

---

## iBooks Asset Guide 5.1 Revision 2



# Contents

## **Overview** 5

### Changes Made in this Release 5

#### What's New in the iBooks Asset Guide 5.1 Revision 2? 5

## **Digital Book Essentials** 9

### Overview 9

#### Multi-Touch Books 9

#### EPUB 9

### Structure of a Version 3 EPUB 9

#### Package Document (.opf) 10

#### Navigation Document 11

### EPUB 3 Features 15

#### Pop-up Footnotes 15

#### Page Progression Direction 17

#### Text Directions 17

#### Languages 19

#### Embedding Language-Appropriate Fonts 19

### Book Versioning 19

#### Version Numbers 22

#### Book Versioning in iBooks Author 22

#### Book Versioning in EPUBs 22

### Book Cover Art 23

### Recommendations for Creating Books 24

#### Presentation and Styling 24

#### In-Book Cover Page 24

#### Interior Image Requirements 24

#### DRM 29

#### Screenshots 29

#### Fonts 29

#### Audio and Video 30

#### Links 35

#### PDFs 36

#### Adobe Page Template 37

### Books with JavaScript Interactivity 37

Overview	37
Best Practices for Developing Scripts for Books with JavaScript Interactivity	37
Designing Content for Books with JavaScript Interactivity	38
Tips for Improving Performance in Books with JavaScript Interactivity	44
Technical Guidelines and Requirements	44
Testing During Development	48
Book Validation	48
Book Samples	49
About Book Samples	49

## **Guidelines for Flowing Books** 52

Overview	52
Content Structure	52
Alignment	52
Nonlinear Content	53
Scrolling in Flowing Books	55
Tables	55
Fonts	55

## **Guidelines for Fixed Layout Books** 56

Overview	56
Setting Up the Document	56
EPUB 3 Fixed Layout	57
Table of Contents	59
Using Fonts in Fixed Layout Books	61
Optimizing Images in Fixed Layout Books	61
Layout	62
Text	62
Read Aloud Books	63
Overview	63
About the Read Aloud Feature User Interface	63
Tools for Creating Read Aloud Content for iBooks	65
Media Overlays Structure	65
Processing Audio for Media Overlays	67
CSS Styling of Media Overlays	68
Ambient Soundtrack	69
Embedding Read Aloud Controls	70
Styling Read Aloud Controls	71
Adding a SMIL File and Audio File to an EPUB	72

## **Guidelines for Multi-Touch Books** 74

Overview 74

## **Revision History** 75

Previous Spec Revisions 75

Changes in the iBooks Asset Guide 5.1 Revision 1 76

Changes in the iBooks Asset Guide 5.1 78

Changes in the iBooks Asset Guide 5.0 79

Changes in the iBooks Asset Guide 4.9 79

Changes in the iBooks Asset Guide 4.8 81

Changes in the iBooks Asset Guide 4.7 Revision 4 82

Changes in the iBooks Asset Guide 4.7 Revision 3 82

Changes in the iBooks Asset Guide 4.7 Revision 2 83

Changes in the iBooks Asset Guide 4.7 83

Changes in iBooks Asset Guide 4.6 83

Changes in iBooks Asset Guide 4.5 Revision 1 83

Changes in iBooks Asset Guide 4.5 84



# Overview

This document provides delivery information for all accepted media and files for iBooks including EPUB and Multi-Touch books created using iBooks Author. It describes the basics of EPUB 3 creation, as well as guidelines specific to each type of book, EPUB and Multi-Touch. In addition to this guide, two example EPUB files are available for your reference in the Examples section on the Deliver Your Content page in iTunes Connect; one is for Flowing books (`flowing3-0.epub`) and the other is for Fixed Layout books (`fixedlayout3-1.epub`).

---

**Note:** The 5.1 and later versions of the iBooks Asset Guide cover EPUB 3 and Multi-Touch books only. For information on the older EPUB 2 format, see our documentation archive. Use this link to learn more: [https://itunesconnect.apple.com/WebObjects/iTunesConnect.woa/wa/jumpTo?page=faqIndex&q=epub2\\_asset\\_guide](https://itunesconnect.apple.com/WebObjects/iTunesConnect.woa/wa/jumpTo?page=faqIndex&q=epub2_asset_guide). You can also refer to <http://idpf.org/epub/201> for EPUB 2 information.

---

**Important:** All information, documentation, and examples must not be shared external to Apple including `iBooks.js`.

## Changes Made in this Release

Date/Version	Changes Made
August 13, 2013 - Version 5.1 Revision 2	Added design considerations for books with JavaScript Interactivity. Identified the Japanese fonts available in iBooks. When creating a new version of a book, keep the same <code>idrefs</code> in the OPF as the previous version. Clarified scrolling in Flowing books. Clarified book cover art. Increased the pixel limits for interior images. Corrected new iPad screenshot sizes. Clarified sample cutting. Corrected typos. Changed the branding of iBookstore to iBooks.

## What's New in the iBooks Asset Guide 5.1 Revision 2?

### JavaScript Interactivity

The section that describes how to design books with JavaScript interactivity has been modified to include design and technical considerations necessary for both Mac OS and iOS. See the section “[Books with JavaScript Interactivity](#)” (page 37) for more information.

## Japanese Fonts: Clarification

In vertical writing mode, characters are rotated automatically based on the font sets and the glyphs that are available to iBooks. The following lists the system-provided fonts you should use for Japanese text:

Hiragino Kaku ProN

Hiragino Mincho ProN

Hiragino Maru Gothic ProN

YuGothic

YuMincho

**Important:** If you want to use a Japanese font that is available in OS or iOS, it is strongly recommended that you use Hiragino Kaku ProN and Hiragino Mincho ProN as both are pre-installed for the reader. YuGothic, YuMincho, and Hiragino Maru Gothic ProN require the reader to download the font from the iBooks Font menu before the font is available for use.

## Book Versioning

If you create a new version of a book, keep in mind that the `idrefs` in the OPF should always refer to the same content across different versions of the book. For example, ID "html9" should refer to the same chapter in both versions of the book, even if the actual name of the HTML file or the order of the elements in the spine may have changed. By keeping the `idrefs` the same, the annotations a customer made in a chapter in one version of the book will remain with the correct chapter when synced to the newer version. If you do not keep the `idrefs` the same, the annotations end up in the wrong chapter in the new version.

## Flowing Books: Scrolling Clarification

By default, Japanese and Chinese books scroll horizontally, while all other languages scroll vertically. You can redefine the scroll direction using the `"ibooks:scroll-axis"` property in the OPF .opf file. See [“Scrolling in Flowing Books”](#) (page 55) for an example.

## Exterior Book Cover Art: Clarification

The book’s cover art that appears on the iBooks Store (also known as the marketing image) is the image delivered alongside the book asset and does not refer to the cover image included in the book asset.

The book’s cover art must use RGB color mode and should be at least 1400 pixels along the shorter axis. For best results, a good rule of thumb is to use an image that is a minimum of 300 dpi.

## Interior Book Images: Pixel Limit

The pixel limit for the interior cover, as with all interior book images, has been increased to 3.2 million pixels from 2 million pixels.

## Screenshots: Clarification

The sizes of the new iPad screenshots with the status bar removed have been corrected. Below, all screenshots are listed and the sizes are shown width by height. When the screenshots are cropped to remove the status bar, those pixels come off the height.

1024 x 768: (iPad 1/2 horizontal)

1024 x 748: (iPad 1/2 horizontal with status bar removed)

768 x 1024: (iPad 1/2 vertical)

768 x 1004: (iPad 1/2 vertical with status bar removed)

2048 x 1536: (new iPad horizontal)

2048 x 1496: (new iPad horizontal with status bar removed)

1536 x 2048: (new iPad vertical)

1536 x 2008: (new iPad vertical with status bar removed)

## Sample Cutting: Clarification

Previous versions of this guide did not specify what is meant by “beginning of the book” when describing how iBooks uses the percentage of words method to cut a sample for preview. Where the sample begins can vary depending on how the book has been structured. The words on which the percentage is calculated are counted from the beginning of the book. For flowing books, the beginning is determined by finding the first landmark where the `epub:type` attribute is `"bodymatter"`. If that attribute is specified, iBooks starts cutting the sample from the publication component in the `spine` item referenced by the `href` attribute in the landmark. If the `"bodymatter"` attribute is not specified, iBooks looks for the first landmark where the `epub:type` attribute is *not* one of the following: `"cover"`, `"frontmatter"`, `"toc"`, `"foreword"`, `"introduction"`, or `"dedication"` and starts with the spine items immediately following the component referenced in the prior `"cover"`, `"frontmatter"`, `"toc"`, `"foreword"`, `"introduction"`, or `"dedication"` landmark.

For example, if the Landmarks structure included `epub:type` attributes of `"cover"`, `"frontmatter"`, `"page1"`, and `"bibliography"`, iBooks searches the `spine` for the component referenced in the `"frontmatter"` landmark and starts the sample cutting with the spine item immediately following it.

---

**Note:** It is recommended that the publication component referenced by the `href` attribute in the "bodymatter" landmark *not* be listed close to the last item on the list of `spine` items. When cutting the sample based on percentage of words, iBooks could potentially include publication components that precede the referenced spine item, which could result in iBooks including the entire book in the sample.

---

## Corrections

Corrected typos in the Book Versioning section, and in the Table of Contents and Ambient Soundtrack sections in the Fixed Layout Books chapter. In these sections, a space was missing between `ibooks:` and `http://` in the `prefix` examples. Note that the space between `ibooks:` and `http://` is important; it needs to be a single, regular space character and not a return, non-breaking space, or any other kind of whitespace.

# Digital Book Essentials

## Overview

iBooks supports two book types: Multi-Touch (.ibooks) and EPUB (.epub). Multi-Touch books are created using the application **iBooks Author**. EPUB can be used to create Flowing books and Fixed Layout books.

## Multi-Touch Books

Multi-Touch books for iPad are created using the iBooks Author application, which generates books using the Apple-developed iBooks Format (.ibooks). Multi-Touch books can contain a wide variety of interactive elements, which makes them particularly well-suited for textbooks, cookbooks, history books, and picture books. iBooks only accepts Multi-Touch books made with the latest version of iBooks Author.

## EPUB

iBooks supports both EPUB 2.0.1 and EPUB 3. Both can be used to create Flowing and Fixed Layout books.

A Flowing book is ideal for text-heavy books. A Flowing book's text can be resized by the reader. Flowing books support images, audio, video, and interactivity.

A Fixed Layout book can be created for books that need a precise design or layout. A Fixed Layout book supports full-bleed images, audio, video, interactivity, and read aloud. Read aloud allows you to sync a narrator's voice with the text in your book.

## Structure of a Version 3 EPUB

The structure and contents of a version 3 EPUB are different from a version 2 EPUB. For a list of the differences between EPUB 2.0.1 and EPUB 3, see <http://idpf.org/epub/30/spec/epub30-changes.html>.

Most importantly, a version 3 EPUB includes a Package Document (.opf) and a Navigation Document (toc.xhtml). The Package Document includes all of the information about a book including metadata, manifest, and spine. The Navigation Document includes information about how to navigate a book and includes the Table of Contents, Landmarks, and Page-list.

The following sections break down the structure and describe the requirements.

## Package Document (.opf)

The Package Document contains information about the book including the metadata, manifest, and spine. It also defines what version of EPUB you are using (version 2 or 3). For EPUB 3, the version must be 3.0.

```
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="bookid"
version="3.0">
```

## The Metadata

You can include as much metadata as pertains to your book, but at minimum, you must include the following items:

- Title
- ID
- Language
- Modified-date (date your book was last modified)

```
<metadata xmlns:opf="http://www.idpf.org/2007/opf"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>Flowing Book</dc:title>
  <dc:identifier id="bookid">1234567890</dc:identifier>
  <dc:language>en</dc:language>
  <meta property="dcterms:modified">2012-05-04</meta>
</metadata>
```

## The Manifest

The `<manifest>` is a listing of every file that is used to create your book, for example, all content documents, CSS, fonts, images, and so on. Use the `properties` attribute to identify the Navigation Document and the cover image of the book.

```
<manifest>
  <item id="pg-1" href="chapter1.xhtml" media-type="application/xhtml+xml"/>
  <item id="css1" href="stylesheet.css" media-type="text/css"/>
  <item id="font" href="fonts/font.ttf" media-type="application/x-font-ttf"/>
  <item id="toc" href="toc.xhtml" media-type="application/xhtml+xml"
properties="nav"/>
  <item id="img1" href="cover-img.jpg" media-type="image/jpeg"
properties="cover-image"/>
</manifest>
```

## The Spine

The `<spine>` is a list that defines the linear reading order of the content documents of the book. The first item in the list is the first item in the book.

```
<spine>
  <itemref idref="cov"/>
  <itemref idref="tit"/>
  <itemref idref="ch1"/>
  <itemref idref="ch2"/>
  <itemref idref="end" linear="no"/>
  <itemref idref="bib"/>
  <itemref idref="cht" linear="no"/>
</spine>
```

Nonlinear content (`linear="no"`) includes documents that supplement the main content, but are outside the reading flow of the book. For example, nonlinear content can be charts, tables, review answers, and so on. See [“Nonlinear Content”](#) (page 53) for more information.

## Navigation Document

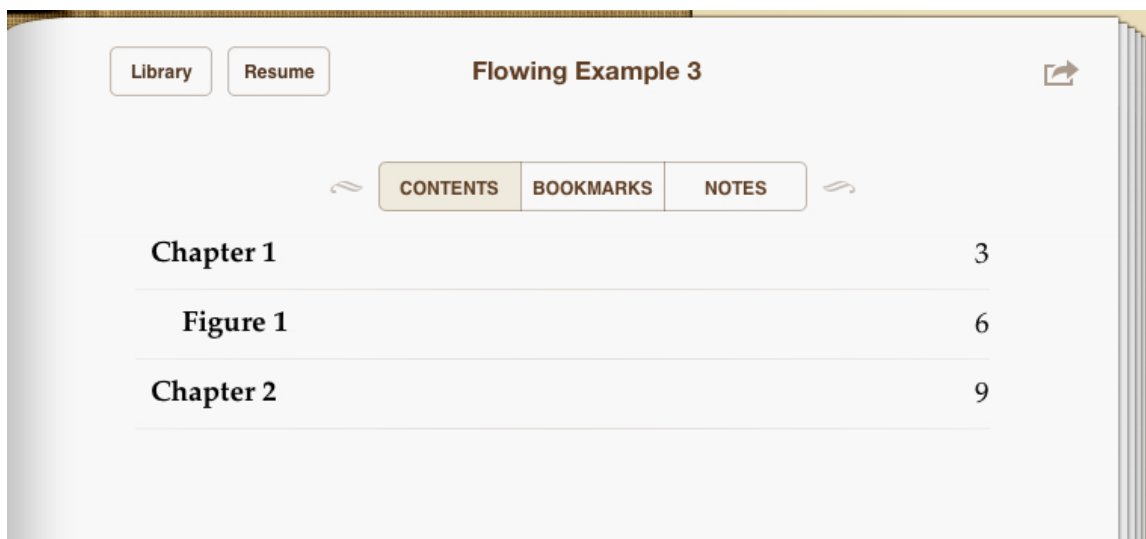
Elements in the Navigation Document are created using the HTML5 `<nav>` element. Because each navigation feature is created using the same `<nav>` element structure, you must also include an `epub:type` attribute to identify the purpose of each `nav` structure. The Navigation Document is also a content document and therefore can be visible in the pages of the book.

Navigation structures include the Table of Contents, Landmarks, and Page-list, described below. A Table of Contents is required for all books; Landmarks is required for Fixed Layout books if you do not provide a custom sample; Page-list is optional for both formats.

## Table of Contents

The main Table of Contents in iBooks is created using a `nav` element with an `epub:type` value of `"toc"`. Readers use the Table of Contents to navigate to key locations in the book. Note that you do not define page numbers for each entry. iBooks calculates those for the reader to accommodate different font and screen sizes.

```
<nav epub:type="toc">
  <ol>
    <li><a href="chapter1.xhtml">Chapter 1</a>
      <ol>
        <li><a href="chapter1.xhtml#figure1">Figure 1</a></li>
      </ol>
    </li>
    <li><a href="chapter2.xhtml">Chapter 2</a></li>
  </ol>
</nav>
```



## Landmarks

The Landmarks structure identifies key component files within the book, such as the cover page, bibliography, and so on. It is created using a `nav` element with an `epub:type` value of `"landmarks"`. The Landmarks navigation structure replaces EPUB 2's `<guide>` element. iBooks references the Landmarks when cutting the sample for a book. A Landmarks `nav` is required for Fixed Layout books if you do not provide a custom sample.



Landmarks can also be used to define the start page of a Flowing book, which is the first page a reader will see the first time they open a book. iBooks opens to the first landmark item that contains the `epub:type` value of `"ibooks:reader-start-page"`. If that value is not specified in the Landmarks navigation structure, iBooks opens to the first spine item that contains one of the following `epub:type` landmarks values:

- `bodymatter`
- `acknowledgements`
- `dedication`
- `epigraph`
- `foreword`
- `preface`
- `introduction`
- `frontmatter`

Within the package, only one `"landmarks" nav` element can be delivered.

