
font-size: 110%; font-weight: bold;
}
```

Specify style for heading elements. In many cases, headings are overwritten by class selector.

```
a {
text-decoration: none;
}
```

link without underline

```
a:link {
color:#00008b;
}
a:visited {
color:#00008b;
}
```

specify color of link

Class selectors

Type selectors specify overall style. Class selectors specify styles for pages that are different from the style specified by type selectors (overall style). The style specified using class selectors include heading style for each type, text-align, and text decoration. In principal, settings such as vertical/horizontal writing (--epub-writing-mode), format (font-face), padding, margins SHOULD NOT be specified using class selector. Those settings SHOULD be specified in vertical page progression style sheets or horizontal page progression style sheets. Regardless of page progression direction setting in EPUB Reading Systems, settings such as fixed vertical/horizontal writing mode (cover page, colophon), vertical/horizontal writing mode, format, margin, and/or padding are

SHOULD be specified in here.

Vertical page progression style sheet.

In principal, prepare same type selectors and class selectors used in Basic Style sheet. Then specify settings such as writing mode (--epub-writing-mode), format (font-face), margin, and padding.

Format

To maintain compatibility with Windows system, developer SHOULD use @ prefix for format name. This standard recommends stating them in the following order.

For Mincho font:

```
font-family:"@Hiragino Mincho ProN","@IPAexMincho","@M S 明朝","serif";
```

gothic format (bold):

```
font-family:"@HiraKakuProN-W6","@M S ゴシック","sans-serif";
```

gothic format (normal):

```
font-family:"@HiraKakuProN-W3","@M S ゴシック","sans-serif";
```

Setting margin and padding

Margin and padding sets physical space and direction. If padding is specified with padding-top and writing mode is vertical, then whole text block will be indented. If class is specified with padding-top inside of vertical page progression stylesheet, then it MUST be replaced with padding-left to archive desired result in horizontal page progression stylesheet.

To specify indentation (location of first letter of line) of list items (use minus sign with text-indent), developers MUST also specify padding-top. Below are example of class that represent one-letter indentation.

```
.indent {  
  padding-top: 1em;  
  text-indent: -1em;  
}
```

Normally, it is desirable to have empty line before and after the indented text block. The below example shows how to represent them.

```
.indent {  
  margin-right: 1em;  
  margin-left: 1em;  
  padding-top: 1em;  
  text-indent: -1em;  
}
```

Horizontal page progression stylesheet.

As a general principal, prepare same type selectors and class selectors used in Basic Stylesheet. Specify vertical/horizontal writing mode (--epub-writing-mode), format (font-face), padding, and margin.

Specify format

This standard RECOMMENDS stating them in the following order.

For Mincho font:

```
font-family:"Hiragino Mincho ProN","IPAexMincho","M S 明朝","serif";
```

gothic format (bold):

```
font-family:"HiraKakuProN-W6","M S ゴシック","sans-serif";
```

gothic format (normal):

```
font-family:"HiraKakuProN-W3","M S ゴシック","sans-serif";
```

Specify padding and margin

margin-* and padding-* specify physical space and direction. If padding is specified with padding-left, and writing mode is horizontal, then the whole text block would be indented. If the class is specified using padding-top in vertical page progression style sheet, then it must be replaced with padding-left in horizontal page progression style sheet to archive the desired result. To specify indentation (specify the location of first letter of line) of list items (use minus sign with text-indent), developers MUST also specify padding-left. Below is an example of class that represents one-letter indentation.

```
.indent {  
  margin-top:1em;  
  margin-bottom:1em;  
  padding-left:1em;  
  text-indent:-1em;  
}
```

Normally, it is desirable to have empty line before and after the indented text block. The example below shows how to represent them.

HTML5 Content Document

Basic Structure

```
<?xml version="1.0" encoding="utf-8"?>  
<!DOCTYPE html>  
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://
```

```

www.idpf.org/ 2007/ops" xml:lang="ja">
<head>
  <meta charset="UTF-8" />
  <link rel="stylesheet" href="../styles/ebook_common.css" type="text/css" />
  <link rel="stylesheet" href="../styles/ebook.css" type="text/css" />
  <link rel="stylesheet" href="../styles/ebook_v.css" type="text/css" class="vertical"
title="縦組" />

  <link rel="alternate stylesheet" href="../styles/ebook_h.css" type="text/css"
class="horizontal" title="横組" />

