
iBookstore Asset Guide 5.0



Contents

Overview 5

Changes Made in this Release 5

 What's New in the iBookstore Asset Guide 5.0? 5

Digital Book Essentials 6

Overview 6

Structure of a Version 2 EPUB 6

 The OPF File 7

 NCX 11

 Configuring Apple Display Options 13

Book Marketing Image 16

Recommendations for Creating Books 16

 Presentation and Styling 16

 In-Book Cover Page 18

 Images 18

 Screenshots 20

 Fonts 20

 Audio and Video 21

 Links 26

 PDFs 27

 Adobe Page Template 28

Books with JavaScript Interactivity 28

 Overview 28

 Best Practices for Developing Scripts for Books with JavaScript Interactivity 29

 Designing Content for Books with JavaScript Interactivity 29

 Tips for Improving Performance in Books with JavaScript Interactivity 35

 Technical Guidelines and Requirements 35

Testing During Development 39

Book Validation 39

Book Samples 40

 About Book Samples 40

Guidelines for Flowing Books 43

Overview 43

- [Content Structure](#) 43
- [Alignment](#) 43
- [Nonlinear Content](#) 44
- [Tables](#) 46
- [Fonts](#) 46

Guidelines for Fixed Layout Books 47

- [Overview](#) 47
- [Setting Up the Document](#) 47
- [Table of Contents](#) 48
- [Using Fonts in Fixed Layout Books](#) 50
- [Optimizing Images in Fixed Layout Books](#) 50
- [Layout](#) 50
- [Text](#) 51
- [Read Aloud Books](#) 52
 - [Overview](#) 52
 - [About the Read Aloud Feature User Interface](#) 52
 - [Tools for Creating Read Aloud Content for iBooks](#) 54
 - [Media Overlays Structure](#) 54
 - [Processing Audio for Media Overlays](#) 56
 - [CSS Styling of Media Overlays](#) 57
 - [Ambient Soundtrack](#) 58
 - [Embedding Read Aloud Controls](#) 58
 - [Styling Read Aloud Controls](#) 60
 - [Adding a SMIL File and Audio File to an EPUB](#) 60
- [Configuring Display Options](#) 61

Guidelines for Multi-Touch Books 62

- [Overview](#) 62
- [Audio and Video Asset Requirements](#) 62
 - [Media Widget Video and Audio Asset Requirements](#) 63
 - [HTML Widget Video and Audio Asset Requirements](#) 64
- [Keynote Best Practices](#) 64

Revision History 65

- [Previous Spec Revisions](#) 65
- [Changes in the iBookstore Asset Guide 4.9](#) 66
- [Changes in the iBookstore Asset Guide 4.8](#) 67
- [Changes in the iBookstore Asset Guide 4.7 Revision 4](#) 69
- [Changes in the iBookstore Asset Guide 4.7 Revision 3](#) 69

[Changes in the iBookstore Asset Guide 4.7 Revision 2](#) 69

[Changes in the iBookstore Asset Guide 4.7](#) 69

[Change in iBookstore Asset Guide 4.6](#) 70

[Changes in iBookstore Asset Guide 4.5 Revision 1](#) 71

[Changes in iBookstore Asset Guide 4.5](#) 71

Overview

This document provides delivery information for all accepted media and files for the iBookstore. It describes the basics of EPUB structure and design, as well as guidelines specific to each type of EPUB and Multi-Touch books. 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 (`flowing2-2.epub`) and the other is for Fixed Layout books (`fixedlayout2-3.epub`).

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

Changes Made in this Release

Date/Version	Changes Made
May 30, 2012 - Version 5.0	Book marketing image requirements have changed. Removed TIFF from the list of recommended image formats.

What's New in the iBookstore Asset Guide 5.0?

Book Marketing Image

The book marketing image (the image used on the Storefront) must be at least 1400 pixels wide (2400 pixels recommended for best results). The file must be a high-quality JPEG with `.jpg` extension or PNG with `.png` extension and must be RGB. Do not increase the size of a smaller image to meet the minimum size standard. Excessively blurry or pixelated images will be rejected.