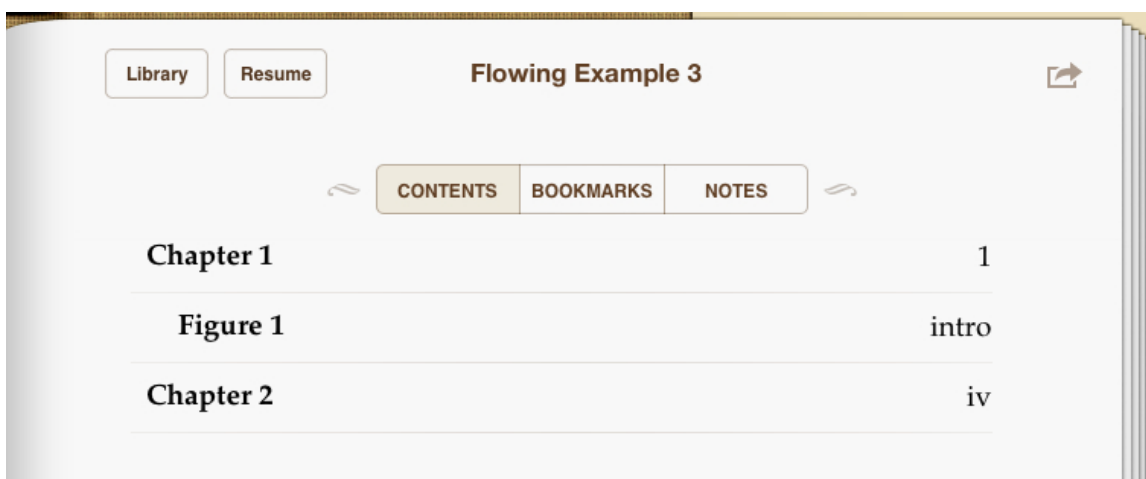
The Landmarks structure uses the `epub:type` attribute to identify both the `<nav>` element and the document functions listed within it. Apple recommends you identify all of the key files in your book. The required `epub:type` attribute describes the publication component referenced by the `href` attribute. The value for the `epub:type` attribute is case-sensitive. Apple suggests you label the first chapter of the book with an `epub:type` value of `"bodymatter"`, with all other `epub:type` attributes tagged with the appropriate type (`"toc"`, `"titlepage"`, `"epilogue"`, `"preface"`, and so on). Within the Landmarks `<nav>` block, there can be only one `epub:type` attribute of each type; for example, there cannot be multiple `epub:type` attributes of type `"bodymatter"`. For a full listing of the values available for `epub:type`, see <http://idpf.org/epub/vocab/structure/>.

## Landmarks Example

```
<nav epub:type="landmarks">
  <ol>
    <li><a href="coverpg.xhtml" epub:type="cover">Cover</a></li>
    <li><a href="titlepg.xhtml" epub:type="titlepage">Title Page</a></li>
    <li><a href="chapter.xhtml" epub:type="bodymatter">Start</a></li>
    <li><a href="bibliog.xhtml" epub:type="bibliography">Bibliography</a></li>
  </ol>
</nav>
```

## Page Mapping Using page-list

The `<nav>` element using the `epub:type="page-list"` attribute provides a method to designate pages in an EPUB that correspond to the pages of the physical book. This is especially useful in a classroom setting when the teacher instructs students to turn to a particular page. You can use the optional `epub:type="page-list"` attribute to define an empty string to that page so that it is not numbered. Similarly, you can define Roman numerals (i, ii, iii), letters (a, b, c), or numbers (1, 2, 3) for page numbers. If you use something other than numbers or a single word, make sure it is meaningful, keep it very short, and check to make sure it doesn't get truncated on the screen.



## Page-list Example

Page-list is supported for both Flowing and Fixed Layout books. The example below shows how to provide page navigation using `epub:type="page-list"`:

```
<nav epub:type="page-list">
  <ol>
    <li><a href="coverpg.xhtml">intro</a></li>
    <li><a href="titlepg.xhtml"></a></li>
    <li><a href="chp1.xhtml#p1">1</a></li>
    <li><a href="chp1.xhtml#p2">2</a></li>
    <li><a href="chp1.xhtml#p3">3</a></li>
  </ol>
</nav>
```

## EPUB 3 Features

This section describes the following features supported in EPUB 3 books:

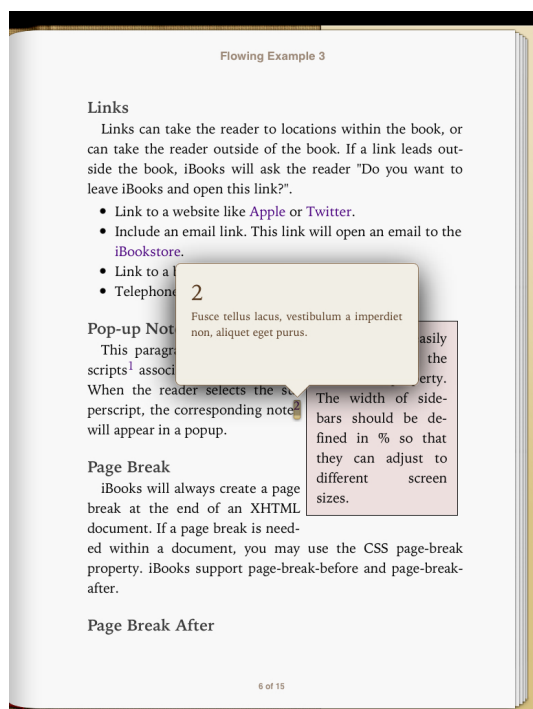
- Pop-up footnotes
- Page progression direction
- Text direction

Other features supported by EPUB 3 include embedded audio and video, read aloud (for Fixed Layout books), and interactivity, which are covered in other sections in this document.

In addition to the features listed above, MathML is also supported. See <http://www.w3.org/Math/> for details.

## Pop-up Footnotes

In EPUB 3 Flowing and Fixed Layout books, you can create pop-up footnotes by labeling footnotes with the appropriate `epub:type` values. You use two elements to create a pop-up footnote: an anchor (`<a>`) element that triggers the popup and the `<aside>` element that contains the footnote text. Both elements have an `epub:type` attribute to identify their purpose: `epub:type="noteref"` to trigger the popup and `epub:type="footnote"` to indicate the footnote's text.



In the example below, the anchor element (`<a>`) has two attributes: `epub:type="noteref"` and a link that references the location of the element that contains the popup's text.

The `<aside>` element that contains the popup's text also has two attributes:

`id="myNote"` that matches the value of the `href` attribute in the link that references it  
`epub:type="footnote"`

Because the `<aside>` element has an `epub:type` of `footnote`, the text is hidden in the main body of the book. The text will only be seen by the reader in the context of the popup.

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/2007/ops">
. . .
<p>
  <a href="chapter.xhtml#myNote" epub:type="noteref">1</a>
</p>
<aside id="myNote" epub:type="footnote">Text in popup</aside>
. . .
</html>
```

---

**Note:** Use of the `epub:type` attribute requires the inclusion of the namespace `xmlns:epub="http://www.idpf.org/2007/ops"` in the `<html>` element.

---

If your book requires a specific text direction, such as right-to-left, and you want the footnote text direction to match, wrap the footnote text in a `<p>` element and add a style to specify the text direction:

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/2007/ops">
. . .
<p>
  <a href="chapter.xhtml#myNote" epub:type="noteref">1</a>
</p>
<aside id="myNote" epub:type="footnote"><p style="direction:rtl">Text in
popup</p></aside>
. . .
</html>
```

## Page Progression Direction

By default, iBooks paginates from left to right. However, for some languages like Japanese and Chinese, you might want right to left pagination. iBooks 3.0 supports both pagination directions. You can define the pagination direction by including the `page-progression-direction` on the `<spine>` element in the OPF. The `page-progression-direction` attribute is a global attribute and therefore defines the pagination flow of the book as a whole. The allowed values are `"ltr"` (left-to-right), `"rtl"` (right-to-left), and `"default"`. If you specify `"default"` or if you do not specify the attribute, iBooks can choose the rendering direction.

```
<spine page-progression-direction="ltr">
  <itemref idref="cov"/>
  <itemref idref="tit"/>
  <itemref idref="ch1"/>
  <itemref idref="ch2"/>
  <itemref idref="end" linear="no"/>
  <itemref idref="bib"/>
  <itemref idref="cht" linear="no"/>
</spine>
```

## Text Directions

iBooks supports both right-to-left and left-to-right text direction, defined with the `dir` attribute on the `<package>` element, as described in the EPUB 3 standard: <http://idpf.org/epub/30/spec/epub30-publications.html#sec-package-elem>.

iBooks also supports both vertical and horizontal text directions using the CSS 3 Writing Modes as outlined in the EPUB 3 standard. Text direction (horizontal and vertical) is defined by the CSS `writing-mode` property. The `writing-mode` must be set on the `<body>` or `<html>` element instead of setting it on a descendent element. Supported values include: `horizontal-tb` (horizontal top-to-bottom), `vertical-rl` (vertical right-to-left), and `vertical-lr` (vertical left-to-right). Each content document can support a single `writing-mode` value. If you want both horizontal and vertical text in your book, then each text direction must be split into separate content documents.

```
html {  
  -epub-writing-mode: vertical-rl;  
}
```

**Important:** If you want to use a Japanese font that is available in OS or iOS, it is strongly recommended that you use Hiragino Kaku ProN and Hiragino Mincho ProN as both are pre-installed for the reader. YuGothic, YuMincho, and Hiragino Maru Gothic ProN require the reader to download the font from the iBooks Font menu before the font is available for use.

For information on scrolling with vertical text, such as Japanese and Chinese, see [“Scrolling in Flowing Books”](#) (page 55).

---

**Note:** Tatechuyoko must be created using the CSS property `-webkit-text-combine`, not the CSS property `writing-mode`. See the Tatechuyoko section, below.

---

Note that if the Table of Contents in the iBooks menu needs to be rendered vertically as opposed to horizontally, the text direction must be specified for the TOC in the Navigation document.

## Tatechuyoko

In vertical text, it is sometimes preferable to include a short run of horizontal numbers or Latin text known as Tatechuyoko. To create Tatechuyoko, you must use the CSS property, `-webkit-text-combine`. For example:

In the CSS:

```
.number {  
  -webkit-text-combine: horizontal;  
}
```

In the HTML of the page:

```
<span class="number">50</span>
```

## Languages

The language of your book should be defined in two places:

- The metadata section of the OPF:

```
<metadata xmlns="http://www.idpf.org/2007/opf"
xmlns:dc="http://purl.org/dc/elements/1.1/" >
...
<dc:language>en</dc:language>
...
</metadata>
```

Language abbreviations can be found in the language subtag registry (<http://www.iana.org/assignments/language-subtag-registry>).

- In the metadata provided to Apple (the language metadata provided to Apple must be the same language as specified in the OPF).

## Embedding Language-Appropriate Fonts

iBooks has a large selection of fonts available via the iOS system fonts. Some languages may require a more extensive font library. If your book requires a more extensive font library, you can embed the font. See “[Defining Book Layout Metadata](#)” (page 57) for EPUB 3 Flowing and Fixed Layout books.

## Book Versioning

For EPUB 3 and Multi-Touch books, you can provide book versioning information when you publish a new version of a book previously published to iBooks. The updated book replaces the old book on the iBooks Store and is available to future consumers. Customers who have downloaded the old version of the book will be notified that a new version is available for download, and if the customer chooses to download it, the new version will replace the prior version on their device.

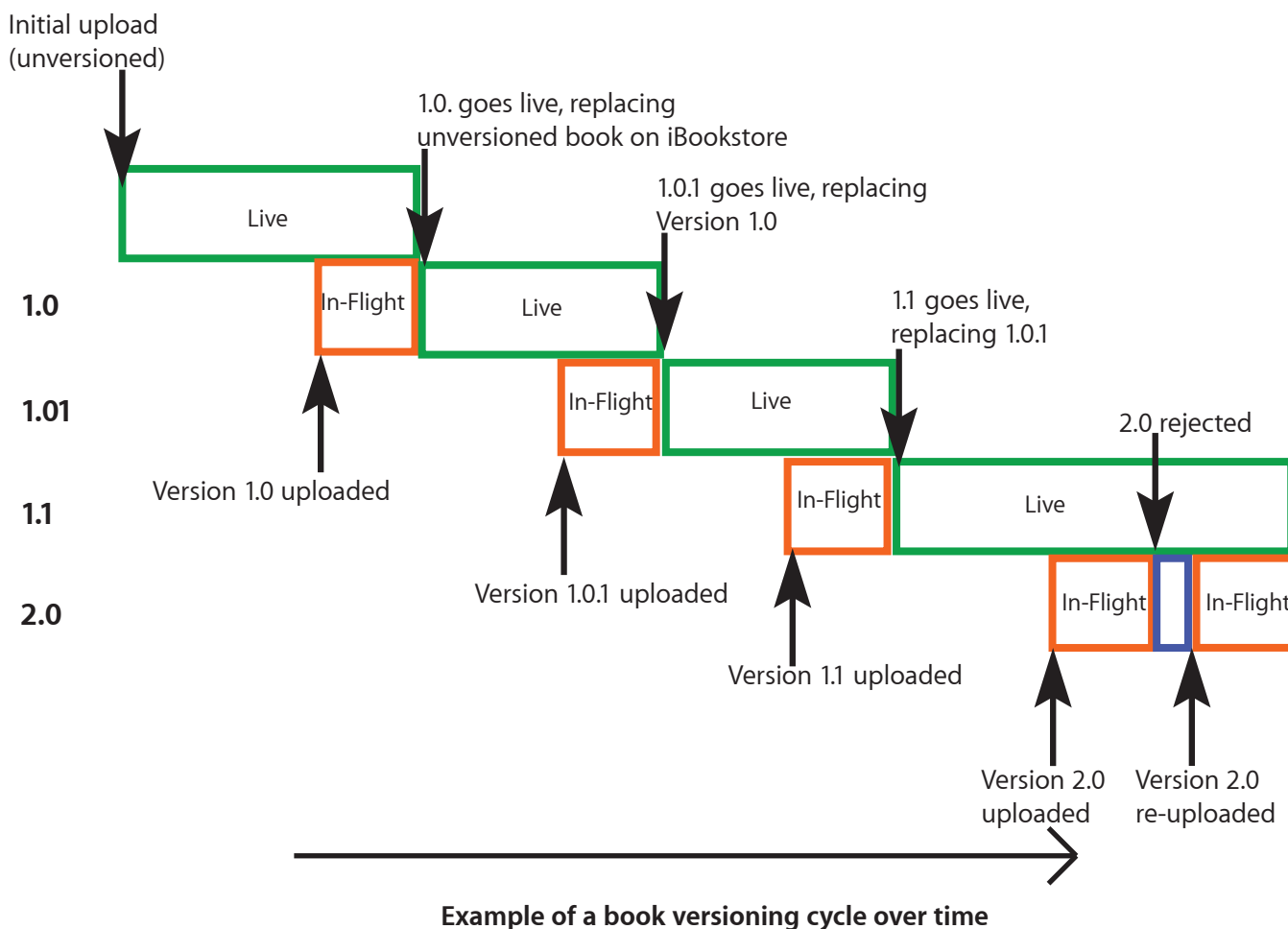
The following terms are used throughout this document to describe the various stages of a book:

- “unversioned” refers to a book that has not had versioning applied in the asset, for example, a book asset that was delivered to iBooks before versioning was available. You can make an “unversioned” book “versioned” by uploading an asset that declares its version string.
- “versioned book” refers to a book asset that has been revised with a version string or in process of being revised.
- “live” applies to the version currently on iBooks, which could be the original or a later version.
- “in-flight” applies to the version that is being or has been submitted, but has not yet gone live.
- “latest” applies to the in-flight version if there is one, and otherwise the live version.

Providing a book asset with a version string is what makes a versioned book. For EPUB 3, the version string is supplied within a `meta` element in the Package Document for the book asset. Note that sample assets cannot be versioned; sample files that contain version numbers will be blocked from delivery. Samples inherit the version number of the source asset. If the previous book version included a custom sample and you do not redeliver a new one, the old sample will be used; if the previous book version did not include a custom sample and you do not deliver one, a new sample will be cut.



At any time in the life of a book, you may have visibility to up to two versions: one version (already live or about to go live) and potentially another version currently under review. After you have uploaded a new version and it goes live, it becomes the only version visible to you until you deliver a new revision. Changes you make during a revision do not affect the “live” version. After you have made all of your changes and uploaded it, it becomes the live one, replacing the currently live version on iBooks.



To explain the changes made in the updated version to your customers, you must supply text using the `<version_whats_new>` tag. This text displays on the iBooks Store under the book's description to provide version details to your customers. Before your new version can go live, you must send an import package that includes the What's New text. If you deliver a new source asset for a versioned book and What's New text has not been supplied, you will see a warning that the What's New text is missing and must be supplied. The book will not go live until you supply it. See the *iBooks Book Package Specification* for details on delivering the import package.

Note that versioning applies only to EPUB 3 and Multi-Touch books.

## Version Numbers

In general, the first number of the version number represents a major revision; the second number would be used for a revision containing several changes/new information; the third number would be used to indicate minor changes, such as fixing a typo or formatting issues. For example, if the first version of the book was 1.0, a subsequent minor revision could be 1.0.1; a more substantial revision could be 1.1; a total rewrite could be 2.0.

Keep the following in mind when providing version numbers:

- You must enter a new version number when publishing an update to a book already available on iBooks
- Use dot-separated integers, with a maximum of two dots, three components, and four integers per component. For example: 1111.1111.1111
- 1.10 is considered larger than 1.9
- Initial zeros are ignored. 1.01 is read as 1.1
- Letters are not permitted
- Each time you update a book, at least the right-most segment of the version must change
- Apple recommends that you declare the first asset delivered to iBooks version 1.0. The version number is consumer-facing and is meant to manage delivering updates to consumers. Future versions should increment as is appropriate.

## Book Versioning in iBooks Author

Book versioning is built into iBooks Author version 2. In iBooks Author, when you publish a new version of a previously published book, you are asked to provide versioning information during the publishing process.

## Book Versioning in EPUBs

To use book versioning in iBooks for EPUB 3, you must include the following `prefix` attribute in the `<package>` element in the OPF file:

```
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="bookid"
version="3.0"
prefix="ibooks: http://vocabulary.itunes.apple.com/rdf/ibooks/vocabulary-extensions-1.0/">
```

---

**Note:** In the prefix example above, the space between `ibooks:` and `http://` is important; it needs to be a single, regular space character and not a return, non-breaking space, or any other kind of whitespace.

---

The version of your book is specified within a `meta` element in the Package Document. The `meta` element has a property value of `"ibooks:version"`:

```
<meta property="ibooks:version">1.1.2</meta>
```

In addition, you must supply some text explaining the changes made in the updated version using the `<version_whats_new>` tag in the metadata XML delivery. See the *iBooks Book Package Specification* for details.