  <title>(タイトル)</title>
</head>
<body class="top">
...
</body>
</html>

```

Characters

Accepted character range

EPUB3 accept UTF-8 characters. EPUB can render fonts that are not included in JIS X 0208 (normally referred to as Shift-JIS), but to do so, developers MUST prepare corresponding fonts separately. For Mac OS X 10.5+, Windows Vista/7+, or iPad, developers MAY use AdobeJapan1-6, but available fonts are limited in all other platform (iPhone can display AdobeJapan1-6, but only in gothic format).

However, since many characters included in JIS X 0208 have to be treated as user specified characters in many cases, this standard specify free IPAex Mincho as standard font for Japanese ebooks.

Surrogate pair

Depending on EPUB Reading System, some might not render fonts included in JIS X 0213:2004 depending on platform, if it is a part of surrogate pair. For example, character “𑀓” is represented in Unicode as 20B9F(D842+DF9F), but this Standard discourages its use since it may not be displayed in certain situation even when its typeface is included in font set.

Use of user specified character

User specified character may be represented in EPUB3 in the following way.

- PNG image
- SVG image
- SVG font

- WOFF

SVG has advantage of low operation cost, however, this standard discourages its use at this time partly because it is not part of EPUB3 requirements. WOFF is the ideal choice, but considering the production process, creating user specified characters using PNG image ensures the highest universality at this time (there might be minor difference in size of standard font and user specified characters even when its size is specified in class `:img.gaiji{width:1em;height:1em;}`). This standard RECOMMENDS the use of PNG or SVG image to represent user specified characters and limit use of WOFF to the case where character property is required. However, WOFF might cause problems in vertical writing mode under Windows systems.

User specified character by image

When using PNG images for user specified character, they SHOULD be black-and-white transparent PNG in rectangular shape (64x64 ~128x128), or 32bits PNG with anti-alias. If given user specified character is Unicode, then use "u" prefix followed by code (i.e. `u4e04.png`) for the file name. If a given user specified character is not Unicode, but has another identification code, then use a letter identification code for the prefix of the file name. If given user specified character does not have any identification code associated with it, then practice best naming conventions. EPUB Reading Systems do not comprehend file names; however, having good naming practice avoids the creation of the same characters with different name or vice versa. Developer MAY also use SVG image to represent user specified character. The difference of it from SVG font is its lack of character property. Developer MAY change SVG font color or select them (there might not be any need to select characters if it does not have associated typeface). Use the following convention.

In CSS, use

```
img.gaiji {  
  width:1em;  
  height:1em;  
}
```

which included in Common stylesheet

In Package Document, add the image to manifest element.

In HTML5 Contents Document, specify "gaiji" (use specified character) as class, then specify its phonetics or alternate character as alt. If possible, add comment to describe character.

User specified character by WOFF

Developer MAY use WOFF user specified character if character property is important.

For publications written in western languages, it is possible to embed font in EPUB documents since alphabets has small file size. However, JIS X 0213 has file size of about 10MB hence it is not practical to embed them in EPUB documents. In principal, the use of WOFF SHOULD be limited to user specified characters. Making of WOFF is out of scope of this document. Use the following convention to use WOFF (example below use font name "HanaMinMinus.woff").

Place WOFF font file under "/OEBPS/font".

Add under manifest element in Package Document. Media type is "application/font-woff". Define Basic style sheet as follows.

```
@font-face {
  font-family: "myFont";
  src: url("HanaMinMinus.woff") format('woff');
}
.myFontClass {
  font-family: "myFont";
}
```

In HTML5 Contents Document, markup as following:

```
<span class="myFontClass">■</span>
```

Replace ■ with characters defined in WOFF. WOFF cannot be used for vertical writing mode in Windows system because WOFF cannot be used for horizontal / vertical writing mode (font with @ prefix). Currently, there is no way to know if this is due to the specification or a bug in Window's Webkit implementations. Also, because of the reason stated, the developers cannot know if it is necessary to specify vertical writing fonts for vertical writing mode. Therefore, this Standard discourages the use of WOFF at this time.

Specifying Style

For the composition of style sheet and method to link them from HTML5 Contents files, refer to "Use of Alternate Style".

Semantic markup

Developers MAY utilize semantic markup in EPUB3 since EPUB3 specifications employ HTML5 and include proprietary property epub:type. Developer should use as much semantic markup as possible to ensure maximum accessibility; however, this standard regards its use as OPTIONAL at this time. For accepted vocabulary for semantic markup and its syntax, refer to <http://idpf.org/epub/vocab/structure/>. This standard explains properties commonly used in Japanese ebook.

epub:type

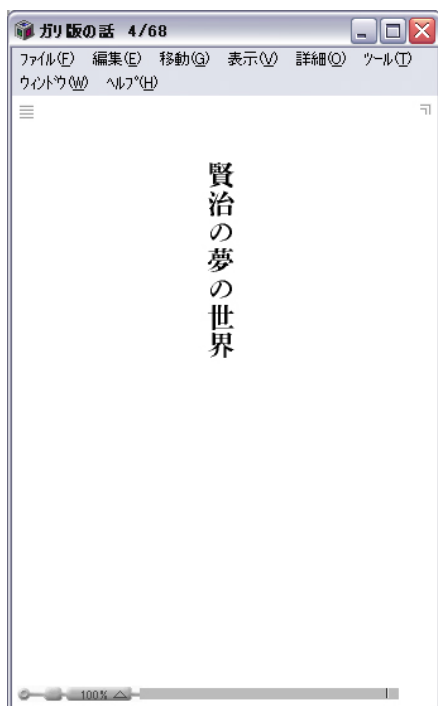
Only selected tags accept epub:type property. Tags accept epub:type property is section, body, aside, h1~h6, hgroup, and "a" tag. However, this also depends on available vocabulary.

Common vocabulary and its use

- For cover page, add epub:type="cover" inside of body tag.
- For main title page, add epub:type="volume" inside of body tag.
- For the first page of chapter, add epub:type="chapter" inside of body tag. If it is desirable to include TOC within main content, then enclose TOC with section tag `<section epub:type="toc">`
- For colophon, enclose colophon content with section tag `<section epub:type="colophon">`

Spread-page representation

In Japanese books, paged media is often placed on the center of two-page or single-page spreads.



This representation cannot be done under current version of EPUB. Developer MAY use table layout (table-cell style, not table tag), but this may not be supported by an EPUB Reading System. Also, use of table layout contradicts with the simplicity principal employed by the separation of structure and style.

This standard proposes following two methods.

- Presume paged media cannot be placed center of page, so use empty space in front to adjust spacing.
- Add `-eif-vertical-text-align:middle;` with bender prefix inside of head element in two-page/single-page spreads file and let EPUB Reading System render them.

The above two may be stated as follows:

```
<style type="text/css">
body {
  margin-right: 30%;
}
@page {
  -eif-vertical-text-align:middle;
}
</style>
```

Background and font color

Developer SHOULD not specify background and font color when planning to use EPUB EPUB Reading System default colors (normally white background with black font). Developer MAY choose any background and font color.

Heading

Heading and structure

- Headings of publication such as chapter, section, paragraph SHOULD correspond to heading level of structure. It is up to the publisher to use/decide if chapter level heading is h1 or h2, however, following the rules MUST be complied.
- Heading level should be consistent across each level (chapter, section, paragraph). For example, if the first chapter heading is markup as `<h1>`, then second chapter MUST be markup as `<h2>`.
- If chapter is marked as h1, then the following section MUST be h2. Moreover, if chapter is h1 and section is h2, then paragraph MUST be h3 (if chapter is h2, then section and paragraph MUST be h3 and h4 respectively).
- The heading in HTML5 documents should be structured as h1, h2, h3 and so forth. However, in EPUB explicit page break is done by creating separate HTML5 documents so some documents have only one heading level. This is unavoidable.

Specifying class

Specifying block indentation and/or empty line around them SHOULD be done by specifying class in style sheet.

Markup when image is used for heading

Sometimes image is used for heading instead of a character string, but heading SHOULD still be included as document structure. There are multiple ways to accomplish this, but this standard RECOMMENDS including image inside of heading.

Example

If use of image is desired in heading page, like in `<h2></h2>`, then specify heading that would not be display as follows. Also, if `display:none` is used, developer SHOULD not assign ID and use as link destination.

Example

```
<h2 style="display:none">Chapter 1</h2>

```

Link

Links inside of contents

- Link is specified by a href.
- Link destination is marked by id.
- id(s) may place inside of h1~h6, div, p, span, or a. (this standard does not recommend empty tag of span such as ``)
- Add # to id when used with a href. This standard discourages the use of id for element specified to be non-displayed by `display:none`. In EPUB, page break is done by separating HTML5 file, therefore it should be specified with file name.

Example