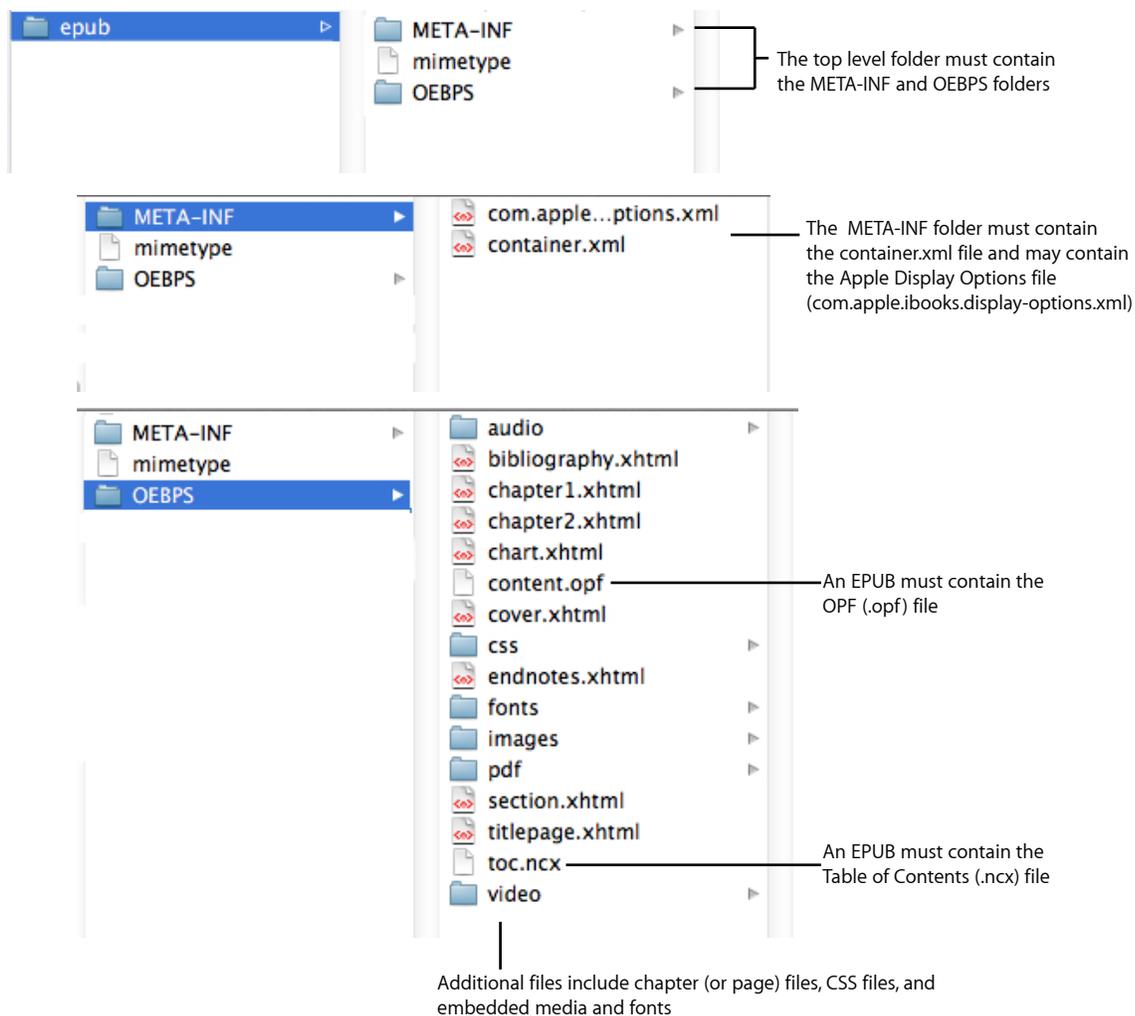
Digital Book Essentials

Overview

This chapter provides information on developing EPUBs and Multi-Touch books. iBookstore supports three book formats: Flowing books, Fixed Layout books, and Multi-Touch books.