---

**Note:** If you create a new version of a book, keep in mind that the `idrefs` in the OPF should always refer to the same content across different versions of the book. For example, ID `"html9"` should refer to the same chapter in both versions of the book, even if the actual name of the HTML file or the order of the elements in the spine may have changed. By keeping the `idrefs` the same, the annotations a customer made in a chapter in one version of the book will remain with the correct chapter when synced to the newer version. If you do not keep the `idrefs` the same, the annotations end up in the wrong chapter in the new version.

---

## Book Cover Art

- The book's cover art that appears on the iBooks Store (also known as the marketing image) is the image delivered alongside the book asset and does not refer to the cover image included in the book asset. The pixel limit for the interior cover, as with all interior book images, has been increased to 3.2 million pixels.
- The book's cover art must use RGB color mode and should be at least 1400 pixels along the shorter axis. For best results, a good rule of thumb is to use an image that is a minimum of 300 dpi.
- The book's cover art file must be a high-quality JPEG with `.jpg` or `.jpeg` extension or PNG with `.png` extension.
- The 3.2 million pixel limit on interior book images does *not* apply to the external cover/marketing image that is sent alongside the book asset.
- It is important that you do not increase the size of a smaller image to meet the minimum image size dimension standard. Excessively blurry or pixelated images will be rejected.

## Recommendations for Creating Books

The following sections provide some recommended practices when creating books.

### Presentation and Styling

Keep the following best practices on presentation in mind when developing your book.

#### Line Breaking

To prevent text from being clipped by the bounds of the content area, insert soft hyphens into long words and especially into linked text and headings. Soft hyphens are described in detail here:

<http://www.w3.org/TR/html401/struct/text.html>

#### Page Breaks

Page breaks are supported in Flowing books. If you include page breaks to mark a chapter break, use `page-break-after` to create a break at the end of a chapter, not `page-break-before` to insert the break at the beginning of the chapter. This modification improves performance with the table of contents.

To indicate that a page break should come before or after an element, set up a style in CSS using the `page-break-before` or `page-break-after` properties. Accepted values for these properties include:

- `auto`: Insert a page break before or after the element as necessary
- `always`: Insert a page break before or after the element

Below is an example of a CSS style to add a page break before all text styled as a heading 1:

```
h1
{
  page-break-before:always;
}
```

### In-Book Cover Page

- The background color on HTML in-book cover pages should be undefined. Specifying colors results in uneven, dark borders around the cover image on the cover page.

### Interior Image Requirements

Keep the following in mind when preparing images:

- All images should be prepared in digital format and should not contain any text. All text must be created using HTML. Embedding text in images creates issues that cause a large number of customer complaints: customers can't use the dictionary or search the text, and in addition, the book becomes not accessible for persons using the VoiceOver feature. Therefore, books with images that contain embedded text will be rejected from sale on iBooks.
- JPEG with `.jpg` or `.jpeg` extension (quality unconstrained) or PNG with `.png` extension.
- RGB (screen standard).
- Images that have any transparent areas should be PNG format or, ideally, use JPEGs with WebKit PNG masks. Images that do not have transparency should be JPEGs.
- Be sure to preview your book in night mode. In night mode, the transparent areas of an image will be black. If your image has dark text within a transparent image, that text could be difficult to read in night mode. Instead, we suggest you use a JPEG with a white background.
- Images (within the EPUB) cannot exceed 3.2 million pixels. Apple recommends providing images that are at least 1.5 times the intended viewing size up to a maximum of 3.2 million pixels. You can calculate whether an image inside the book file exceeds 3.2 million pixels by multiplying the height of the image with the width.
  - In Flowing books, images can be sized by adding dimensions to a wrapper element that contains the image. The dimensions must be added to the wrapper instead of the image element because iBooks uses the image's dimensions to ensure that the image will always fit on a page, no matter the page size. It is recommended to size the container using percentage units to maintain adaptability for various screen sizes. For example:

HTML:

```
<div class="image-container">
  
</div>
```

CSS:

```
img {width: 100%;}
.image-container {width: 90%; }
```

- In Fixed Layout books, publishers can scale images down using CSS. For example, an image that is defined as 40px wide in CSS should actually be 60px wide. A full-bleed image should be no smaller than 738 x 985 (the size of a single page, full screen, in iBooks on the iPad).
- Use `img` tags wherever possible, but define style attributes such as dimensions and positioning in the CSS.

- To ensure proper viewing of images in content, use the HTML `img` tag instead of wrapping images in `svg:img`.
- The maximum recommended size is about 10 MB of un-encoded image data per XHTML file.
- For accessibility, the `alt` attribute for an image must be included; the value must be an appropriate replacement for the image. It should just be what you would have put in the prose if the image was not included, as shown in the following example:

```
<p>The hillside was covered in poppies.  
  
A dog was asleep on the porch.</p>
```

The following example is not as effective; the `alt` text only describes the image, instead of being a textual replacement for the image. When reading without the image, the text does not flow as well as the example above.

```
<p>The hillside was covered in poppies.  A dog was asleep on the porch.</p>
```

The following `alt` attributes are not acceptable: `alt="none"`, `alt="nothing"`, `alt="image"`, or `alt="page 3"`. Leaving out the `alt` attribute is also not acceptable. The attribute `alt=""` is acceptable in cases where the image is decorative and does not have any content or meaning.

- Apple recommends that you run final image assets through image optimizations, such as ImageOptim (which is a free, open source optimization tool).

## Gaiji

Gaiji are small, inline images that represent characters that are not available in a character or font set. Gaiji are typically used for older symbols or characters in Japanese that have fallen out of use. iBooks controls image dimensions to ensure that images fit on the page, but with Gaiji, you may want to define a specific image size. You can use one of two different mechanisms to ensure the image dimensions you define are respected. Use these mechanisms only for small, inline Gaiji. Be sure to test these images on a small screen like an iPhone or iPod Touch.

1. If a book is written in Japanese, follows the EBPAJ guide, and has one or more class names that contain the term "gaiji", then iBooks will respect the dimensions for the images that have class names containing the term "gaiji". For example:
  - Written in Japanese

```
<metadata>
. . .
<dc:language>ja</dc:language>
. . .
</metadata>
```

- Follows EBPAJ guide 1.0 or 1.1

1.0:

```
<metadata>
. . .
<dc:description id="ebpaj-guide">ebpaj-guide-1.0</dc:description>
. . .
</metadata>
```

1.1:

```
<metadata>
. . .
<meta property="ebpaj:guide-version">1.1</meta>
. . .
</metadata>
```

- Class name(s) on the image element that is either equal to "gaiji", or is prefixed with "gaiji-"

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

img.gaiji-line {
width:      1em;
height:     auto;
}

img.gaiji-wide {
width:      auto;
```

```
height:      1em;  
}
```

2. **Custom Class Name:** If the book is not defined as following the EBPAJ guide, you can define a custom class name for which iBooks will respect an image's dimensions. The custom class name is defined in the metadata section of the Package Document (.opf) file, and requires the inclusion of the `ibooks` prefix in the package element. For example:

```
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="bookid"  
version="3.0"  
prefix="ibooks: http://vocabulary.itunes.apple.com/rdf/ibooks/vocabulary-extensions-1.0/">  
.  
.  
.  
  <metadata>  
    .  
    .  
    .  
    <meta property="ibooks:respect-image-size-class">gaiji</meta>  
    .  
    .  
    .  
  </metadata>  
</package>
```

---

**Note:** Mechanism 2 takes precedence over Mechanism 1. In other words, if a custom class is defined, then iBooks will respect dimensions for that class, and not search for class names that include "gaiji".

---

## Gaiji Accessibility

To make the gaiji image accessible for readers who use VoiceOver, do one of the following:

1. When gaiji is used to display a character that is not available in a particular font, the alternative text should be the unicode character.
2. When gaiji is used to represent an image similar to emoji, use a short description of the image's content (for example, "thumbs up", "sunrise", or "smiling woman").
3. When gaiji is used to represent a new or made-up character that does not exist in the font and has no unicode representation, use the phonetic Hiragana or Katakana string in the alternative text to represent the pronunciation of the new or made-up character.



## DRM

All text, fonts, and images have DRM (Digital Rights Management), a form of encryption, applied. Other media including audio, video, and PDF do not have DRM applied.

## Screenshots

You can provide up to five screenshots per book. The screenshots must be in PNG (.png) or JPEG (.jpg or .jpeg) format. For a cleaner look, you can optionally remove the status bar.

Below, all screenshots are listed and the sizes are shown width by height. When the screenshots are cropped to remove the status bar, those pixels come off the height.

- 1024 x 768: (iPad 1/2 horizontal)
- 1024 x 748: (iPad 1/2 horizontal with status bar removed)
- 768 x 1024: (iPad 1/2 vertical)
- 768 x 1004: (iPad 1/2 vertical with status bar removed)
- 2048 x 1536: (new iPad horizontal)
- 2048 x 1496: (new iPad horizontal with status bar removed)
- 1536 x 2048: (new iPad vertical)
- 1536 x 2008: (new iPad vertical with status bar removed)

## Fonts

A book can be styled using system fonts or embedded fonts. Both system fonts (fonts that exist on the device) and embedded fonts (developer supplied fonts within the book) are defined using standard CSS. Embedded fonts must be listed in the manifest of the OPF .opf. If you intend to provide embedded fonts in a Flowing Book or a Fixed Layout Book, you must set the "specified-fonts" option to true. See ["Defining Book Layout Metadata"](#) (page 57).

In Flowing Books, if the "specified-fonts" property is set to true, the reader can select a new font when reading the book; however, they will always have the option in the iBooks interface to return to the book's original fonts.

**Note:** iBooks' fonts follow the font orientation guidelines of [UTR \(Unicode Technical Report\) 50 draft 6](#).

## Best Practices for Fonts

- It is recommended that explicit font families only be used to achieve an intended effect, such as conveying that a note is handwritten.

- Take into account that users will have control over font face, font size, and justification in Flowing books – test your book to make sure it works in most, if not all, cases.
- Font sizes should be defined in `em` or `px`, not by name (for example, x-small, small, medium, large).
- The main text of a book should either not have a defined `font-size` or should have a `font-size` of `1em`. This will ensure ideal readability and font scaling.
- Check the font hinting and metrics on embedded fonts to avoid layout problems. Be sure to test them to avoid layout issues and cropped text.
- OpenType, TrueType, and SVG embedded fonts are supported.
- SVG text should be reserved for use of irregular text paths.
- Embedded fonts must be declared in the OPF and CSS.
- Font licensing should be verified before embedding fonts.

## Font Mangling

iBooks supports Font Mangling, a method of font encryption that is part of the EPUB spec. For more information about font mangling, see [http://idpf.org/epub/20/spec/FontManglingSpec\\_2.0.1\\_draft.htm](http://idpf.org/epub/20/spec/FontManglingSpec_2.0.1_draft.htm).

## Audio and Video

You can embed audio and video inside a book to enhance a customer's reading experience. This section describes asset requirements for video and audio and how to embed the content for EPUB books only. For audio and video requirements for Multi-Touch books, see the knowledge base articles on the Apple Support Site:

<http://support.apple.com/kb/HT5065> – Add video to your book

<http://support.apple.com/kb/PH2791> – Add a movie or an audio file

**Important:** When sending books with embedded audio or video content, keep in mind that the maximum file size for the `.zip` file is 2GB. For usability's sake, the maximum recommended size is 500 MB. Larger files take longer to download and may become unwieldy on older devices.

## Encoding Audio

Audio must be encoded using iTunes as follows (with `.m4a` as the file extension):

- Stereo
- AAC/MP4

- 256 kbps

See [“Embedding Audio and Video”](#) (page 33) for an example of embedding audio.

## Source File Recommendations for Video

### HD Source

HD videos embedded in a book must include posterframe art and meet the following requirements.

- Apple ProRes 422 (HQ)
- ITU-R BT.709 color space, file tagged correctly as 709
- VBR expected at ~220 Mbps
- 1920 x 1080 square pixel aspect ratio material
- Native frame rate of original source:
  - 23.98 progressive frames for content originally authored at 24fps and inverse-telecined
  - 24 progressive frames for content originally authored for at film frame rate in a tapeless workflow
  - 25 progressive frames for content originally authored for PAL delivery
  - 29.97 either interlaced or progressive content originally authored for NTSC delivery
  - Telecined 23.98 content containing a 3:2 pulldown will not be accepted
  - HD with source shot at 24 but authored at 29.97 will not be accepted
- All source must contain correct field information as verified by Compressor or Dumpster
- Content may be delivered matted: letterbox, pillarbox, or windowbox.
- Stereo audio tracks, uncompressed linear PCM, channels properly mapped as LT and RT or L and R

**Important:** All videos must begin and end with at least one black frame.

### SD Source

SD videos embedded in a book must include posterframe art and meet the following requirements.

- Apple ProRes 422 (HQ)
- VBR expected at 40-60 Mbps
- 720 x 480 and 853 x 480 encoded pixels; for display at either 640 x 360 for 16:9 content or 640 x 480 for 4:3 content
- All encoded content must include pixel aspect ratio (pasp) that defines content as either 4:3 or 16:9.

- Native frame rate of original source:
  - 23.98 progressive frames for content originally authored at 24fps and inverse-telecined
  - 24 progressive frames for content originally authored for at film frame rate in a tapeless workflow
  - 25 progressive frames for content originally authored for PAL delivery
  - 29.97 either interlaced or progressive content originally authored for NTSC delivery
  - Telecined 23.98 content containing a 3:2 pulldown will not be accepted
  - HD with source shot at 24 but authored at 29.97 will not be accepted
- All source must contain correct field information as verified by Compressor or Dumpster
- Content may be delivered matted: letterbox, pillarbox, or windowbox.
- Stereo audio tracks, uncompressed linear PCM, channels properly mapped as LT and RT or L and R

## Encoding Video

Encode videos using Compressor 3.5.2 or later (an application included in Final Cut Studio 3):

1. Select the H.264 for iPod video and iPhone 640X480 setting found in the Apple Devices directory of the Compressor settings list.
2. In the settings inspector on the Encoder pane, select iPod/iPhone (VGA) for native 4X3 content or iPod/iPhone (Anamorphic) for 16X9 recorded in a full 4X3 raster (referred to as “anamorphic”).
3. When using the anamorphic encoder instructions, select the gear across from the Aspect Ratio scroll and select 16:9 (640 x 480). This creates a Pixel Aspect value that appears in the Geometry pane.
4. Use the Geometry pane to enter crop values to remove any black borders or inactive pixels.
5. Click the Preview button on the Compressor Batch window to verify the accuracy of the crop values and also to access the source and setting buttons to test the results of the aspect ratio setting.

---

**Note:** Video files must be H.264 in an MPEG-4 container and the file extension `.m4v` (`.mp4` is allowed, but less preferable).

---

## Video Posterframe

The posterframe is the image that displays inline in the book. It is usually a clip (or frame) from the video. The posterframe should meet the following requirements:

- Approximately 150px tall by 300px wide
- Same aspect ratio as the video
- 300KB recommended maximum file size

## Embedding Audio and Video

All audio and video are embedded in XHTML documents using standard HTML5.

When sending books with embedded audio or video content, keep the following in mind:

- The maximum file size for the .zip file is 2GB. The larger the file is, the longer it will take for a consumer to download your book.
- Keep audio and video content to a minimum to avoid long download times.
- All audio files need to be the same sample rate.
- It may take longer for your book to appear on iBooks because the files need to be checked by Apple for quality control purposes.
- Note that audio and video files do not have DRM applied in the final book, regardless of contract or the metadata; however, DRM is applied to the text in the book.
- Fallback video content is not currently supported. All video must be h.264 in an MPEG-4 container with the file extension .m4v (.mp4 is allowed, but less preferable)

For optimal user experience, audio and video content should appear on its own line in the EPUB and be centered on the page using `text-align:center` defined in the CSS. iBooks provides default dimensions for audio and video. The standard video pane dimensions are a 1:2 ratio and display at 300px wide by 150px tall on the iPad. However, dimensions can also be defined using CSS. iBooks will ensure that video and audio fit on a page, no matter the screen size. If posterframe art and videos are not sized to these proportions, the remaining space within the video pane will be black.

---

**Note:** To enable readers to revisit embedded media, Apple recommends that you provide entries in the Table of Content that point to media.

---

Refer to the EPUB Example 3.0 (`flowing3-0.epub`) that is available in the Examples section of the Deliver Your Content page in iTunes Connect. Embed videos or audio using standard HTML5 tags:

```
video src="video/H264-640x480.m4v" controls="controls"
poster="images/posterimage.jpg" width="320" height="240"/>
```

```
<audio src="audio/loop.m4a" controls="controls" width="200"/>
```

Note that the video tag must include the required poster image. The customer sees the poster image in the text and clicks the image to start the video. The `controls` attribute allows readers to play, pause, and scrub the media; `autoplay` will automatically start playing the audio or video without stopping. The `controls` and

`autoplay` attributes are boolean; including them will activate the attribute no matter their value. For example, `autoplay="false"` is the same as `autoplay="true"`. Note that although `autoplay` is supported, Apple recommends that it not be used. It is best to allow readers to control their own reading experience as they know the environment in which they are reading.

The following two examples show how to embed audio and video content in the EPUB, followed by a screenshot of how the audio and video would appear on the device:

## Audio

`<h3>Below is an embedded audio file.</h3>`

`<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>`

```
<audio src="audio/loop.m4a" controls="controls"/>
```

`<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>`

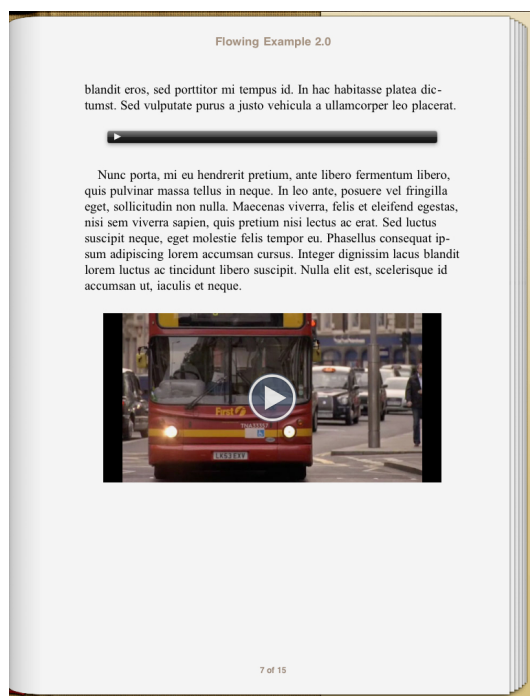
## Video

`<h3>Below is an embedded video file.</h3>`

`<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>`