```
<a href="contents_0001.xhtml#mark01">link</a>
```

In order to use a link within a clickable map, the map image MUST be displayed in magnification (factor of). In other words, since image size change is not synchronized with region specified as clickable, a clickable map is difficult to use when its display size often changes according to device display sizes (smartphone, PC).

External Links

External link is also specified by a href. Protocol (scheme) is EPUB EPUB Reading System dependent (`http://`、`https://`、`mailto://`、`tel://`).

Main content and paragraph

"Paragraph" in western languages is not the same as paragraph (danraku) in Japanese. In western languages, the space indented depends on the number of spaces entered by the user and by the EPUB Reading System; however, in Japanese, the full size space occupies exactly one-character length. Also, the space between paragraph and the space between lines are usually the same in Japanese, but in western languages, the space between paragraphs are sometimes wider than the ones used between lines. Therefore, if p element is used to mark paragraphs, the space between paragraphs becomes wider than the one of space between lines (under current version of Webkit, the space between paragraphs become wider when ruby annotation exists within a paragraph).

This standard RECOMMENDS following the convention when dealing with paragraphs.

- The only elements that can be direct child element of body element is block elements.
- All block elements are inside of p or div elements, except heading.
- If paragraph display style is same, then use
 for line break and indentation by full space.
- If it is desirable to have empty line within paragraph to represent passing-time or scene change, use
 in succession (use of
 in succession is usually discouraged, but this Standard believes this is the right implementation).
- If indentation of text block is desired for the use with quotation, then treat them as separate blocks. Enclose the block with div element, then use style to specify empty lines before/after the block text. Also, the length of empty line is the same as default empty line, use integer value to specify them (unit is em).

Example: style for vertical writing mode.

```
.start {
  margin-right:1em; /*one-linespace before text block*/
  margin-left:1em; /*one-line space after text block*/
  padding-top:2em; /*two-letter indentation*/
}
.indent {
  margin-right:1em; /*one-line space before text block*/
  margin-left:1em; /*one-line space after text block*/
  padding-top:2em; /*two-letter indentation*/
  text-indent:-2em; /*two-letter indentation*/
}
```

Specify content

<p>
鎌田慧の「ビラまき三年カキ八年」によれば、ガリ切り筆耕時代の賢治は、
.....はいう。

町の活版所におけるジョバンニの労働に、.....旅へとつれだす。

</p>

<div class="start">

<p>

(当時のわたしは)一枚の原紙で二、三千枚刷ることができた。一日一万枚刷
るくらい軽かったもんさ。.....もはや相手にされなくなってしまったのだ。

</p>

</div>

<p>

プロのプリンターとしての.....かれはくやしそうだった。

あとでしらべてみてわかった。.....しかし笑った側もまったくまちがってい
たわけではない。

</p>

<div class="indent">

<p>

第十条 日本国民たる要件は、法律でこれを定める。

</p>

<p>

第十一条 国民は、すべての基本的人権の享有を妨げられない。この憲法が国
民に保障する基本的人権は、侵すことのできない永久の権利として、現在及び
将来の国民に与えられる。

</p>

</div>

View

鎌田慧の「ピラまき三年カキ八年」によれば、ガリ切り筆耕時代の賢治は、……はいう。

町の活版所におけるジョパンニの労働に、……旅へとつれだす。

(当時のわたしは)一枚の原紙で二、三千枚刷ることができた。一日一万枚刷るくらい軽かったもんさ。……もはや相手にされなくなってしまうのだ。

プロのプリンターとしての……かれはくやしそうだった。

あとでしらべてみてわかった。……しかし笑った側もまったくまちがっていたわけではない。

第十条 日本国民たる要件は、法律でこれを定める。

第十一条 国民は、すべての基本的人権の享有を妨げられない。この憲法が国民に保障する基本的人権は、侵すことのできない永久の権利として、現在及び将来の国民に与へられる。

Indentation, block indentation, and list

Block indentation

Block indentation is when certain text block is displayed lower than the normal (in vertical writing mode, character is indented lower, and in horizontal writing mode, character is indented to right). It is sometimes referred to as "indentation", but this standard defines indentation as "displacement of first character of top most line in paragraph" and distinguishes it from block indentation.

Lists and indentation

Lists (ol, ul) can be represented in horizontal writing mode, but lists are not suitable in vertical writing mode (cannot use Chinese numbers on numbered lists). Thus, in vertical writing mode, the "list" is represented using indentation, not li or ol. If it must be a numbered list, then use list.

Ruby annotation

Use HTML5 ruby element to represent ruby annotation.