Structure of a Version 2 EPUB

Flowing books and Fixed Layout books share a common structure as shown in the image below:



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

The OPF File

The OPF `.opf` file describes how the files containing the book content and the resources (media, CSS, and fonts) are connected into a logical publication. The OPF includes the metadata for the book, the manifest (which lists all the files in the book), the spine (which indicates the reading order of the content), and the guide (which lists the key files, much like signposts, so iBooks knows where the key parts of the book are). The sections that follow provide more details.

The Metadata

The `<metadata>` section of the OPF file includes information such as the title of the book, the creator, the publisher, the date, the book identifier, and the language. The `<dc:title>`, `<dc:identifier>`, and `<dc:language>` tags are required.

Keep in mind that the language supplied in the `<dc:language>` tag drives a number of iBooks features. Books with incorrectly labeled languages may not perform as expected for customers and could cause customer complaints. (See http://www.loc.gov/standards/iso639-2/php/code_list.php for a list of codes.) The example below shows the `<metadata>` block:

```
<?xml version="1.0" encoding="UTF-8"?>
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="bookid"
version="2.0">
  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns="http://www.idpf.org/2007/opf">
<!--Title of the book is required-->
  <dc:title>Flowing Example 2.2</dc:title>
  <dc:creator opf:role="aut" opf:file-as="APPLE">Apple Inc.</dc:creator>
  <dc:publisher>Apple Inc.</dc:publisher>
<!--A unique identifier for the book (using the ISBN is recommended)-->
  <dc:identifier id="bookid">1234567890</dc:identifier>
  <dc:date >2011-11-01</dc:date>
<!--Indicates the language of the book's content-->
  <dc:language>en</dc:language>
  <meta name="cover" content="img1"/>
</metadata>
```

The Manifest

The `<manifest>` section of the OPF file lists every file that makes up the book and specifies the file type of each. These files include the content documents (XHTML docs), the CSS, any embedded fonts, images, any audio files, video files, PDF files, any JavaScript files (used for books with interactive content), and any SMIL files (used for read aloud books). The following example shows the `<manifest>` block of the OPF file from a Fixed Layout book containing interactivity and read aloud content (for the sake of brevity, not all files are shown):

```
<manifest>
  <!--supplementary -->
  <item id="ncx" href="toc.ncx" media-type="application/x-dtbnex+xml"/>
  <item id="css" href="css/stylesheet.css" media-type="text/css"/>
  <item id="css-1" href="css/page01.css" media-type="text/css"/>
  . . .

  <!--javascript-->
  <item id="js-ib" href="js/ibook.js" media-type="text/javascript"/>

  <!--content-->
  <item id="pg001" href="page001.xhtml" media-type="application/xhtml+xml"
media-overlay="moPage001"/>
  <item id="pg002" href="page002.xhtml" media-type="application/xhtml+xml"
media-overlay="moPage002"/>
  <item id="pg003" href="page003.xhtml" media-type="application/xhtml+xml"
media-overlay="moPage003"/>
  . . .

  <!--images (only one .jpg and one .png are shown)-->
  <item id="cover1" href="images/cover.jpg" media-type="image/jpeg"/>
  <item id="pg4-fl" href="images/page4/flower.png" media-type="image/png"/>
  . . .

  <!--av-->
  <item id="narrat" href="audio/sonnet.m4a" media-type="audio/m4a"/>
  <item id="video1" href="video/video.m4v" media-type="video/mp4"/>
```

```
<!-- smil -->
<item id="moPage001" href="smil/page001.smil" media-type="application/smil+xml"/>
<item id="moPage002" href="smil/page002.smil" media-type="application/smil+xml"/>
<item id="moPage003" href="smil/page003.smil" media-type="application/smil+xml"/>
. . .
</manifest>
```

Unmanifested Files

All files included in EPUBs must be listed in the EPUB manifest (OPF file). EPUBs containing unmanifested files (that is, files not listed in the manifest) will fail import, as these files are by definition not intentionally included.