```
<video src="video/H264-640x480.m4v" controls="controls"
poster="images/posterimage.jpg" />
```

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>



## Links

Since the EPUB format is based on HTML, links can be used to augment and enhance book content. Links can be within a document, as in the case of an index or endnotes, or can link to the web or other external resources such as an author or publisher website.

iBooks also has a very simple linking architecture that enables you to link directly to books within the Store. This could be used for marketing from your own website or online advertisements, or it could be used within an EPUB to direct a customer to other related titles at the end of a book. For more information about linking to the store, see the section "Linking to the iBookstore" in the *Publisher User Guide*.

## Linking Directly to Your Book in iBooks

To create a direct link to your book on iBooks, use the iTunes Link Maker available here: <http://itunes.apple.com/linkmaker/>.

You can also create your own link based on the book's ISBN. There are two ways to construct a URL for your book using its ISBN number.

iBooks-specific link example: `<a`

```
href="http://itunes.apple.com/us/book/isbn9781451648553">Steve Jobs, by Walter Isaacson</a>
```

Generic ISBN link: `<a href="urn:isbn:9781451648553">Steve Jobs, by Walter Isaacson</a>`

See <http://tools.ietf.org/html/rfc3187> for more information on how International Standard Book Numbers (ISBN) can be supported within the URN (Uniform Resource Names) framework.

iBooks will automatically include a page at the end of Flowing books that includes links to other books by the same author, prioritized by most popular.

## Link Styles

By default, links are underlined and blue. You can customize the style of anchor elements (links) using CSS. If you define the color of the anchor element, you must also include the `specified-fonts` layout metadata. See “Defining Book Layout Metadata” (page 57).

## PDFs

PDFs are embedded using a standard anchor element tag that links an image to the PDF file. When the image is tapped, it opens the PDF in a new window where the reader can pan and zoom-in.

---

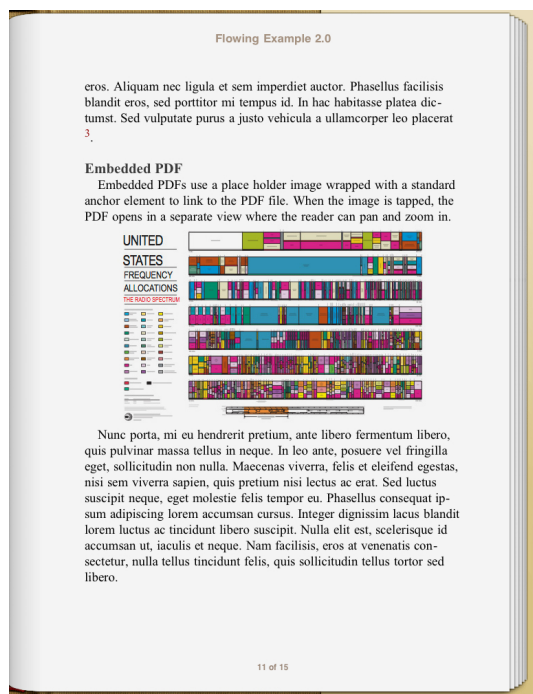
**Note:** When including PDFs, keep in mind the maximum overall book size of 2GB. Larger PDFs may also cause poor performance on some devices.

---

The tag references the PDF file and the image is nested in the anchor element.

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>  
<a href="pdf/sample.pdf"></a>
```





## Adobe Page Template

- The Adobe Page Template XPGT is currently not supported. Use CSS3 instead.

## Books with JavaScript Interactivity

### Overview

Interactive content is supported for Flowing books, Fixed Layout books, and in HTML widgets in Multi-Touch books. This section provides information and tips for creating interactivity using JavaScript. EPUBs with JavaScript interactivity require iBooks 1.5 or later and iOS 5.0 or later; Mac OS X 10.9 (Mavericks) required for desktop versions. Multi-Touch books and iBooks Textbooks require iBooks 2.0 or later and iOS 5.0 or later.

In addition to this guide, an example book that contains interactivity ([fixedlayout3-1.epub](#)) is available for your reference in the Examples section on the Deliver Your Content page in iTunes Connect.

### Best Practices for Developing Scripts for Books with JavaScript Interactivity

iBooks supports JavaScript similarly to Safari. In books, JavaScript allows for features like drag and drop, and touch or mouse events to initiate audio or animation. However, scripts developed for books should take into account the following best practices:

- **Develop scripts that perform well on both Mac OS and iOS devices.** Interactivity on desktop computers requires input from a mouse while interactivity on iOS devices require touch input.
- **When developing interactive books, be mindful of file size and the amount of interactivity on each page.** Interactivity can be a great addition to a book, but good performance must be a priority.
- **Books must not rely on external resources.** Do not use JavaScript to access resources outside of the book. Books must be self-contained.
- **Books must not include databases.** iBooks does not sync the status of AJAX, HTML5 databases, or SQL, and therefore does not save this information. Do not include these resources for interactivity in your books.
- **Use JavaScript for triggering interactivity, but use CSS for animations and transitions.** Animations and transitions perform best when created using CSS rather than JavaScript. Use CSS 3D transforms instead of setting top/bottom/left/right positions— this creates smoother transitions because they are hardware accelerated, but should be reserved for high-priority elements.
- **Avoid using the JavaScript alert feature.** Apple recommends that you not use the JavaScript alert feature, which is frequently used to alert users of errors. If you choose to use this feature, keep in mind while you are crafting these alerts that they will be seen by your readers.

## Designing Content for Books with JavaScript Interactivity

The following sections discuss the technical considerations and design tips to keep in mind when designing content for books with JavaScript interactivity.

### Preventing Default Behavior

By default, iBooks recognizes gestures and clicks to cue menus, pagination, and zoom. However, in books with JavaScript interactivity, iBooks needs to know when to ignore gestures and clicks so that the reader can interact with the interactive portions of your book without prompting iBooks' user interface options to appear. You can disable this default behavior for the interactive element by sending the `preventDefault` message to the event object.

---

**Note:** It is important to keep a significant portion of the page edge unaffected by `preventDefault` to allow the reader to turn the page.

---

For more information on `preventDefault`, see:

[http://developer.apple.com/library/safari/#documentation/appleapplications/reference/safariwebcontent/HandlingEvents/HandlingEvents.html%23//apple\\_ref/doc/uid/TP40006511-SW24](http://developer.apple.com/library/safari/#documentation/appleapplications/reference/safariwebcontent/HandlingEvents/HandlingEvents.html%23//apple_ref/doc/uid/TP40006511-SW24)

## Designing for Both iOS Devices and Mac OS

Design interactive books to work on both iOS devices and Mac OS. Interactivity on desktop computers use input from a mouse while interactivity on iOS devices use touch input. This means that your script must determine if it should provide a touch or mouse event as required by that platform. You can query if a platform supports mouse or touch by using `hasFeature` as defined in the EPUB 3 spec (<http://www.idpf.org/epub/30/spec/epub30-contentdocs.html#app-ers-hasFeature>). For example, if `navigator.epubReadingSystem.hasFeature('touch-events')` returns `true`, then you should provide touch events in your JavaScript.

If your book uses features that are specific to iOS like the accelerometer or multi-finger gestures, you need to include input alternatives, like buttons, that are supported on desktop computers.

Keep in mind that a finger is larger than a mouse, so interactive objects should have a hit area that's large enough for a finger. Because iBooks will scale the page to fit the screen of the device, the size of the hit area is relative to the page size defined in the viewport and may vary for different devices.

When placing interactive objects, avoid placing them close to page edges and text. Objects too close to the edge of the page may cause the reader to unintentionally turn the page. Similarly, objects placed too close to text may cause the reader to unintentionally trigger the re-reading of a word in read aloud content.

## Interaction Cues

Most of the interaction in a book should have a cue so that the reader knows what they can interact with. The following are some ideas for cues:

Most of the interaction in a book should have a cue so that the reader knows what they can interact with. On Mac OS, you can use CSS to style the mouse so that interactive areas are indicated with a pointer (see [http://www.w3schools.com/cssref/pr\\_class\\_cursor.asp](http://www.w3schools.com/cssref/pr_class_cursor.asp)). In iOS, you'll want to use visual cues like the following:

### General interactivity cues:

- Animate or move elements on a page, alerting the reader that elements on the page or in the book can have interaction. In the example below, the title page begins as a blank page, but has text and images that move onto the screen. This example indicates interactivity in the following pages of the book.

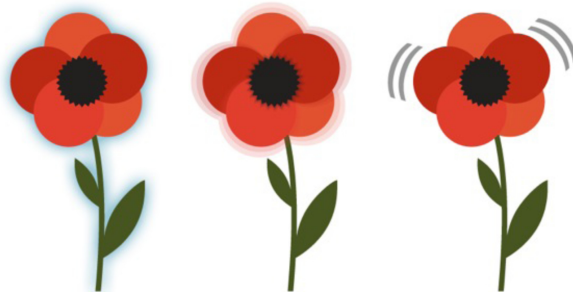


### Touch to trigger interactivity:

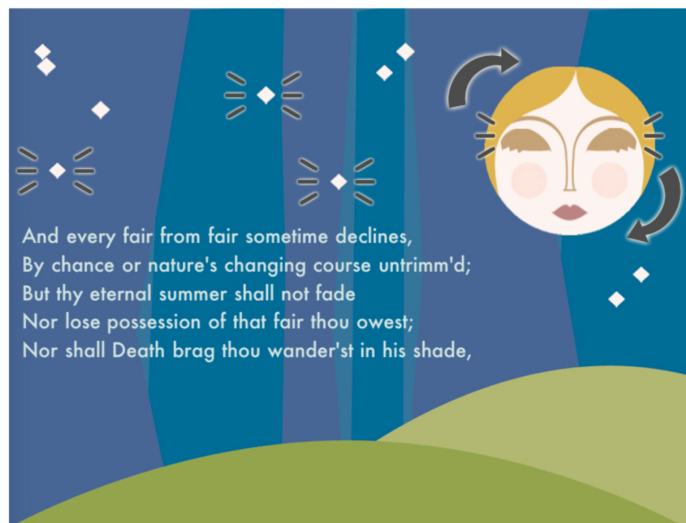
- Use bold formatting or add a glow behind words that trigger sound or animation. In the example below, touching winds triggers a wind noise.

Rough **winds** do shake

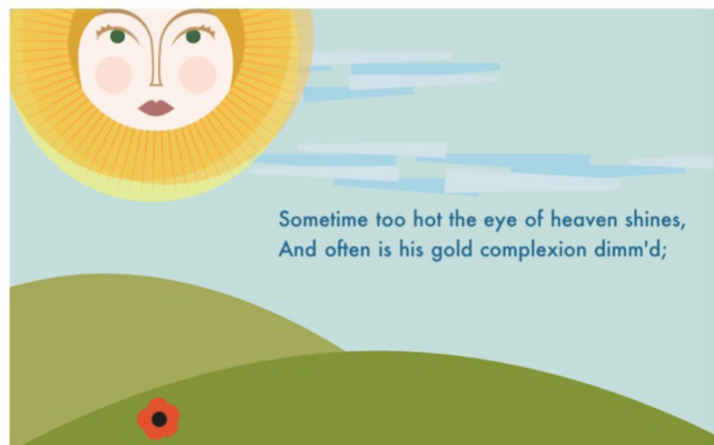
- Add a glow, pulse, or wiggle to an interactive object. In the example below, the flower on the left is glowing, the flower in the middle is pulsing, and the flower on the right is wiggling.



- Have an animatable element perform a behavior that prompts the reader to interact with the other elements. In the example below, the pulsing stars and blinking eyes invite the reader to discover that the stars can be interacted with, and that the moon can be dragged across the screen.



- Create a scene with an out of place element, an unusual blank area, or incomplete image that prompts the reader to investigate the odd element, fill in the blank, or complete the image. In the two examples below, the single red flower invites the reader to interact with the other flower stalks or the empty hillside to complete the flowered landscape.

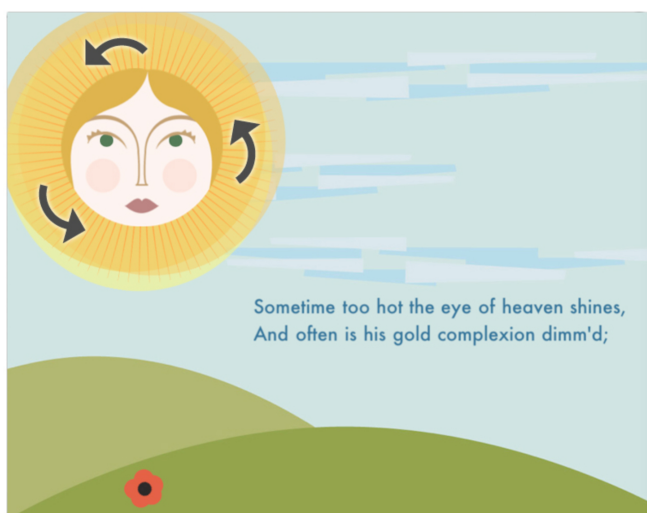


### Drag and drop elements:

- Design elements to feel separate from rest of page, as if they were on a separate layer. When working with drag and drop elements, be sure to keep them behind any text in your book, like in the example below.



- Make the element float, wiggle, or move. In the example below, the rotating halo around the sun invites readers to drag the sun across the screen.



### Example Code

In addition to this guide, a simple example of a book with JavaScript interactivity ([fixedlayout3-1.epub](#)) is available for your reference in the Examples section on the Deliver Your Content page in iTunes Connect. This example contains examples of key interactions: drag and drop, touch to initiate audio, touch to create a new element, and touch to change the state of an element.

## Page Layout

Each page of an interactive, Fixed Layout book is a separate web view, similar to two Safari windows next to each other. Content is isolated on each page with the spine as a firm barrier between the two pages. The body must have dimensions defined in the CSS identical to the aspect ratio of the viewport dimensions.

## Tips for Improving Performance in Books with JavaScript Interactivity

Keep the following in mind to improve performance:

- Image optimization is key for improving performance. Refer to [“Optimizing Images in Fixed Layout Books”](#) (page 61) for methods of optimizing images.
- Limit the number of animations per page; keep only those that are most necessary.
- Use CSS (not JavaScript) for animations.
- Use CSS 3D transforms instead of setting top/bottom/left/right positions. This creates smoother transitions because they are hardware accelerated, but should be reserved for high-priority elements.
- Reduce page complexity.
- Performance is best when JavaScript is tailored to the book. Third-party JavaScript libraries can often be large and may result in poor performance.

## Technical Guidelines and Requirements

All books with JavaScript interactivity designed for iBooks should follow these technical guidelines and requirements:

### Including JavaScript Files In the OPF

JavaScript should be in its own document, not inline in the XHTML. All JavaScript files must be included in the manifest of the OPF and given the mimetype of `application/javascript`.

### Using the iBooks JavaScript Library

The iBooks JavaScript Library (`ibooks.js`) is a collection of pre-written JavaScript that make it easier to develop books with JavaScript interactivity for iBooks. The iBooks JS Library provides functionality to:

- defer an event
- make an element draggable
- make an element stamp-able
- make an element toggle-able



- add audio
- define constants

To begin using `ibooks.js`, first include the script in your markup.

```
<head>
  <meta name="viewport" content="width=575, height=432"/>
  <meta content="text/html; charset=UTF-8"/>
  <title>Fixed Layout Example 2.3</title>
  <link href="css/stylesheet.css" type="text/css" rel="stylesheet"/>
  <link rel="stylesheet" href="css/page06.css" type="text/css" media="screen"
title="no title" charset="utf-8"/>
  <script src="js/ibooks.js" type="text/javascript" charset="utf-8"></script>
</head>
```

When a page has loaded, `ibooks.js` appends a CSS class build-in to the body. The addition of this class can be used to trigger animations.

## Deferred Events

Sometimes, a developer wants to fire a delayed event after the content has loaded. By adding `ibooks-deferred-event` to an element, `ibooks.js` will append a CSS class `active` to the target element after a delay, by default this value in milliseconds is 1000. This delay can be defined on a per element basis by adding a HTML attribute `data-deferred-event-delay` and setting the desired value in milliseconds.

```
<!-- "active" class appended after default delay, 1000ms -->
<div class="ibooks-deferred-event"></div>
```

```
<!-- "active" class appended 5000ms after content load -->
<div class="ibooks-deferred-event" data-deferred-event-delay="5000"></div>
```

## Draggable Elements

Draggable elements respond to touch events and can be moved around the page. HTML elements can be made draggable by simply appending the CSS class `ibooks-draggable`.

```
<!-- Makes the target element draggable -->
<div class="ibooks-draggable"></div>
```

## Stampable Elements

Any element with a CSS class `ibooks-stampable` will act as a parent container for stampable elements. Stampable elements respond to touch gestures; on touch, `ibooks.js` will append an empty `<div>` element with the CSS class `stamp` to the parent container. You can then style the appended element.

`ibooks.js` relies on a SVG path to define a valid touch area; without a SVG path `ibooks.js` will not respond to touch events. It is worth noting that SVG patches are especially useful when defining irregular shaped hit areas, such as hills.

```
<div class="ibooks-stampable">
  <!--svg defines irregularly-shaped hit area for replicating script-->

  <svg version="1.1" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
    width="1224px" height="792px" viewBox="0 0 1224 792">
    <path fill="#82983E"
    d="M612.806,387.128c-8.555,0-17.023,0.229-25.394,0.656c-50.48-31.565-116.654-60.656-189.107-60.656
    c-8.653,0-16.846,0.42-24.805,1.205v136.949l14.525,0.023l296.721,0.478l215.824,0.347
    C900.57,466.128,771.732,387.128,612.806,387.128z"/>
  </svg>
</div>
```

## Toggle-able Elements

`ibooks.js` provides the ability to make elements toggle-able, by appending the CSS class `ibooks-toggleable`, on touch elements will toggle the CSS class `active`.

```
<!-- Inactive, toggle-able element -->
<div class="ibooks-toggleable"></div>

<!-- Active, toggle-able element -->
<div class="ibooks-toggleable active"></div>
```

## Audio

Any HTML element can be used to trigger audio by appending the CSS class `ibooks-media-audio` and defining a value for the HTML attribute `data-ibooks-audio-src`. Additionally, the audio source playhead can be reset each time the element is touched by defining `data-ibooks-audio-reset-on-play`. Read aloud can be paused after the media has played or is paused by setting an HTML attribute:

`"ibooks:pause-readaloud"` to `"true"`. See ["Controlling Narration with Read Aloud Books"](#) (page 71) for more information.