```
<ruby>奮<rt>ふん</rt>戦<rt>せん</rt>記<rt>き</rt></ruby>
```

The statement above is simpler than the ruby annotation in XHTML. In XHTML, rp element is used for EPUB Reading Systems not supporting XHTML ruby, but use of rp element is no longer recommended. Ruby support is REQUIRED in EPUB EPUB Reading System.

Image

Image MAY be specified as inline object or use float to wrap text around (left or right depending on writing mode), but it cannot be anchored within the page. For one image occupying the whole page to be displayed, as in case of cover, two-page/single-page spread, or comic, specify one image only in HTML5 Contents Document, then define style to fit image within the page.

Inline image

If image is simply specified, then it is treated as inline image.

Float

When it is desirable to place image on page, use float property, top/bottom on vertical writing mode and left/right on horizontal writing mode, and have text wrap around. When float is specified, clear property MUST be used to cancel float.

Caption

It is often desired to display text below image as caption. Representing caption with markup in EPUB might cause the image and caption to be displayed in separate pages depending on the EPUB Reading System. Currently, it is more appropriate to combine image and caption as a single image than use markup to specify the caption. SVG format can display caption clearly even when image is enlarged. Also, by adding caption text on img element alt property will improve accessibility.

SVG image

SVG format illustration, such as graph or table, created by Illustrator or other software MAY be used in EPUB documents. SVG format MAY also be used for bitmap image with text caption such as picture or illustration. When SVG format image is used, specify media-type as "image/svg+xml" under manifest in Package Document. To embed SVG format to HTML5, state the following inside of img tag.

```

```

SVG table

Table created for EPUB may be displayed across multiple pages undesirably depending on EPUB Reading System and screen size, just as in image and its caption. Currently, it is more desirable to prepare table as image. SVG format is able to display text clearly even when image is enlarged.

Character decoration

Use span tag, not font tag, to markup character to decorate then specify style. If multiple styles are specified, then latter one takes precedence. For example, if "emphasis dot" (-epub-text-emphasis-style) is specified in multiple locations, then only the last one takes effect.

Vertical/horizontal mixed writing mode

Vertical/horizontal mixed writing mode can be specified in common class "tcy" in Common Stylesheet.

Example

```
<span class="tcy">9</span>月<span class="tcy">12</span>日
```

Lower placement or upper placement has limited use.

Define class below and specify them in span. Do not markup with sub or sup tags. This MAY be used in both horizontal and vertical page progression, so these classes can be included in Basic Stylesheet.

```
.sup { /*upper placement (ue-tsuke)*/  
  font-size:50%;  
  vertical-align:super;  
  letter-spacing: -2px;  
}  
.sub { /*lower placement (shita-tsuke)*/  
  font-size:50%;  
  vertical-align:sub;  
  letter-spacing: -2px;  
}
```

Kanbun can be represented this way, but return mark and furigana cannot be used at the same time. Currently, there are no methods to specify both return mark and furigana at same time.

Example:

```
国破<span class="sup">レテ</span>山河在<span class="sup">リ</span><br />烽火  
連<span class="sub">二</span>三月<span class="sub">一</span>
```

Vertical writing mode and horizontal writing mode SHOULD NOT be used within single page.

When both vertical and horizontal writing mode is used within single page (or within single HTML5 Contents Document), the EPUB specification does not ensure correct rendering in case of overflow, therefore, developer SHOULD not use both modes of writing within a single page. To specify cover page or horizontal writing mode in colophon or main content, split HTML5 files and specify --epub-writing-mode inside of body element. Developers MUST NOT specify --epub-writing-mode to child elements of body element.

Representing comic books

Package Document

Basic structure