The Spine

The `<spine>` section of the OPF file indicates the linear reading order of the book's content (XHTML) files. When the person reading the book uses "next page" navigation, the pages are displayed based on the spine order. Each spine item is identified by `idref` and that value must match the `<item id>` listed in the `<manifest>` for the corresponding spine item, as illustrated below:

The items listed in the spine are identified using the item id in the `<manifest>` block

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

<!--content-->
<item id="cov" href="cover.xhtml" media-type="application/xhtml+xml"/>
<item id="tit" href="titlepage.xhtml" media-type="application/xhtml+xml"/>
<item id="sec" href="section.xhtml" media-type="application/xhtml+xml"/>
<item id="ch1" href="chapter1.xhtml" media-type="application/xhtml+xml"/>
<item id="ch2" href="chapter2.xhtml" media-type="application/xhtml+xml"/>
<item id="end" href="endnotes.xhtml" media-type="application/xhtml+xml"/>
<item id="bib" href="bibliography.xhtml" media-type="application/xhtml+xml"/>
<item id="cht" href="chart.xhtml" media-type="application/xhtml+xml"/>
```

The Guide

The `<guide>` block of the OPF file lists the key components of the book, such as the cover page, table of contents, bibliography and so on. The guide elements tell iBooks where the key parts of the book are to make it easier for readers to navigate them.

Within the package there may be one guide element, containing one or more reference elements. The guide element identifies fundamental structural components of the publication, to enable Reading Systems to provide convenient access to them.

The structural components of the books are listed in reference elements contained within the guide element. These components could refer to the table of contents, list of illustrations, foreword, bibliography, and other standard parts of the book.

The required `type` attribute describes the publication component referenced by the `href` attribute. The values for the `type` attributes must be selected from the list defined below when applicable. Other types may be used when none of the predefined types are applicable; their names must begin with the string `other`. The value for the `type` attribute is case-sensitive.

Apple strongly recommends that all book elements (for example, the cover, table of contents, first chapter, index, and so on) be identified in the `<guide>` block. The first chapter of the main body of text must be tagged with the type of `"text"`, with all other `<guide>` elements tagged with the appropriate type (`"toc"`, `"title-page"`, `"epilogue"`, `"preface"`, and so on). Use all of the reference types from the OPF spec (http://idpf.org/epub/20/spec/OPF_2.0.1_draft.htm#Section2.6) that apply. Within the `<guide>` block, there can be only one `<reference>` element of each type; for example, there cannot be multiple `<reference>` elements of type `"text"`:

```
<guide>
  <reference type="cover" ... />
  <reference type="toc" ... />
  <reference type="text" ... />
  <reference type="epilogue" ... />
</guide>
```

- When a customer first opens a Flowing book (as opposed to a Fixed Layout Book) on the iPad or iPhone, it will display the first spine item for which there exists a `<reference>` element in the `<guide>` with any of the following types:
 - `text`
 - `start`
 - `acknowledgements`
 - `dedication`
 - `epigraph`
 - `foreword`
 - `preface`
- To designate a start page with higher precedence over the types listed above, use the `<reference>` element of type `"other.reader-start-page"`.

Note A nonlinear document should not be included in the guide.

NCX

An NCX (Navigation Control XML) file defines a table of contents for a book. Two tags can be used to provide navigation within the EPUB: `<navMap>`, which is the actual table of contents and `<pageList>`, which provides page mapping. Both are described in the sections below.

Important 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. See [“Table of Contents”](#) (page 48) in the Fixed Layout chapter for information specific to Fixed Layout books.

Table of Contents Using `<navMap>`

The `<navMap>` tag is required must be specified in the NCX. This tag lets the reader jump directly to any of the major structural elements of the document (for example, chapter or section). These sections are defined by the book's creator and specified in the NCX.

The example below shows how to define a table of contents using `<navMap>`:

```
<?xml version="1.0" encoding="UTF-8"?>
<ncx xmlns="http://www.daisy.org/z3986/2005/ncx/" version="2005-1">
  <head>
    <meta name="dtb:uid" content="0123456789"/>
    <meta name="dtb:depth" content="1"/>
    <meta name="dtb:totalPageCount" content="0"/>
    <meta name="dtb:maxPageNumber" content="0"/>
  </head>
  <docTitle>
    <text>Fixed Layout Example 2.3</text>
  </docTitle>