---

**Note:** Only one audio source can be played at a time.

---

```
<!-- Toggles between play, pause on touch -->
<div class="ibooks-media-audio" data-ibooks-audio-src="audio/source.m4a"></div>

<!-- Plays source on touch, resets playback position to start on subsequent touches -->
<div class="ibooks-media-audio" data-ibooks-audio-reset-on-play="true"
data-ibooks-audio-src="audio/source.m4a"></div>
```

## Defining Constants

If desired, many of the `ibooks.js` constants can be defined by the developer; these variables are located within `initConfigurables`.

```
/**
 * Configuration of user defined constants.
 */
iBooksBaseController.prototype.initConfigurables = function() {
```

```
// CSS class name on active elements
iBooks.ACTIVE_CSS_CLASS = "active";

// CSS class name appended to body on page load
iBooks.CSS_CLASS_ON_LOAD = "build-in";

// Delay in milliseconds before deferred events fire
iBooks.DEFERRED_EVENT_DELAY = "1000";

// CSS selector for page
iBooks.PAGE_CSS_SELECTOR = ".page";

// CSS class for stamped elements
iBooks.STAMPED_ELEMENT_CSS_CLASS = "stamp";
};
```

## Testing During Development

During the development of your EPUBs, use the Book Proofer tool to rapidly proof books without syncing through iTunes. Changes made to your EPUB are automatically synced to your iPad, iPhone, or iPod touch. The Book Proofer tool and documentation are available on iTunes Connect.

## Book Validation

All books must adhere to EPUB standards. To ensure quality content, your book must pass validation at the time of import. There are a few issues commonly encountered with books that prevent them from importing into the iBooks system. Review all books carefully to ensure that the following are addressed prior to delivery:

- All books must pass Transporter or iTunes Producer validation.
- All characters in the URIs must be valid with non-alphanumeric characters properly encoded (e.g., space should be encoded as '%20'). This problem most often surfaces in URIs in the NCX files and is due to spaces included in filenames.
- All files included in EPUBs must be listed in the book manifest (OPF file). Books containing unmanifested files will fail import, as these files are by definition not intentionally included.

- Only UTF-8 and UTF-16 encodings are allowed in books. Any books that are improperly encoded will fail import, so it is important to use the correct character encodings.
- The maximum book size is constrained by the ZIP standard, and is currently 2 GB.

If you use custom attributes, they must be HTML5 data attributes. A data attribute is an attribute that exists outside of a namespace and starts with the string `data-`.

For more information on HTML5 data attributes, see:

[http://www.w3.org/TR/html5/dom.html#embedding-custom-non-visible-data-with-the-data-\\*-attributes](http://www.w3.org/TR/html5/dom.html#embedding-custom-non-visible-data-with-the-data-*-attributes)

Example:

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/2007/ops">

  <head>...</head>
  <body>
    ...
    <p class="text" data-name="value">text</p>
    ...
  </body>
</html>
```

## Book Samples

### About Book Samples

iBooks automatically creates samples for all books (except read aloud books). What the sample is made up of depends on the type of book.

If you do not want iBooks to cut the sample automatically, you can create a separate, custom `.epub` file and deliver it for use as the sample on iBooks. Note that not all tools support this option. If you contract with a third party to manage your content delivery, they should be able to do this for you. Contact your delivery company directly for more details.

---

**Note:** A purchased book replaces its sample in the customer's library. Be sure to include all sample content in the full book.

---

## Samples for Flowing Books

If a book is predominantly text, the sample is based on a percentage of the number of words (the percentage used is specified in your contract). Where the sample begins can vary depending on how the book has been structured. The words on which the percentage is calculated are counted from the beginning of the book. For flowing books, the beginning is determined by finding the first landmark where the `epub:type` attribute is "bodymatter". If that attribute is specified, iBooks starts cutting the sample from the publication component in the `spine` item referenced by the `href` attribute in the landmark. Then, the sample is cut either from the first item in the Landmarks `<nav>` block, or the percentage of the words, whichever is greater. For example, if the first item in the Landmarks `<nav>` block contains only 2% of the total word count and the percentage specified in the contract is 5%, then the remaining 3% is taken from the next item in the Landmarks `<nav>` block. Any images, movies, or audio included in that percentage will be included in the sample.

If the "bodymatter" attribute is not specified in the Landmarks structure, iBooks looks for the first landmark where the `epub:type` attribute is *not* one of the following: "cover", "frontmatter", "toc", "foreword", "introduction", or "dedication" and starts with the `spine` items immediately following the component referenced in the prior "cover", "frontmatter", "toc", "foreword", "introduction", or "dedication" landmark.

---

**Note:** It is recommended that the publication component referenced by the `href` attribute in the "bodymatter" landmark *not* be listed close to the last item on the list of `spine` items. When cutting the sample based on percentage of words, iBooks could potentially include publication components that precede the referenced `spine` item, which could result in iBooks including the entire book in the sample.

---

## Samples for Fixed Layout Books

If your book is a picture book or other Fixed Layout book, you must indicate `pre-paginated` in the `<meta property="rendition:layout">` tag in the `<metadata>` element within the OPF file. (See [“Defining Book Layout Metadata”](#) (page 57) for instructions.) The `pre-paginated` option determines how the sample is cut. Without this option, the sample is cut as if the content is a text-heavy book like a novel. With this option, the sample is cut based on a percentage of the number of pages rather than word count. iBooks determines the first page of the book (each page of a Fixed Layout Book, whether a single page or a two-page spread, must be a separate XHTML file) and then cuts the sample from a percentage of total page count. The sample will

look to the Landmarks `<nav>` block of the `.opf` to determine the start of the content of the book. For optimal samples, include `epub:type="bodymatter"` to indicate the beginning of the book's content. Note that a Landmarks `nav` is required for Fixed Layout books only if you do not provide a custom sample.

## Samples for Books with Interactive Content

To provide an optimal quality sample for books with interactive content, we recommend that you supply your own custom sample due to the dynamic, complex nature of an interactive book. However, a custom sample is optional for books with interactive content. To create a custom sample, create a separate `.epub` file that includes the pages you want in the sample and deliver it within the `<assets>` block with `type="preview"`.

---

**Note:** Automatically-generated samples are not supported for books with JavaScript interactivity if they are also read aloud.

---

## Samples for Read Aloud Books

If your book is a read aloud book, you must indicate `pre-paginated` in the `<meta property="rendition:layout">` tag in the `<metadata>` element within the OPF file. (See [“Defining Book Layout Metadata”](#) (page 57) for instructions.) Read aloud content requires a custom sample. If you do not provide a custom sample, the book won't be accepted to the iBooks Store. To create a custom sample, create a separate `.epub` file that includes the pages you want in the sample and deliver it within the `<assets>` block with `type="preview"`.

## Samples for Multi-Touch Books

Multi-Touch books require a sample file to be offered on iBooks and iBooks Author creates one for you when you publish your book. During the publishing process iBooks Author creates a package with the extension `.itmsp` and saves it to the folder that contains the your `.ibooks` file, cover art, and sample file. Then at the end of that export process, iTunes Producer opens the package for you to add your metadata and publish your book to iBooks.

# Guidelines for Flowing Books

## Overview

This chapter describes guidelines for creating Flowing Books. Refer to [“Digital Book Essentials”](#) (page 9) for best practices that apply to all types of books.

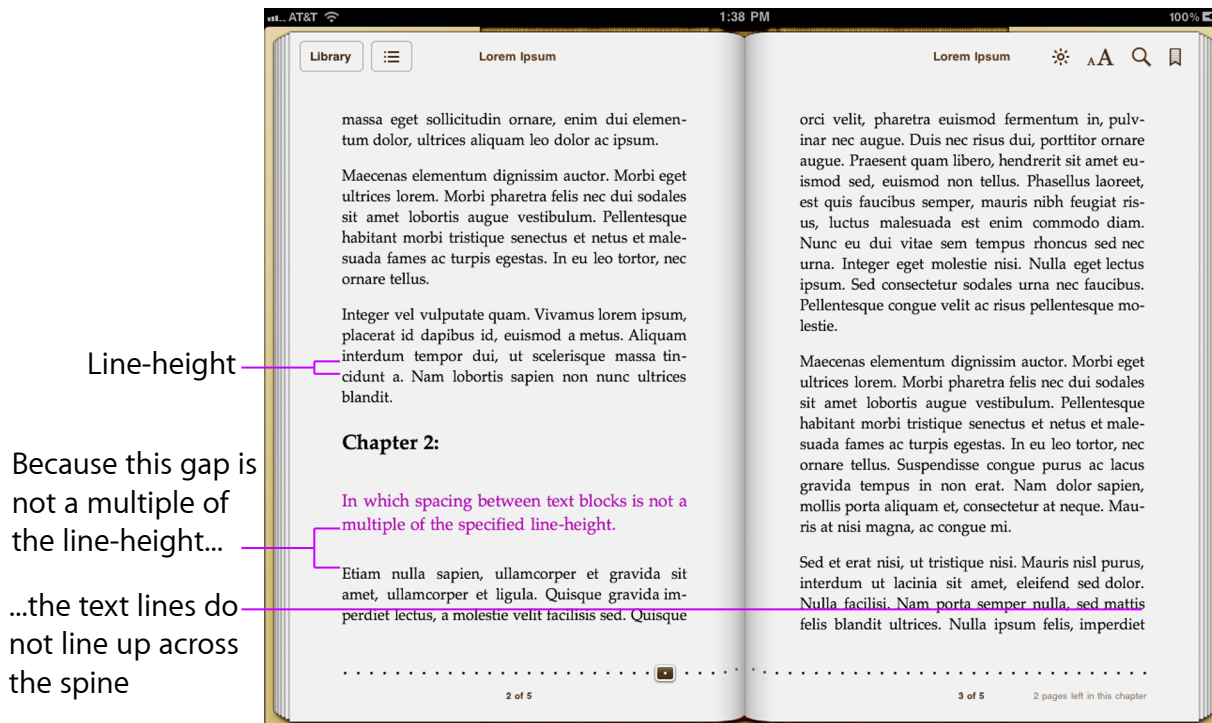
## Content Structure

- In Flowing books, supplying Landmarks in the Navigation document is recommended but not required.
- In Flowing books, divide each chapter into its own XHTML document. iBooks creates a page break between documents.
- Separating chapters into documents improves performance in iBooks.
- Use header tags (for example, `<h1>`, `<h2>`) for headers. Do not create a header using `<p>` styled to look like a header. iBooks relies on HTML semantics when laying out the text of a book. Using a `<p>` instead of a header results in poor text layout such as unexpected hyphenation.

## Alignment

If specifying line-height, extra spacing between text blocks and any padding around images should be a multiple of the specified line-height to keep text aligned across the spine.





## Nonlinear Content

Nonlinear content includes documents that supplement the main content, but are outside the reading flow of the book. For example, nonlinear content can be charts, tables, review answers, and so on. You provide the link on the page that points to a nonlinear document (for example, `<a href="answerkey.xhtml">See Answer</a>`). That link is a standard anchor element and can be associated with both text or an image.

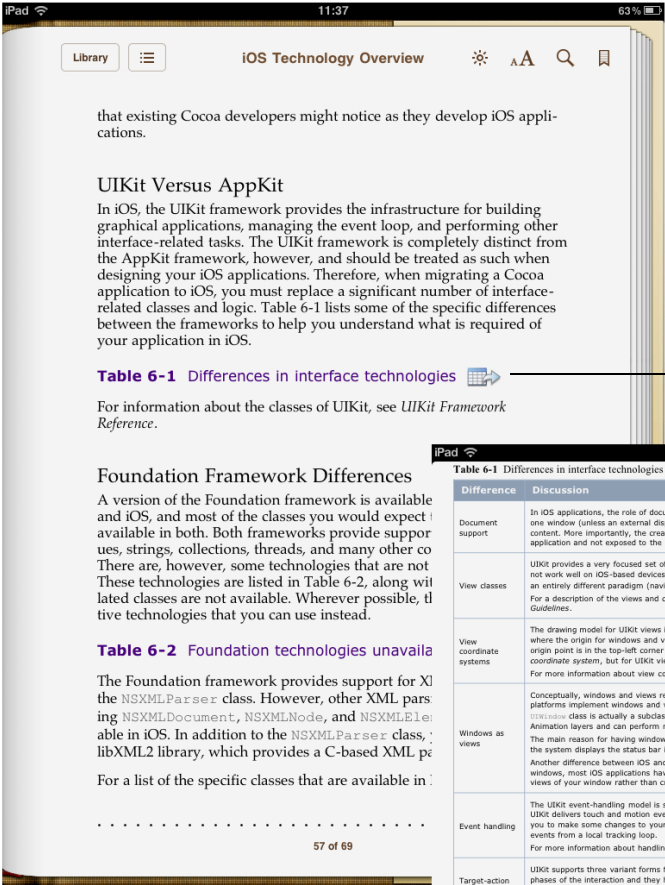
---

**Note:** Nonlinear documents are not supported for Fixed Layout books.

---

To indicate if a spine item is nonlinear, specify a value of `no` for the optional `linear` attribute. Note that the `linear` attribute is not required on spine items. All items without a `linear` attribute will default to `linear="yes"`.

When a reader selects a link that leads to a nonlinear document, the content opens in a separate window that overlays the book. Because the window is outside the flow of the book, it maintains its layout and formatting. The content is searchable from within the book and can be panned and zoomed in on. The following graphic shows an example of a nonlinear document.



Clicking this link opens  
the non-linear content  
in its own window

### UIKit Versus AppKit

In iOS, the UIKit framework provides the infrastructure for building graphical applications, managing the event loop, and performing other interface-related tasks. The UIKit framework is completely distinct from the AppKit framework, however, and should be treated as such when designing your iOS applications. Therefore, when migrating a Cocoa application to iOS, you must replace a significant number of interface-related classes and logic. Table 6-1 lists some of the specific differences between the frameworks to help you understand what is required of your application in iOS.

**Table 6-1** Differences in interface technologies

For information about the classes of UIKit, see *UIKit Framework Reference*.

### Foundation Framework Differences

A version of the Foundation framework is available on both iOS and Mac OS X, and most of the classes you would expect to find are available in both. Both frameworks provide support for strings, collections, threads, and many other common tasks. There are, however, some technologies that are not available in iOS. These technologies are listed in Table 6-2, along with the Foundation Framework classes that are available in iOS. Wherever possible, the Foundation Framework provides alternative technologies that you can use instead.

**Table 6-2** Foundation technologies unavailable in iOS

The Foundation framework provides support for XML parsing through the `NSXMLParser` class. However, other XML parsing technologies are not available in iOS. In addition to the `NSXMLParser` class, the `libXML2` library, which provides a C-based XML parser, is also available in iOS. For a list of the specific classes that are available in iOS, see the *Foundation Framework Reference*.

Table 6-1 Differences in interface technologies	
Difference	Discussion
Document support	In iOS applications, the role of documents is deemphasized in favor of a simpler content model. Because applications typically have only one window (unless an external display is connected), the main window acts as the sole environment for creating and editing all application content. More importantly, the creation and management of any actual document-related files is handled behind the scenes by the application and not exposed to the user.
View classes	UIKit provides a very focused set of custom views and controls for you to use. Many of the views and controls found in AppKit would simply not work well on iOS-based devices. Other views have more iOS-specific alternatives. For example, instead of the <code>NSOutlineView</code> class, iOS uses an entirely different paradigm (navigation controllers) to manage the display of hierarchical information. For a description of the views and controls available in iOS, along with information on how to use them, see <i>iOS Human Interface Guidelines</i> .
View coordinate systems	The drawing model for UIKit views is nearly identical to the model in AppKit, with one exception. AppKit views use a coordinate system where the origin for windows and views is in the lower-left corner by default, with axes extending up and to the right. In UIKit, the default origin point is in the top-left corner and the axes extend down and to the right. In AppKit, this coordinate system is known as a <i>modified</i> coordinate system, but for UIKit views it is the default coordinate system. For more information about view coordinate systems, see <i>View Programming Guide for iOS</i> .
Windows as views	Conceptually, windows and views represent the same constructs in UIKit as they do in AppKit. In implementation terms, however, the two platforms implement windows and views quite differently. In AppKit, the <code>NSWindow</code> class is a subclass of <code>NSView</code> , but in UIKit, the <code>UINavigationController</code> class is actually a subclass of <code>UIViewController</code> instead. This change in inheritance means that windows in UIKit are backed by Core Animation layers and can perform most of the same tasks that views do. The main reason for having window objects at all in UIKit is to support the layering of windows within the operating system. For example, the system displays the status bar in a separate window that floats above your application's window. Another difference between iOS and Mac OS X relates to the use of windows. Whereas a Mac OS X application can have any number of windows, most iOS applications have only one. When you want to change the content displayed by your application, you swap out the views of your window rather than create a new window.
Event handling	The UIKit event-handling model is significantly different from the one found in AppKit. Instead of delivering mouse and keyboard events, UIKit delivers touch and motion events to your views. These events require you to implement a different set of methods but also require you to make some changes to your overall event-handling code. For example, you would never track a touch event by extracting queued events from a local tracking loop. For more information about handling events in iOS applications, see <i>Event Handling Guide for iOS</i> .
Target-action model	UIKit supports three variant forms for action methods, as opposed to just one for AppKit. Controls in UIKit can invoke actions for different phases of the interaction and they have more than one target assigned to the same interaction. Thus, in UIKit a control can deliver multiple distinct actions to multiple targets over the course of a single interaction cycle. For more information about the target-action model in iOS applications, see <i>Event Handling Guide for iOS</i> .
Drawing and printing support	The drawing capabilities of UIKit are scaled to support the rendering needs of the UIKit classes. This support includes image loading and display, string display, color management, font management, and a handful of functions for rendering rectangles and getting the graphics context. UIKit does not include a general-purpose set of drawing classes because several other alternatives (namely, Quartz and OpenGL ES) are already present in iOS. In iOS 4.2 and later, applications can use the UIKit printing support to deliver data wirelessly to a nearby printer. For more information about graphics and drawing, see <i>Drawing and Printing Guide for iOS</i> .
Text support	The primary text support in iOS is geared toward composing email and notes. The UIKit classes let applications display and edit simple strings and somewhat more complex HTML content. In iOS 3.2 and later, more sophisticated text handling capabilities are provided through the Core Text and UIKit frameworks. You can use these frameworks to implement sophisticated text editing and presentation views and to support custom input methods for those views. For more information about text support, see <i>Text, Web, and Editing Programming Guide for iOS</i> .
The use of accessor methods versus properties	UIKit makes extensive use of properties throughout its class declarations. Properties were introduced to Mac OS X in version 10.5 and thus came along after the creation of many classes in the AppKit framework. Rather than simply mimic the same getter and setter methods in AppKit, properties are used in UIKit as a way to simplify the class interfaces. For information about how to use properties, see "Declared Properties" in <i>The Objective-C Programming Language</i> .
Controls and cells	Controls in UIKit do not use cells. Cells are used in AppKit as a lightweight alternative to views. Because views in UIKit are themselves very lightweight objects, cells are not needed. Despite the naming conventions, the cells designed for use with the <code>UITableView</code> class are actually based on the <code>UITableViewCell</code> class.
Table views	The <code>UITableView</code> class in UIKit can be thought of as a cross between the <code>NSOutlineView</code> and <code>NSCollectionView</code> classes in the AppKit framework. It uses features from both of those AppKit classes to create a more appropriate tool for displaying data on a smaller screen. The <code>UITableView</code> class displays a single column at a time and allows you to group related rows together into sections. It is also a means for displaying and editing hierarchical lists of information. For more information about creating and using table views, see <i>Table View Programming Guide for iOS</i> .
Menus	Nearly all applications written for iOS have a much smaller command set than do comparable Mac OS X applications. For this reason, menu bars are not supported in iOS and are generally unnecessary anyway. For those few commands that are needed, a toolbar or set of buttons is usually more appropriate. For data-based menus, a picker or navigation controller interface is often more appropriate. For context-sensitive commands in iOS, you can display those on the edit menu in addition to (or in lieu of) commands such as Cut, Copy, and Paste.
Core Animation layers	In iOS, every drawing surface is backed by a Core Animation layer and implicit animation support is provided for many view-related properties. Because of the built-in animation support, you usually do not need to use Core Animation layers explicitly in your code. Most animations can be performed simply (and more directly) by changing a property of the affected view. The only time you might need to use layers directly is when you need precise control over the layer tree or when you need features not exposed at the view level.

The example Flowing Book EPUB file (`flowing3-0.epub`) located in the Examples section of the Deliver Your Content page in iTunes Connect shows an example that links to a nonlinear document.

---

**Note:** If the `<spine>` references a PDF, the `linear` attribute must be set to `no`. If the `linear` attribute is set to `yes`, delivery will fail.

---

## Scrolling in Flowing Books

iBooks has several viewing themes that the reader can select for Flowing books. These include Book, Full Screen, and Scroll. The Scroll theme scrolls vertically for books with horizontal text, and scrolls horizontally for books with vertical text. This ensures an unbroken/unpaginated text flow.

By default, Japanese and Chinese books scroll horizontally, while all other languages scroll vertically. To redefine the scroll direction, the book must include the following metadata in the `.opf` file: `"ibooks:scroll-axis"`. Possible values are `vertical`, `horizontal`, and `default`. For example:

```
<metadata>
. . .
<meta property="ibooks:scroll-axis">vertical</meta>
. . .
</metadata>
```

This example is appropriate for a book with horizontal Japanese or Chinese text. When rendered with iBooks' Scroll theme, horizontal text is most easily read scrolling top to bottom on the vertical axis.

## Tables

iBooks automatically recognizes tables. In Flowing books, iBooks sizes large tables to fit within the width of the page. When a reader double taps the table, the table opens in a new web view that overlays the book. In this web view, the reader can pan and zoom in on the table.

## Fonts

- Font sizes should be defined in `em` or `px`, not by name, for example, `small` and `large`.
- The main text of a book should either not have a defined `font-size` or should have a `font-size` of `1em`. This will ensure ideal readability and font scaling.

# Guidelines for Fixed Layout Books

## Overview

This chapter describes guidelines for creating Fixed Layout books. Refer to [“Digital Book Essentials”](#) (page 9) for best practices that apply to all types of books. Fixed Layout books lend themselves to interactive content, so be sure to check out [“Books with JavaScript Interactivity”](#) (page 37) in Chapter 2. Note that nonlinear documents are not supported for Fixed Layout books.

Fixed Layout books are used to create highly-designed books like children's picture books, cookbooks, and art books. Fixed Layout books support text overlapping images and full-bleed images—two features not possible with Flowing books. These features are made possible with the support of CSS positioning. This chapter provides guidelines to follow when creating a Fixed Layout book. Not following these guidelines could adversely affect the way the book is displayed.

In addition to this guide, an example Fixed Layout book (`fixedlayout3-1.epub`) is available for your reference in the Examples section on the Deliver Your Content page in iTunes Connect.

---

**Note:** Fixed Layout is a global definition that affects an entire book. Fixed Layout cannot be used on a per-page basis.

---

## Setting Up the Document

Fixed Layout books are like Flowing books, but with the following additions:

- iBooks 3.0 or later supports EPUB 3 Fixed Layout books. Using EPUB 3, you have the choice of creating a single spread using one or two content documents. By default, a book with two documents per spread maintains the book-like appearance including the book spine, pages, and page curl. By default, a book with a single document per spread does not include the book-like appearance and instead the spread looks like a single sheet, similar to the PDF view. If you create a Fixed Layout book that is two documents per spread, but you do not want it to have the book-like appearance, you can hide the book binding by including the `binding` layout property and giving it a value of `false`. See [“Defining Book Layout Metadata”](#) (page 57).

- In Fixed Layout books that have two documents per spread, the first page of a left-to-right paginated book (for example, English) will be on the right side of the spread. In a right-to-left paginated book (for example, Japanese), the first page of content will be on the left page of the spread. Generally, this first page is the cover page.
- A Fixed Layout book must have the height and width of the book defined in a `<meta>` tag in the head of each XHTML document. For example, `<meta name="viewport" content="width=600, height=1000" />`. iBooks uses these dimensions to determine the aspect ratio of the book. These dimensions must be consistent and *must* match the dimensions of the `<body>` tag.

## EPUB 3 Fixed Layout

In addition to the EPUB 3 requirements defined in [“Structure of a Version 3 EPUB”](#) (page 9) and [“EPUB 3 Features”](#) (page 15), the guidelines in this section are specific to Fixed Layout books.

- A Fixed Layout book must have pre-paginated in the `<meta property="rendition:layout">` tag in the `<metadata>` element within the OPF file. See [“Defining Book Layout Metadata”](#) (page 57) below.
- The Landmarks navigation structure is required for Fixed Layout books only if you do not provide a custom sample. iBooks references the Landmarks when cutting the sample for the iBooks Store. It allows iBooks to determine where the main content of the book starts (see [“Landmarks”](#) (page 12)).

## Defining Book Layout Metadata

EPUB 3 Fixed Layout books use the `<metadata>` structure to define a book’s layout. In EPUB 2, this metadata was defined in the Apple Display Options file, but in EPUB 3, the metadata is defined in the `<metadata>` element within the OPF file. For example:

```
<metadata xmlns="http://www.idpf.org/2007/opf"
xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <dc:title>Fixed Layout Book</dc:title>
  <dc:identifier id="bookid">0123456789</dc:identifier>
  <dc:language>en</dc:language>
  <meta property="dcterms:modified">2012-08-15T00:00:00Z</meta>
  <meta property="ibooks:version">3.0</meta>
  <meta property="rendition:layout">pre-paginated</meta>
  <meta property="rendition:spread">none</meta>
  <meta property="rendition:orientation">auto</meta>
  <meta property="ibooks:binding">>false</meta>
</metadata>
```

The following table explains the available layout property attributes:

`rendition:layout`

Example:

```
<meta property="rendition:layout">pre-paginated</meta>
```

Defines whether a book is a flowing book or a Fixed Layout book. Supported values are `reflowable` (standard Flowing book) and `pre-paginated` (Fixed Layout book). This is equivalent to `fixed-layout : true|false` in EPUB 2.

`rendition:spread`

Examples designating two content documents per spread:

```
<meta property="rendition:spread">auto</meta> or <meta  
property="rendition:spread">both</meta>
```

Example designating a single content document per spread:

```
<meta property="rendition:spread">none</meta>
```

Defines if there are one or two content documents per spread. Supported values are `auto`, `both`, and `none`.

`ibooks:specified-fonts`

Example:

```
<meta property="ibooks:specified-fonts">true</meta>
```

Must be defined if your book contains embedded fonts. This is equivalent to `specified-fonts : true|false` in EPUB 2.

`rendition:orientation`

Portrait only example:

```
<meta property="rendition:orientation">portrait</meta>
```

Landscape only example:

```
<meta property="rendition:orientation">landscape</meta>
```

Orientation not locked:

```
<meta property="rendition:orientation">auto</meta>
```

Defines the global (applies to all platforms) orientation to use when the book is opened. Allowed values: `landscape`, `portrait`, or `none`. If not specified, `none` is the default and the orientation will not be locked; the user can rotate the device to view in both landscape and portrait orientations. `rendition:orientation` is equivalent to `orientation-lock` in EPUB 2.

`ibooks:ipad-orientation-lock`

Portrait only example:

```
<meta property="ibooks:ipad-orientation-lock">portrait-only</meta>
```

Landscape only example:

```
<meta property="ibooks:ipad-orientation-lock">landscape-only</meta>
```

Defines the the orientation to use when the book is opened on iPad. Allowed values: `landscape-only` or `portrait-only`. If not specified, the orientation will use the global orientation. If no orientation metadata is specified, the orientation will not be locked and the user can rotate the device to view in both landscape and portrait orientations.

`ibooks:ipad-orientation-lock` is equivalent to `orientation-lock` in EPUB 2.

`ibooks:iphone-orientation-lock`

Portrait only example:

```
<meta property="ibooks:iphone-orientation-lock">portrait-only</meta>
```

Landscape only example:

```
<meta property="ibooks:iphone-orientation-lock">landscape-only</meta>
```

Defines the the orientation to use when the book is opened on iPhone or iPod touch. Allowed values: `landscape-only` or `portrait-only`. If no orientation metadata is specified, the orientation will not be locked and the user can rotate the device to view in both landscape and portrait orientations.

`ibooks:iphone-orientation-lock` is equivalent to `orientation-lock` in EPUB 2.

`ibooks:binding`

Example:

```
<meta property="ibooks:binding">false</meta>
```

Defines if the book binding is visible in a two-document-per-spread Fixed Layout book. Supported values are `true` or `false`. By default, the value is `true`.

## Table of Contents

Fixed Layout books include a thumbnail table of contents allowing for easy visual navigation. This visual table of contents is the default for Fixed Layout books. A traditional list table of contents is also created using the HTML5 `<nav>` element for EPUB 3 books. The list table of contents is very helpful with large books with distinct chapters or sections such as cookbooks. However, the list table of contents is not very helpful with a short picture book and may be suppressed for those books. To suppress the list table of contents for an EPUB 3 book, include only one item in the `<nav epub:type="toc">` block in the Navigation Document—the thumbnail table of contents will still be available. The following shows a thumbnail table of contents:





In Fixed Layout books, `epub:type="page-list"` for EPUB 3 can be used to define the page numbers that iBooks assigns to each page. For example, the first page of the book is page 1 by default, but in some books, this may be the cover page and shouldn't be numbered. You can use `pageList` or `epub:type="page-list"` to define an empty string to that page so that it is not numbered. Similarly, you can define i, ii, iii or a, b, c then 1, 2, 3 for page numbers. Using `<pageList>` or `epub:type="page-list"` in Fixed Layout books requires iBooks 2.0 or higher.

The following shows an EPUB 3 example of `epub:type="page-list"` within an `<nav>` block:

```
<?xml version="1.0" encoding="utf-8"?>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:epub="http://www.idpf.org/2007/ops"

epub:prefix="ibooks: http://vocabulary.itunes.apple.com/rdf/ibooks/vocabulary-extensions-1.0">
. . .
<nav epub:type="page-list">
  <ol>
    <li><a href="page001.xhtml"> </a></li>
    <li><a href="page002.xhtml">i</a></li>
    <li><a href="page003.xhtml">ii</a></li>
    <li><a href="page004.xhtml">iii</a></li>
    <li><a href="page005.xhtml">iv</a></li>
    <li><a href="page006.xhtml">v</a></li>
    <li><a href="page007.xhtml">vi</a></li>
```



```
<li><a href="page008.xhtml">vii</a></li>
</ol>
</nav>
. . .
</body>
</html>
```

---

**Note:** In the `prefix` example above, the space between `ibooks:` and `http://` is important; it needs to be a single, regular space character and not a return, non-breaking space, or any other kind of whitespace.

---

## Using Fonts in Fixed Layout Books

Fonts are defined in the `opf` and CSS just as in Flowing Books. In addition to the font recommendations in “[Fonts](#)” (page 29), the following apply to working with fonts in Fixed Layout books:

- Use pixels instead of ems for the font-size.
- Text placement should be specified in pixels.
- Do not use fractional measurements for letter spacing, font size, and text placement. An example of a fractional measurement is 5.255px.

## Optimizing Images in Fixed Layout Books

In addition to the image recommendations in “[Interior Image Requirements](#)” (page 24), the following apply to working with images in Fixed Layout books:

- Images inside Fixed Layout books must be in JPEG or PNG format. To reduce file size, use JPEG for any images that do not require transparency. Using JPEGs results in a smaller file size and better performance. We recommend using a quality of 85 for JPEGs.
- Images inside Fixed Layout books must not exceed 3.2 million pixels. You can calculate whether an image inside the book file exceeds 3.2 million pixels by multiplying the height and width of the image.

## Layout

- Because iBooks scales the book to fit within the viewport, both landscape-dominant and portrait-dominant books will maintain their intended layout in either orientation of the device, even after an orientation change.
- Readers can choose to read the book at the zoom-level that is most comfortable for them.
- If a book has text that goes sideways across a page, a reader may need to rotate the device to better read the book. In that case, the book's orientation can be locked. See [“Defining Book Layout Metadata”](#) (page 57).
- When developing Fixed Layout books, you may notice that iBooks sometimes remembers the appearance of your book even after it has been deleted and replaced with an updated version. This is due to caching. If you anticipate making a number of revisions to your books while editing, Apple recommends that you use the Book Proofer app (which can be downloaded from the Deliver Your Content page in iTunes Connect). Another option is to modify metadata in the Package Document (.opf file) to iterate the date-modified metadata. For example:

```
<metadata xmlns="http://www.idpf.org/2007/opf"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>Fixed Layout Example 3</dc:title>
  <dc:identifier id="bookid">0123456789</dc:identifier>
  <dc:language>en</dc:language>
  <meta property="dcterms:modified">2012-08-15T00:00:00Z</meta>
</metadata>
```

## Text

“Wavy” and other strongly-formatted text can be created using SVG text paths. We recommend that SVG should be used sparingly.



---

**Note:** SVG text selection is best when each letter is in its own `<tspan>`.

---

## Read Aloud Books

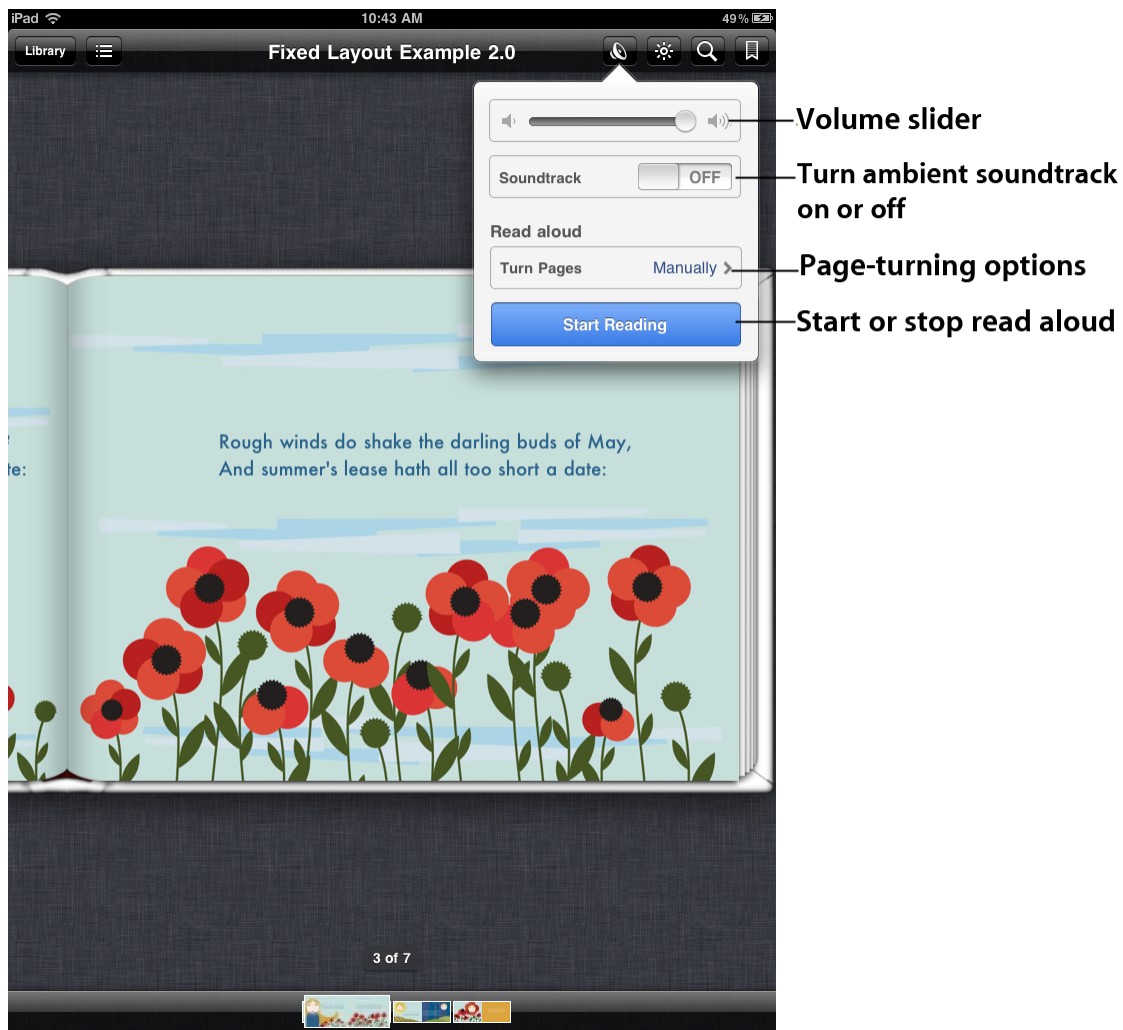
### Overview

In Read Aloud books, an audio file using a narrator’s voice can read the text on a page and the words can be highlighted as they are spoken. The reader can choose to have the pages turn automatically to keep the session flowing, or set them to turn manually. A tap on the screen displays controls in the upper-right corner of the book to customize the reading and listening experience. Note that read aloud content is supported only in Fixed Layout Books. iBooks syncs text and audio using Media Overlays, an EPUB-specific subset of Synchronized Multimedia Integration Language (SMIL). This section provides information about Media Overlays, SMIL files, and tips for using Media Overlays to create iBooks read aloud content.

Read Aloud books require a custom preview. See [“Samples for Read Aloud Books”](#) (page 51).

### About the Read Aloud Feature User Interface

Books that have the read aloud feature have an audio button in the top toolbar. Tapping the audio button opens a popover with a volume slider, a Turn Pages option menu, and a Start Reading/Stop Reading button. Reading begins on the current page and text is highlighted as the audio is read. If automatic page turn is on, the page turns after the audio content on the page finishes playing.



## Tools for Creating Read Aloud Content for iBooks

To create read aloud content for iBooks, you need:

- a Fixed Layout book
- a narrative audio file
- an audio editing tool for marking times within an audio file

---

**Note:** This document has instructions for marking start times and end times in an audio file using Audacity, a free, cross-platform audio editing tool. You can use another audio editing tool to mark start and end times in an audio file. Steps for marking start and end times in an audio file will vary depending on the tool you are using.

---

## Media Overlays Structure

During the read aloud narration, the text being read can be highlighted word-by-word, sentence-by-sentence, or not highlighted at all. Highlighting words during read aloud is accomplished using Media Overlays. The Media Overlay is EPUB3's method of syncing a portion of an audio file to a phrase of corresponding text. Text phrases are identified using a standard HTML `id` attribute. The corresponding audio is referenced by a start time and end time. The identified text and audio are paired together using a SMIL XML file. The SMIL file contains a series of `<par>` elements, each containing an `<audio>` element and a `<text>` element. Both the `<text>` and `<audio>` elements contain a required `src` attribute. The `src` attribute used in the `<text>` element uses a URL with a fragment identifier (the segment attached to the end of the `src` attribute starting with a # (hash)) to point to the identified word, text phrase, or sentence. The `src` attribute used in the `<audio>` element is a URL pointing to the location of the audio file within the EPUB bundle. The highlighting of words or sentences in the `<text>` element is defined by the fragment identifier, and the corresponding spoken words or sentences in the `<audio>` element is defined by the attributes, `clipBegin` and `clipEnd`.

---

**Note:** Apple recommends that you start the audio on the title page and have the title and author read as part of the audio.

---

## SMIL File Example

```
<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns="http://www.w3.org/ns/SMIL" version="3.0"
  profile="http://www.idpf.org/epub/30/profile/content/">
  <body>
    <par id="par1">
      <text src="page1.xhtml#word0"/>
      <audio src="audio/page1.m4a" clipBegin="5s" clipEnd="15s"/>
    </par>
    <par id="par2">
      <text src="page1.xhtml#word2"/>
      <audio src="audio/page1.m4a" clipBegin="15s" clipEnd="25s"/>
    </par>
  </body>
</smil>
```

## HTML File Example

```
<p>
  <span id="word0">Shall</span>
  <span id="word1">I</span>
  <span id="word2">compare</span>
  <span id="word3">thee</span>
  <span id="word4">to</span>
  <span id="word5">a</span>
  <span id="word6">summer's</span>
  <span id="word7">day?</span>
</p>
```

## Notes

- All `<par>` elements must follow the narrative order of the book. (For example, `<par id="par2">` must follow `<par id="par1">`)
- Highlighting the words during read aloud can be as detailed or broad as the content-creator defines it. For children's books, word-by-word highlighting is strongly preferred. Text ID attributes could also be defined at a sentence level.
- The highlighting is defined using CSS. You can set the color of the highlight or make the color of the highlight the same color as the text to turn off the highlighting. See [“CSS Styling of Media Overlays”](#) (page 68).
- Create one SMIL document per XHTML document.

## Pages Without Audio

You can set the timing for page turning in pages that do not have audio. iBooks has two default zoom levels in each orientation: page and spread. When a reader zooms into a page, each page is focused independently during navigation. When a reader zooms to a spread, the spread is treated as a single step during book navigation.

1. If **Turn Pages** is set to **Automatically**, iBooks pauses reading for 3 seconds on any pages or spreads that do not have any associated audio. After 3 seconds, reading continues, and the reader is taken to the next page or spread.
2. If **Turn Pages** is set to **Manually**, iBooks takes the reader to pages or spreads with no audio, and the corner of the page is immediately turned up, indicating to the reader that it is time for them to turn the page.

To override this behavior and skip the spread entirely, provide a `<par>` that corresponds to the skipped spread, and define a duration of 0s. If you want a pause longer than 3 seconds, build that time into the audio file.

## Processing Audio for Media Overlays

This section describes how to work with pages that have audio, including marking start and end times of the audio.

### Marking Start and End Times in Audio Files

Narrative audio can be one long audio file or a series of clips. An audio editing tool can be used to mark the start and end time of a word or text phrase within an audio file, making it easier to define in an SMIL file.

---

**Note:** The following instructions show how to mark start and end times in an audio file using Audacity, a free, cross-platform audio editing tool. You can use another audio file tool to mark the start and end times in an audio file, however steps to do this will vary depending on the audio editing tool used.

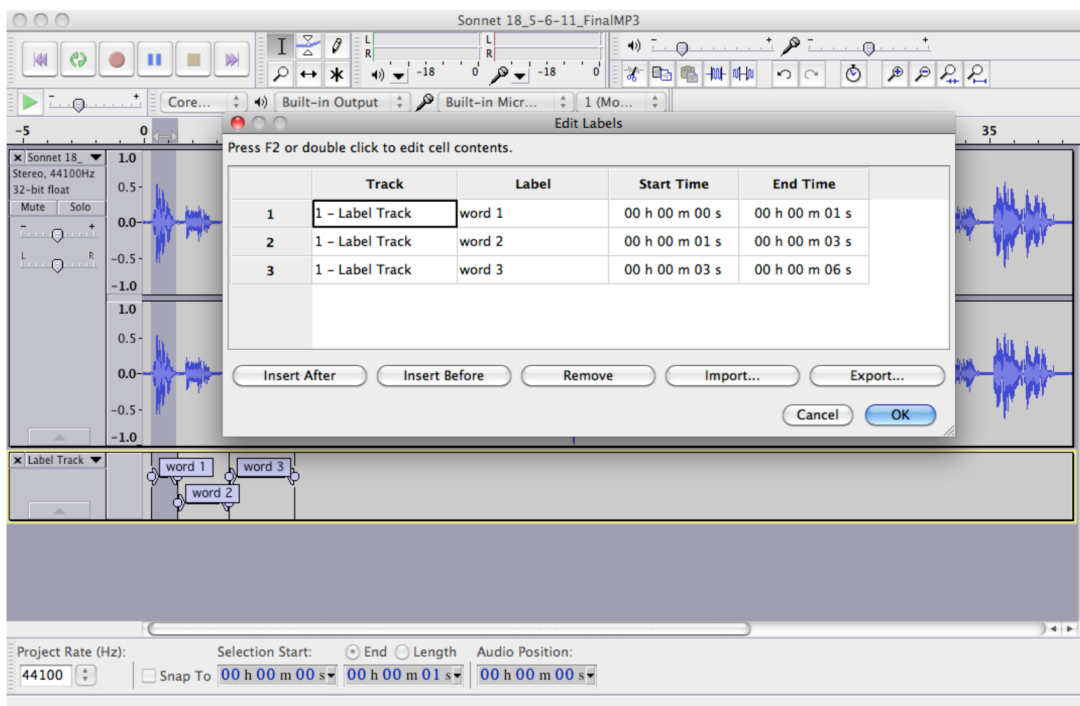
---

#### To mark the start and end times in an audio file using Audacity:

1. Import the audio file.
2. Move the cursor to the desired start time.
3. Press **Command-B** to add a label to the cursor location.
4. Name the label using a naming convention with no spaces.
5. In the label track, drag the the right-side handle of the newly added marker to the end time. The label now represents a span, marking both the start and end time of a phrase.

6. To export one or more labels, go to **File > Export Labels** and save the labels as a text document.

Final audio files should be encoded following the standard iBooks audio encoding guidelines. Audio tracks should be in stereo. Use iTunes to encode audio files as AAC at 256 kbps with `.m4a` as the file extension.



## CSS Styling of Media Overlays

The currently-playing content will highlight as it is read aloud. iBooks has a default blue highlight, but you can specify another style by defining your own CSS styles for the media overlay active class. This class supports standard CSS styling (for example, color, text-shadow, and so on) to indicate that the content is active. If the book's text has a defined color (not default black), the book must include a defined `-epub-media-overlay-active` color because the text color will overwrite iBook's default color for the actively-playing word.

To define your own media overlay active class, you must define the class name in the Package Document (`.opf`) metadata, using the meta property, `active-class`. For example:

```
<meta property="media:active-class">-epub-media-overlay-active</meta>
```



Then, include that class name in your stylesheet. CSS example:

```
.-epub-media-overlay-active {  
    color: red;  
}
```

## Ambient Soundtrack

To add an ambient soundtrack to your book, include an audio element with the attribute `epub:type="ibooks:soundtrack"` and include both the `xmlns:epub` namespace and the `epub:prefix="ibooks:"` namespace (see the code example below).

---

**Note:** It is best to have a single soundtrack throughout the entire book. If you define a new soundtrack for each spread, know that there is a delay between when the previous soundtrack ends and the new soundtrack starts.

---

## Ambient Soundtrack Example

```
<html xmlns="http://www.w3.org/1999/xhtml"  
xmlns:epub="http://www.idpf.org/2007/ops"  
epub:prefix="ibooks: http://vocabulary.itunes.apple.com/rdf/ibooks/vocabulary-extensions-1.0/">  
...  
    <audio epub:type="ibooks:soundtrack" src="audio/sonata14.m4a"/>  
...  
</html>
```

---

**Note:** In the `prefix` example above, the space between `ibooks:` and `http://` is important; it needs to be a single, regular space character and not a return, non-breaking space, or any other kind of whitespace.

---

If you have a single soundtrack that is intended for the entire book, include the same audio file in every XHTML document. iBooks will play this soundtrack seamlessly across documents. This audio file will play on a loop.

For ideal performance, use CSS to place soundtrack audio outside of the page boundary.

```
audio {  
    position: absolute;  
    top: -30px;  
}
```

## Embedding Read Aloud Controls

If a book contains Media Overlays, the read aloud controls automatically appear in the iBooks toolbar. However, you may also embed Start, Stop, and toggle controls in the content of the book by including the iBooks `readaloud` attribute. If you include this attribute, you must also include the iBooks namespace and prefix in `<html>`.

### iBooks Prefix

```
epub:prefix="ibooks: http://vocabulary.itunes.apple.com/rdf/ibooks/vocabulary-extensions-1.0"
```

---

**Note:** In the prefix example above, the space between `ibooks:` and `http://` is important; it needs to be a single, regular space character and not a return, non-breaking space, or any other kind of whitespace.

---

## Embedded Controls Example

```
<p ibooks:readaloud="startstop">Read Aloud</p>
```

The following attributes are supported:

"start", which starts `readaloud`

"stop", which stops `readaloud`

"startstop", which plays `readaloud` if stopped, or stops `readaloud` if playing.

## Defining the Page Turn Style

You can define the Page Turn Style of the embedded read aloud controls. The page turn style can be automatic or manual. These two values function the same as if the reader selected Automatic or Manual page turn directly from the Read Aloud menu.

Example:

```
<p ibooks:readaloud="startstop" ibooks:readaloud-turn-style="automatic">Automatic</p>  
<p ibooks:readaloud="startstop" ibooks:readaloud-turn-style="manual">Manual</p>
```

## Controlling Narration with Read Aloud Books

By default, if a Read Aloud book includes audio or video files in addition to the read aloud narration, the media will play in parallel with read aloud. You can control the read aloud behavior with a few simple changes.

To control read aloud behavior, make sure the namespace is set to

`http://vocabulary.itunes.apple.com/rdf/ibooks/vocabulary-extensions-1.0/`; this is typically done on the document's HTML tag.

- If you are using our iBooks JS library, add a `data-pause-readaloud` attribute to your trigger element with its value set to `"true"`
- If you are using `<audio>` or `<video>` elements, add a `pause-readaloud` attribute to your media element with its value set to `"true"`

When a user interacts with the media elements, the narration will pause while the media plays; narration will resume when the media is finished playing.

The following examples show how to use the `pause-readaloud` attribute with static media elements:

```
<video src="video/H264-640x480.m4v" ibooks:pause-readaloud="true" controls="controls"  
  poster="images/posterimage.jpg"/>
```

```
<audio src="audio/loop.m4a" ibooks:pause-readaloud="true" controls="controls"/>
```

The following examples show how to use the `pause-readaloud` attribute using the iBooks JS library:

```
<div class="ibooks-media-audio" data-ibooks-audio-src="audio/loop.m4a"  
  data-ibooks-pause-readaloud="true">Audio</div>
```

Additionally, when using iBooks JS including `data-ibooks-audio-reset-on-play` and setting it to `"true"` will force the media element to play from the beginning each time the user triggers it.

Using this attribute requires iBooks 1.5 or later.

## Styling Read Aloud Controls

Use `.-ibooks-media-overlay-enabled` to style `"startstop"` element depending on the current state.

```
html #mybutton {  
    /* style when readaloud is stopped */  
    color: green;  
}
```

```
html.-ibooks-media-overlay-enabled #mybutton {  
    /* style when readaloud is playing */  
    color: red;  
}
```

## Adding a SMIL File and Audio File to an EPUB

SMIL documents and audio files must be included in the manifest of the OPF. In addition to listing the SMIL and narrative audio files, a SMIL document must be cross-referenced with its corresponding XHTML document by including the `media-overlay` attribute on the XHTML document. The `media-overlay` attribute has a value equal to the `id` name of the corresponding SMIL document.

### Manifest Example

```
<manifest>  
    ...  
    <item id="page1" href="page1.xhtml" media-type="application/xhtml+xml"  
        media-overlay="mo-page1"/>  
    <item id="audiol" href="page1.smil" media-type="application/smil+xml"/>  
    <item id="narrat" href="audio/page1.m4a" media-type="audio/mpeg"/>  
    ...  
</manifest>
```

### Media Duration

The Package Document must include the duration of each Media Overlay as well as of the entire book. The `refines` attribute references the ID of the corresponding SMIL document.

For example:

```
<package>  
    <metadata xmlns="http://www.idpf.org/2007/opf"  
        xmlns:dc="http://purl.org/dc/elements/1.1/" >  
        . . .
```

```
<meta property="media:active-class">-epub-media-overlay-active</meta>

<meta property="media:duration">0:00:48</meta>
<meta property="media:duration" refines="#moPage001">0:00:03</meta>
<meta property="media:duration" refines="#moPage002">0:00:06</meta>
<meta property="media:duration" refines="#moPage003">0:00:06</meta>
<meta property="media:duration" refines="#moPage004">0:00:05</meta>
<meta property="media:duration" refines="#moPage005">0:00:15</meta>
<meta property="media:duration" refines="#moPage007">0:00:12</meta>