```
<?xml version="1.0"?>
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="pub-id"
version="3.0" xml:lang="ja">
  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title>(title)</dc:title>
    <dc:creator id="creator">(author)</dc:creator>
    <meta refines="#creator" property="role" scheme="marc:relators" id="role">aut</
meta>
    <dc:publisher>(name of publisher)</dc:publisher>
    <dc:language>ja</dc:language>
    <dc:identifier id="pub-
id">urn:uuid:C19BE97E-99CA-4DE5-9062-96EC6B0DC2F0</
dc:identifier>
    <meta property="dcterms:modified">2011-09-23T00:00:00Z</meta>
    <dc:type>comic</dc:type>
  </metadata>
  <manifest>
    <item id="comic_style0001" href="styles/comic.css" media-type="text/css" />
    <item id="toc" href="toc.xhtml" properties="nav" media-type="application/
xhtml+xml" />
    <item id="IMG00001" href="images/P01.jpg" media-type="image/jpeg" />
    <item id="IMG00002" href="images/P02.jpg" media-type="image/jpeg" />
    <item id="IMG00003" href="images/P03.jpg" media-type="image/jpeg" />
    <item id="IMG00004" href="images/P04.jpg" media-type="image/jpeg" />
    <item id="IMG00005" href="images/P05.jpg" media-type="image/jpeg" />
    <item id="XHTML00001" href="text/P01.xhtml" media-type="application/xhtmll
+xml" />
    <item id="XHTML00002" href="text/P02.xhtml" media-type="application/xhtmll
+xml" />
    <item id="XHTML00003" href="text/P03.xhtml" media-type="application/xhtmll
+xml" />
    <item id="XHTML00004" href="text/P04.xhtml" media-type="application/xhtmll
+xml" />
    <item id="XHTML00005" href="text/P05.xhtml" media-type="application/xhtmll
+xml" />
```

```
</manifest>
<spine page-progression-direction="rtl">
  <itemref idref="XHTML00001" properties="page-spread-left" />
  <itemref idref="XHTML00002" properties="page-spread-right" />
  <itemref idref="XHTML00003" properties="page-spread-left" />
  <itemref idref="XHTML00004" properties="page-spread-right" />
  <itemref idref="XHTML00005" properties="page-spread-left" />
</spine>
</package>
```

dc:type

child element of dc:type of metadata is comic:

```
<dc:type>comic</dc:type>
```

spine itemref properties

spine itemref has "properties" property. The property specifies which side of two-page spread contents would be displayed. The property value is "page-spread-left" to display contents on left side of two-page spreads and "page-spread-right" for right side of pages. For comic, page compositions are as follows:

- Cover page is left page
- Place endpaper page (back of cover page) on right, and then place title page is left page
- The page side alternates from right to left for remaining content.

Developer MAY specify properties="page-spread-left" in succession, if continuous display of content on left page is desired; however, rendering is EPUB Reading System dependent. If compatibility is of most importance, then developer MAY prepare blank page for endpaper page and alternate between "page-spread-left" and "page-spread-right" for the remaining of content to ensure proper rendering. If two-page spread is REQUIRED, then black page MUST be placed between cover page and beginning of content (endpaper page in above example). For single page, developer SHOULD NOT use a blank page, and just let EPUB Reading System render content in the order specified in spine.

CSS

Defining basic style for comics

Normally, style is NOT REQUIRED for comics since one HTML5 file specify one image, however, to fit image within a page, prepare following style.

```

@charset "UTF-8";
@page {
  margin:0px;
}
@media all and (orientation:portrait){
  img.fit {
    width:100%;
  }
}
@media all and (orientation:landscape){
  img.fit {
    height:100%;
  }
}
body {
  text-align:center;
  background-color:#eeeeee;
  margin-top: 0px;
  margin-bottom:0px;
  margin-left: 0px;
  margin-right: 0px;
}

```

Image files

Developer MAY use EPUB3 normal media type image such as JPEG, PNG, GIF, or SVG. Animation GIF is not allowed. For SVG format, EPUB specification allows the register of a SVG file on spine, but considering its effect on universality, this standard specifies embedding SVG in the HTML5 Contents Document.

HTML5

Representing comics

As a general principal, comic ebooks are composed of one image for each HTML5 file. The use of Alternate Style is not allowed on comics.

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://
www.idpf.org/2007/ops" xml:lang="ja">
  <head>
    <meta charset="UTF-8" />

```



```
<link rel="stylesheet" href="../styles/comic.css" type="text/css" />
<title>(title)</title>
</head>
<body>
  
</body>
</html>
```

About clickable map

Clickable map is useful in some TOC, but many EPUB Reading Systems support them only when clickable map is displayed in magnification. When a map image is enlarged/reduced to fit screen size, the specified region within map does not shift correctly with image enlargement/reduction, and therefore it is practically impossible to ensure intended result. TOC file is prepared for table of content entries.

Translation: Daisuke Muro
Special Thanks: Peter Brantley (Internet Archive)