  <!--TOC-->
  <navMap>
    <navPoint id="p001" playOrder="1">
      <navLabel>
        <text>Animated text</text>
```

```
        </navLabel>
        <content src="page001.xhtml#word1"/>
    </navPoint>
    <navPoint id="p002" playOrder="2">
        <navLabel>
            <text>Embedded read aloud controls</text>
        </navLabel>
        <content src="page001.xhtml#readAloud"/>
    </navPoint>
    <navPoint id="p003" playOrder="3">
        <navLabel>
            <text>Looping animation</text>
        </navLabel>
        <content src="page002.xhtml#loopingAnimation"/>
    </navPoint>
    ...
</navMap>
</ncx>
```

Page Mapping Using <pageList>

A <pageList> 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 <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.

Currently, <pageList> is supported for Flowing and Fixed Layout books. Apple recommends including <pageList> in all books to avoid the need to redeliver the book when <pageList> is supported.

The example below shows how to provide page navigation using <pageList> (for the sake of brevity, the <navMap> tag has been omitted):

```
<?xml version="1.0" encoding="UTF-8"?>
<ncx xmlns="http://www.daisy.org/z3986/2005/ncx/" version="2005-1">
    <docTitle>
```

```
    <text>Fixed Layout Example 2.3</text>
</docTitle>
. . .

<!--PageList-->
  <pageList>
    <pageTarget type="normal" value="1">
      <navLabel>
        <text></text>
      </navLabel>
      <content src="page001.xhtml"/>
    </pageTarget>
    <pageTarget type="normal" value="2">
      <navLabel>
        <text>2</text>
      </navLabel>
      <content src="page002.xhtml"/>
    </pageTarget>
    <pageTarget type="normal" value="3">
      <navLabel>
        <text>3</text>
      </navLabel>
      <content src="page003.xhtml"/>
    </pageTarget>
    . . .
  </pageList>
</ncx>
```

Configuring Apple Display Options

You can customize the iBooks reading experience by setting display options in the `com.apple.ibooks.display-options.xml` file, within the META-INF folder. Display options include setting page orientation, specific platforms, and fonts. For example, when a book is opened on an iPhone, you can set it so the book always opens in Landscape mode. Other display options are specific to Fixed Layout books. The following lists the supported display options for Flowing books and Fixed Layout books (which are annotated in detail below the example):

- Specific platforms: "iphone", "ipad", or "*" for all platforms
- "specified-fonts" (used only in Flowing books to indicate the book uses embedded fonts)
- "fixed-layout" (useful for precisely-designed books)
- "open-to-spread" (used only in Fixed Layout books)
- "orientation-lock" (useful for Fixed Layout books)
- "interactive" (used to indicate the book has JavaScript interactivity)

Picture Book Example

Below is a sample `com.apple.ibooks.display-options.xml` file, which is required for Fixed Layout books. In this example, the EPUB is a picture book.

```
<?xml version="1.0" encoding="UTF-8"?>
<display_options>
  <platform name="*">
    <option name="specified-fonts">true</option>
    <option name="interactive">true</option>
    <option name="fixed-layout">true</option>
    <option name="open-to-spread">true</option>
  </platform>
  <platform name="iphone">
    <option name="orientation-lock">landscape-only</option>
  </platform>
</display_options>
```

```
<display_options>
```

Display Options

Begins the display options block.

```
<platform name="*">
```

Platform

Identifies the platform(s) to which the specified display options apply.

The following values are supported for the name attribute (note that the values must be all lowercase letters):

"*" applies to all platforms

"iphone" applies only to the iPhone and includes iPod touch

"ipad" applies only to the iPad

```
<option name="specified-fonts">true</option>
```

Font Option

Indicates if embedded fonts are used for the specified platform. (Note that the option names and values must be all lowercase letters.)

A book can use system fonts (fonts that already exist on the device) or embedded fonts (fonts you provide to use within the book). Embedded fonts can be used in Flowing books.

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

To indicate the book uses embedded fonts, use the "specified-fonts" option and set the value to true.

"specified-fonts" indicates that the book contains specific, custom fonts that have been used intentionally for effect (for example, a handwritten font to convey a handwritten note, or a special font used as the drop cap (initial letter of first word of a chapter)). Allowed values: true or false.