. . .
</metadata>
```

# Guidelines for Multi-Touch Books

## Overview

iBooks Author allows you to create and publish amazing Multi-Touch books for iPad. With galleries, video, interactive diagrams, 3D objects, mathematical expressions, and more, these books bring content to life in ways the printed page never could. Multi-Touch books can contain a wide variety of interactive elements, which makes them particularly well-suited for textbooks, cookbooks, history books, and picture books. iBooks only accepts Multi-Touch books made with the latest version of iBooks Author.

The iBooks Author support site includes best practices for audio, video, fonts, 3D objects, accessibility, and more:

<http://support.apple.com/kb/HT5065> – Add video to your book

<http://support.apple.com/kb/PH2791> – Add a movie or an audio file

<http://support.apple.com/kb/HT5067> – About the Keynote widget

<http://support.apple.com/kb/HT5068> – About HTML5 widget creation

---

**Note:** Multi-Touch books require a custom sample for previews. See “[Samples for Multi-Touch Books](#)” (page 51) for details.

---

# Revision History

## Previous Spec Revisions

The following table lists the previously-released specifications and the revisions:

Date/Version	Summary
February 21, 2013 - Version 5.1 Revision 1	Clarified image size, text direction, and versioning. Added support for Tatechuyoko. Added a layout property to hide the book binding in Fixed Layout books. iBooks' fonts follow the font orientation guidelines of UTR 50 draft 6. Clarified that nonlinear documents are not supported for Fixed Layout books.
October 31, 2012 - Version 5.1	Added information on support for EPUB 3. Added versioning. Added support for Gaiji. Made changes to Read Aloud books based on new EPUB 3 support. Changed how linking works. Changed iBook.js to iBooks.js (and iBook JS to iBooks JS) to reflect the updated name. Updated iBooks namespace throughout. Removed all sections relevant to EPUB 2; refer to the <i>iBookstore Asset Guide 5.0</i> for EPUB 2.
May 30, 2012 - Version 5.0	Book marketing image requirements have changed. Removed TIFF from the list of recommended image formats.
April 3, 2012 - Version 4.9	Both Flowing and Fixed Layout books can have custom page numbers. Added a new chapter for Multi-Touch books. Added best practices for images and Flowing books. Added asset requirements for screenshots. Clarified that Read Aloud books require a custom preview.
December 8, 2011 - Version 4.8	Reorganized and simplified the structure of the document and eliminated redundant sections. Renamed “Standard EPUBS” to “Flowing Books” throughout the document. Clarified iBooks’ treatment of tables. Added best practices for fonts and Fixed Layout Books. Added a new section for books that contain interactive content. Inline CSS within HTML comments will be ignored. A PDF referenced in the <spine> must be set to <code>linear=no</code> .
July 13, 2011 - Version 4.7 Revision 4	Added a chapter describing how to deliver read aloud content. Made several corrections throughout the document.

Date/Version	Summary
June 16, 2011 - Version 4.7 Revision 3	Removed documentation on features not supported in iBooks 1.3.
June 1, 2011 - Version 4.7 Revision 2	Clarified multimedia assets delivery requirements.
March 1, 2011 - Version 4.7	Added a link to an example use of <code>linear="no"</code> . Added color space requirement for HD source.
February 9, 2011 - Version 4.6	Removed size recommendations for full-bleed images. Images should not contain embedded text.
December 21, 2010 - Version 4.5 Revision 1	Added support for Fixed Layout Books (such as picture books, cookbooks, and art books). Added support for embedded fonts and for <code>linear="no"</code> . Page breaks are now supported for standard (Flowing) EPUBs.
December 1, 2010 - Version 4.5	First release of the standalone asset specification for books. This guide contains only the video and audio source formats for books; it does not list formats for music, film, and TV. An additional chapter describes best practices in preparing book content for iBooks.

## Changes in the iBooks Asset Guide 5.1 Revision 1

### Text Direction: Tatechuyoko

In vertical text, it is sometimes preferable to include a short run of horizontal numbers or Latin text. This is known as Tatechuyoko. To create Tatechuyoko, you must use the CSS property, `-webkit-text-combine`. See [“Tatechuyoko”](#) (page 18) for details.

### Text Direction: Clarification

Each content document can support a single `writing-mode` value. If you want both horizontal and vertical text in your book, then each text direction must be split into separate content documents. Tatechuyoko should not be created using the CSS `writing-mode` property.

### Gaiji Images

To make a gaiji image accessible for readers who use VoiceOver, the `alt` attribute for a gaiji image must be included. See [“Gaiji Accessibility”](#) (page 28) for details.



## Book Cover/Marketing Image

The 2 million pixel limit on image size does *not* apply to the external cover/marketing image that is sent alongside the book asset. It applies only to images inside the book asset.

## Book Versioning

Apple recommends that you declare the first asset delivered to iBooks version 1.0. The version number is consumer-facing and is meant to manage delivering updates to consumers. Future versions should increment as is appropriate.

Changed the word “preview” to “sample.” Added a sentence to explain where the What’s New text is displayed.

## Fonts

iBooks' fonts follow the font orientation guidelines of [UTR \(Unicode Technical Report\) 50 draft 6](#).

## Flowing Books: Scrolling with Vertical Text

For Flowing books, iBooks has several viewing themes that the reader can select. The Scroll theme scrolls vertically for books with horizontal text, and scrolls horizontally for books with vertical text. This is useful for a book with vertical Japanese or Chinese text. When rendered with iBooks' Scroll theme, vertical text is most easily read scrolling right to left on the horizontal axis. See [“Scrolling in Flowing Books”](#) (page 55) for details on scrolling, and the metadata required for defining a book's scroll direction.

## Fixed Layout Books: Hide Book Binding

If you create a Fixed Layout book that is two documents per spread, but you do not want it to have the book-like appearance, you can hide the book binding by including the `binding` layout property and giving it a value of `false`. See [“Defining Book Layout Metadata”](#) (page 57).

## Fixed Layout Books: Clarification

Nonlinear documents are not supported Fixed Layout books; only Flowing books support nonlinear documents.

## Read Aloud Books: Clarification

If you include the `readaloud` attribute, you must also include the iBooks namespace and prefix in `<html>`. See [“iBooks Prefix”](#) (page 70) for an example.

## Changes in the iBooks Asset Guide 5.1

### EPUB 3 Support

iBooks 3.0 supports EPUB 3 for both Flowing and Fixed Layout books including features such as pop-up footnotes, page progression direction, text direction, interactivity, and MathML. See [“EPUB 3 Features”](#) (page 15) for more information.

The structure of a version 3 EPUB differs from a version 2 EPUB. A version 3 EPUB includes a Package Document (`.opf`) and a Navigation Document (`toc.xhtml`). The Package Document includes all of the information about a book including Metadata, Manifest, and Spine. The Navigation Document includes information about how to navigate a book and includes the Table of Contents, Landmarks, and Page-list. In EPUB 3, the Apple Display Options file has been replaced with Layout Metadata in the Package Document. See [“Structure of a Version 3 EPUB”](#) (page 9).

### Book Versioning

For EPUB 3 and Multi-Touch books, you can provide book versioning information when you publish a new version of a book previously published to iBooks. The updated book replaces the old book on the Store and is available to future consumers. Customers who have downloaded the old version of the book will be notified that a new version is available for download, and if the customer chooses to download it, the new version will replace the prior version on their device. For Multi-Touch books, book versioning is built into iBooks Author version 2. In iBooks Author, when you publish a new version of a previously published book, you are asked to provide versioning information during the publishing process. For EPUBs, see [“Book Versioning in EPUBs”](#) (page 22).

### Read Aloud Books

In Read Aloud books, the currently-playing text will highlight as it is read aloud. iBooks allows you to embed read aloud controls in the page of a book. With iBooks 3.0, you can define if the page turn style is automatic or manual for the embedded read aloud controls. See [“CSS Styling of Media Overlays”](#) (page 68) for more information.

New to EPUB 3, the Package Document must include the duration of each Media Overlay as well as the duration of the narration audio for the entire book. See [“Media Duration”](#) (page 72) for an explanation.

### Links

You can link directly to your book on the iBooks Store from your own website, from online advertisements, or within a book. The linking structure can link by ISBN. See [“Linking Directly to Your Book in iBooks”](#) (page 35) for details.

By default, links in EPUBs are underlined and blue. To customize the style of the links, you can use CSS to define the color and style of the anchor element. If you customize the style, you must also include the `specified-fonts` layout metadata. See [“Defining Book Layout Metadata”](#) (page 57).

## Images

Clarified image sizes for book cover art and images inside the book. Added support for Gaiji. See [“Gaiji”](#) (page 26) for more information.

## Screenshots

You can deliver up to five screenshots for a book.

## Miscellaneous Changes

ibook.js has been changed to ibooks.js (and iBook JS to iBooks JS) to reflect the updated name. The iBooks namespace has changed to `xmlns:ibooks="http://vocabulary.itunes.apple.com/rdf/ibooks/vocabulary-extensions-1.0/"`.

# Changes in the iBooks Asset Guide 5.0

## Book Cover Art

The book’s cover art, which is also known as the marketing image, that appears on iBooks must use RGB color mode and be at least 1400 pixels along the shorter axis. The book’s cover art file must also be a high-quality JPEG with .jpg extension or PNG with .png extension. Do not increase the size of a smaller image to meet the minimum image size dimension standard. Excessively blurry or pixelated images will be rejected.

# Changes in the iBooks Asset Guide 4.9

## Flowing and Fixed Layout Books: Custom Page Numbers

In Flowing and Fixed Layout books, `<pageList>` can be used to define the page numbers that iBooks assigns to each page. For example, the first page of the book is page 1 by default, but in some books, this may be the cover page and shouldn’t be numbered. You can use `<pageList>` to define an empty string to that page so that it is not numbered. Similarly, you can define Roman numerals (i, ii, iii), letters (a, b, c), or numbers (1, 2, 3) for page numbers. If you use something other than numbers or a single word, make sure it is meaningful, keep it very short, and check to make sure it doesn’t get truncated on the device. Using `<pageList>` requires iBooks 2.1 and newer.

## Multi-Touch Books for iPad

Added a new chapter to describe best practices and asset requirements and encoding for Multi-Touch books for iPad. Multi-Touch books require iBooks 2.0 or later, iOS 5.0 or later, Mac OS X 10.7.3 Lion or later, and iTunes 10.5.3 or later.

## NCX File Clarifications

For optimal customer experience, Apple requires the `navMap` element in the NCX. iBooks uses the data provided in the NCX to build the customer-facing table of contents. The `navMap` element must contain one or more `navPoint` elements. Each `navPoint` creates an entry in the table of contents. However, if the `navMap` contains only one `navPoint`, iBooks suppresses the list table of contents, which could be useful when creating Fixed Layout books. For example, a picture book does not need a list table of contents as it does not have chapters. iBooks will always create a thumbnail table of contents for a Fixed Layout book.

## Images: Best Practices

When testing your book, be sure to preview it in night mode (the Night option under Theme). The Night reading theme makes reading books in the dark easier on the eyes. In night mode, the transparent areas of an image will be black. If your image has dark text within a transparent image, that text could be difficult to read in night mode. Instead, we suggest you use a JPG with a white background.

We recommend providing images that are at least 1.5 times the intended viewing size up to a maximum of 2 million pixels (image height multiplied by image width should be less than 2,000,000px). For example, a single-page, full-bleed image should be around 1200px by 1600px.

## Links

When setting up a link to a book on iBooks, use `itms-books://` instead of `http://` at the beginning of the iBooks URL. This takes the readers directly to the book on iBooks without first displaying a message that they're leaving the iBooks app. See [“Links”](#) (page 35) for more details.

## Flowing Books: Font Best Practices

Font sizes should be defined in ems (`em`) or pixels (`px`), not by name, for example, small or large. The main text of a book should either not have a defined `font-size` or should have a `font-size` of `1em`. This will ensure ideal readability and font scaling.

## Read Aloud Books

Read Aloud Books require a custom preview. See [“Samples for Read Aloud Books”](#) (page 51).

## Screenshot Requirements

Screenshots must be full-sized iPad 1/2 images (1024 x 768 or 768 x 1024) or full-sized new iPad images (2048 x 1536 or 1536 x 2048) in PNG (.png) or JPEG (.jpg) format. For a cleaner look, you can optionally remove the status bar (making the images 1004 x 768 or 748 x 1024 for iPad 1/2, or 2028 x 1536 or 1536 x 2028 for the new iPad). Note that screenshots can only be provided when the book is first delivered.

### Adobe Creative Suite 5.5 Error Fix

The Adobe Creative Suite 5.5 bug that resulted in XHTML errors in EPUBs exported by InDesign has been fixed. If you have encountered this problem using InDesign CS5.5, apply the 7.5.2 update and regenerate the EPUB file. You can find the update here: <http://www.adobe.com/downloads/>.

## Changes in the iBooks Asset Guide 4.8

### Flowing Books: Text Alignment

If specifying line-height, extra spacing between text blocks should be a multiple of the specified line-height to keep text aligned across the spine. See “[Alignment](#)” (page 52).

### Flowing Books: Tables

In Flowing Books, iBooks sizes large tables to fit within the width of the page. When a reader double taps the table, the table will open in a new web view that overlays the book. In this web view, the reader can pan and zoom in on the table.

### Flowing Books: Spine Items

If the `<spine>` references a PDF, the `linear` attribute must be set to `no`. If the `linear` attribute is set to `yes`, delivery will fail.

### CSS Styles

If you define CSS styles inline using a `<style>` tag in the XHTML document, note the following changes. In iOS 5, HTML comments in an XHTML document will always be treated as comments, even if they appear in inline CSS. This behavior is compliant to XHTML standards. Previously, HTML comments within a style tag were ignored, thus applying the style within them, but beginning in iOS 5, `<style>` nested within `<!-->` will be ignored. Verify your books and correct if necessary.

### Image Alt Attribute Clarification

For accessibility, the `alt` attribute for an image must be included. To read best practices on using the `alt` attribute, see “[Interior Image Requirements](#)” (page 24).

### Fixed Layout Books: Best Practices

When developing Fixed Layout Books, you may notice that iBooks sometimes remembers the appearance of your book even after it has been deleted and replaced with an updated version. This is due to caching. If you anticipate making a number of revisions to your books while editing, Apple recommends that you use the Book Proofer app (which can be downloaded from the Deliver Your Content page in iTunes Connect). Another option is to include the edition's date in the metadata of the OPF. Changing this date will circumvent iBooks' caching.

A Fixed Layout Book must include a `<reference>` of `type="text"` in the `<guide>` block of the `.opf` file.

### Read Aloud Books

If a Read Aloud Book includes audio or video files in addition to the read aloud narration, you can include the `pause-readaloud` attribute (or `data-pause-readaloud` if using iBooks JS) on the `<audio>` or `<video>` element. The namespace must be set to `http://apple.com/ibooks/html-extensions` and `pause-readaloud` to `"true"`. Using this attribute requires iBooks 1.5. See [“Controlling Narration with Read Aloud Books”](#) (page 71) for more information.

### Books with JavaScript Interactivity

A new section describing delivery requirements and best practices for books with interactive content has been added.

### Adobe Creative Suite 5.5 Error

Adobe Creative Suite 5.5 has a bug that may result in XHTML errors in EPUBs exported by InDesign. See [“Book Validation”](#) (page 48) for details.

## Changes in the iBooks Asset Guide 4.7 Revision 4

### Read Aloud Content

You can add read aloud content to Fixed Layout Books using Media Overlays, an EPUB-specific subset of Synchronized Multimedia Integration Language (SMIL). These SMIL files sync audio with the text, so readers can follow along as the words are read aloud. In some books, the words can be highlighted as they are spoken. See [“Read Aloud Books”](#) (page 63) for more information.

## Changes in the iBooks Asset Guide 4.7 Revision 3

### Corrections

Previous versions of the iBooks Asset Guide incorrectly described features of iBooks 1.3.

## Changes in the iBooks Asset Guide 4.7 Revision 2

### Multimedia Delivery Requirements

Videos embedded in a multimedia EPUB must use H.264 video codec and end with MP4 or M4V file extension. Audio embedded in a multimedia EPUB must use AAC audio codec and end with M4A file extension.

## Changes in the iBooks Asset Guide 4.7

### <spine> Element

The <spine> must have at least one item with the attribute `linear="yes"`, which is the default. Any spine item without the `linear` attribute defaults to `linear="yes"`. Any spine item with the `linear="no"` attribute will open in its own window. See [“Nonlinear Content”](#) (page 53) for more information and a link to an example.

### Samples

Removed the sentence “If you deliver via iTunes Producer, creating custom samples is not an option at this time.” You can now deliver samples using iTunes Producer.

### Assets

HD video source must be tagged as 709 color space.

## Changes in iBooks Asset Guide 4.6

### Images: Clarifications

The following sentence was removed from the Images section: “We recommend full-bleed images that are at least 1476 x 1970 pixels (double the size of a single page in iBooks on the iPad) up to a 2 million pixel limit.” It was removed because the size (1476 x 1970 pixels) results in an image that is more than 2 million pixels.

Embedding text in images creates issues that cause a large number of customer complaints: customers can’t use the dictionary or search the text, and in addition, the book becomes not accessible for persons with disabilities. Therefore, books with images that contain embedded text will be rejected from sale on iBooks.

## Changes in iBooks Asset Guide 4.5 Revision 1

### Picture Book Support

New features have been added to support how a book displays when a customer opens a fixed layout (picture) book. Fixed Layout Books require that a display option be set to indicate the EPUB has a fixed layout (`"fixed-layout"=true`). See [“Text Directions”](#) (page 17) for instructions. Other display options for Fixed Layout Books specify if the EPUB should open to a two-page spread, and open in portrait or landscape orientation. Display options can be specified by platform (iPad and iPhone/iPod touch).

### Embedded Font Support

EPUBs can now use embedded fonts, in addition to the already-supported system fonts. If you want to use embedded fonts, you must specify that the EPUB contains fonts. Otherwise, the embedded fonts will not be respected, and user-specified fonts will be used instead.

### Navigation Support

In the OPF `.opf` file, the `<spine>` element is used to indicate the linear reading order of the content. When the person reading the book uses “next page” navigation, the pages are displayed based on the spine order. If a spine item is auxiliary to the main flow of the book (for example, an answer key in a textbook), you can use the optional `linear` attribute set to `no` to skip over the item to avoid disrupting the reading flow.

### Page Break Support

In standard, Flowing EPUBs, you can use the CSS properties `page-break-before` and `page-break-after` to add page breaks before or after an element. See [“Page Breaks”](#) (page 24).

## Changes in iBooks Asset Guide 4.5

First release of the standalone asset specification for books. (The version number of 4.5 was used to match the current version of the schema.) This guide contains only the video and audio source formats for books; it does not list formats for music, film, and TV. An additional chapter describes best practices in preparing book content for iBooks.





Apple Inc.

© 2013 Apple Inc.

All rights reserved.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Apple Inc.

1 Infinite Loop

Cupertino, CA 95014

408-996-1010

Apple, the Apple logo, Final Cut, Final Cut Studio, iBook, iBooks, iPad, iPhone, iPod, iPod touch, iTunes, Keynote, Mac, Mac OS, Numbers, OS X, Pages, Safari, TrueType, and WebObjects are trademarks of Apple Inc., registered in the U.S. and other countries.

Multi-Touch is a trademark of Apple Inc.

iBookstore is a service mark of Apple Inc.

Adobe, Acrobat, and PostScript are trademarks or registered trademarks of Adobe Systems Incorporated in the U.S. and/or other countries.

Java is a registered trademark of Oracle and/or its affiliates.

Times is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.