```
<option name="interactive">true</option>
```

Interactive Option

Indicates if the book contains interactive content on the specified platform. (Note that the option names and values must be all lowercase letters.)

"interactive" indicates that the book contains JavaScript interactivity. Allowed values: true or false. If this option is set to true, iBookstore can identify the book as interactive.

```
<option name="fixed-layout">true</option>  
<option name="open-to-spread">true</option>  
<option name="orientation-lock">landscape-only</option>
```

Options for Fixed Layout Books

Describes the particular display option to apply to the specified platform. If the book has a fixed layout, you must indicate that it has a fixed layout in the `com.apple.ibooks.display-options.xml` file.

"`fixed-layout`" indicates that the Fixed Layout book has a precise layout. This option defines both how the book is presented in iBooks (full-bleed images, text over images) and also how the sample is cut. Allowed values: `true` or `false`. If not specified, `false` is the default.

"`open-to-spread`" indicates that the Fixed Layout book should open to a two-page spread. Allowed values: `true` or `false`. If not specified, `false` is the default.

"`orientation-lock`" indicates the orientation to use when the book is opened. Allowed values: `landscape-only`, `portrait-only`, or `none`. If not specified, `none` is the default and the orientation will not be locked.

```
</platform>  
</display_options>
```

Book Marketing Image

- The book marketing image (the image used on the Storefront) must be at least 1400 pixels wide (2400 pixels recommended for best results).
- The file must be a high-quality JPEG with `.jpg` extension or PNG with `.png` extension and must be RGB.
- Do not increase the size of a smaller image to meet the minimum size 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.

CSS Styles

Important 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.

In the example below, `.mystyle` will not be applied in iOS 5.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Title</title>
    <meta http-equiv="Content-Type" content="text/html;CHARSET=utf-8"/>
    . . .
    <style type="text/css">
      <!--
      .mystyle {...}
      -->
    </style>
```

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.

Images

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 in the iBookstore.
- JPEG with .jpg extension (quality unconstrained) or PNG with .png extension.
- RGB (screen standard).
- Images that have any transparent areas should be PNG format or, ideally, use JPGs with WebKit PNG masks. Images that do not have transparency should be JPGs.
- 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 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). Two million pixels is the largest image size allowed. For example, a single-page, full-bleed image should be around 1200px by 1600px.
 - In Flowing books, iBooks scales images to fit on a page no matter what the page size or orientation. This prevents an image from being clipped at a page break. To accomplish this scaling, iBooks overwrites any dimensions defined on ``. Therefore, to size images, wrap them in a container `<div>` and size that `<div>` using percentage, not pixels, to maintain adaptability to various screen sizes.

- In Fixed Layout books, developers may 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 pixels (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).

DRM

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

Screenshots

- Full-sized iPad 1/2 images (1024 x 768 or 768 x 1024) in PNG (.png) or JPEG (.jpg) format.
- 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).

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 referenced using standard CSS. Embedded fonts must be listed in the manifest of the .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 ["Configuring Apple Display Options"](#) (page 13) for details).

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 does not apply DRM to embedded font files.

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.

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 “[Audio and Video Asset Requirements](#)” (page 62).

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 23) 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 the iBookstore 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 2.2 (`flowing2-2.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>

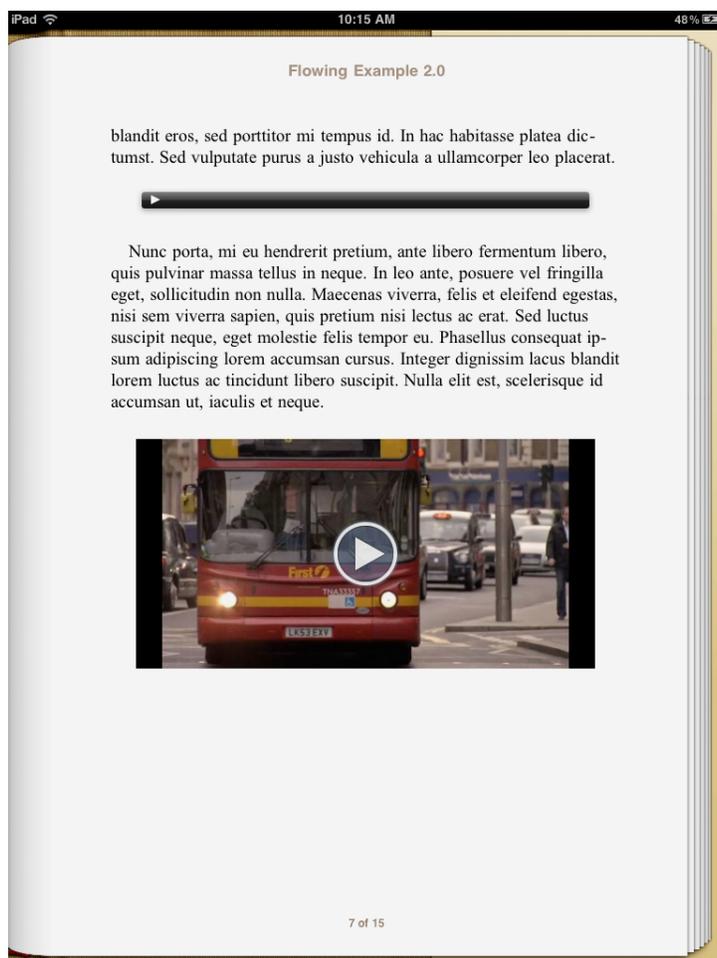
Video

<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 may be used to augment and enhance book content. Links may be within a document, as in the case of an index or endnotes, or may link to the web or other external resources such as an author or publisher website.

The iBookstore 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*.

When setting up the link, use `itms-books://` instead of `http://` at the beginning of the iBookstore URL. This takes the readers directly to the book in iBookstore without first displaying a message that they're leaving the iBooks app:

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

iBookstore will automatically include a page at the end of your book that includes links to other books by the same author.

Note iBooks supports CSS color for the anchor element if the `specified-fonts` is set to `true` in the Apple Display Options file (`com.apple.ibooks.display-options.xml`). See [“Configuring Apple Display Options”](#) (page 13) for details.

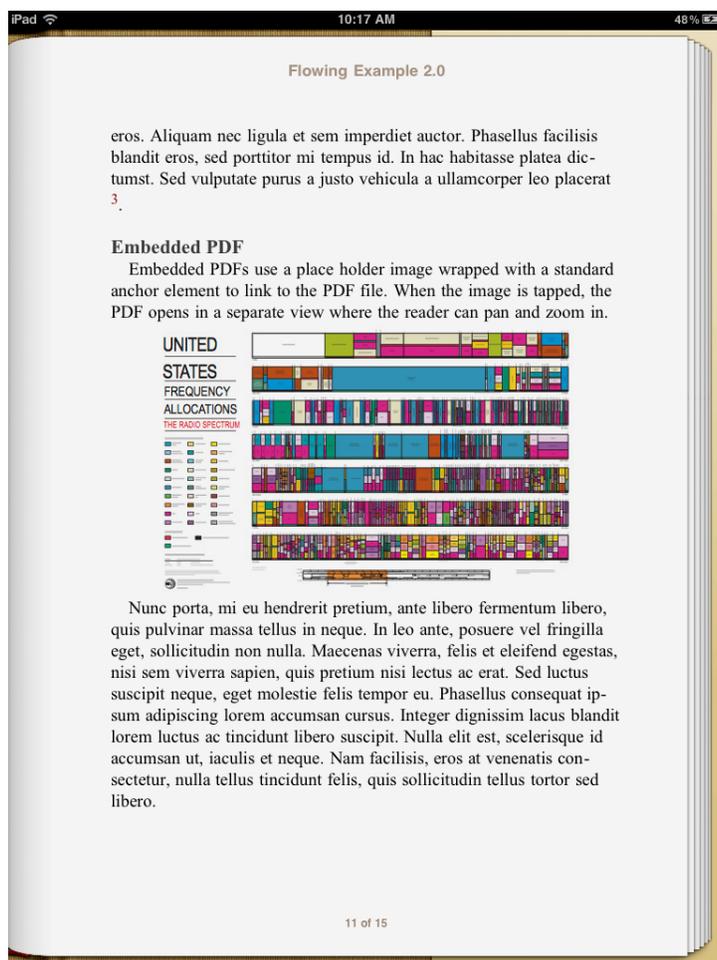
PDFs

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 adipiscing 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 both Flowing books and Fixed Layout books, but interactivity is particularly suitable for Fixed Layout books. This section provides information and tips for creating interactivity using JavaScript. Books with JavaScript interactivity require iBooks 1.5 or later and iOS 5.0.

In addition to this guide, an example book that contains interactivity ([fixedlayout2-3.epub](#)) is available for your reference in the Examples section on the Deliver Your Content page in iTunes Connect.

Note iBooks supports scripting within version 2 EPUBs. iBooks does not support EPUB 3 at this time.

Best Practices for Developing Scripts for Books with JavaScript Interactivity

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

- **Develop scripts that perform well on mobile devices.** When developing JavaScript interactivity for mobile devices, 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 to cue menus, pagination, and zoom. However, in books with JavaScript interactivity, iBooks needs to ignore gestures so that the reader can touch 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

Designing for Fingers

Interactivity should be designed for fingers, not a mouse. 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.

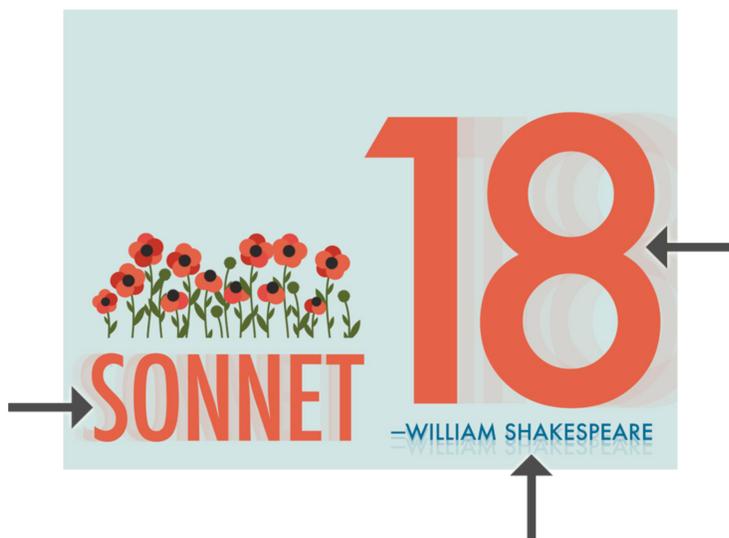
When placing interactive objects, avoid placing them close to page edges and text. Objects too close to the edge of the page may cause the reader to unintentionally turn the page. Similarly, objects placed too close to text may cause the reader to unintentionally trigger the re-reading of a word in read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

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.

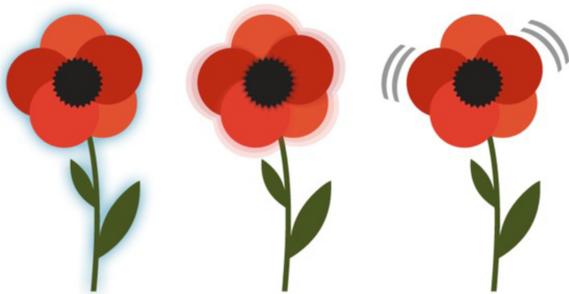


Touch to trigger interactivity:

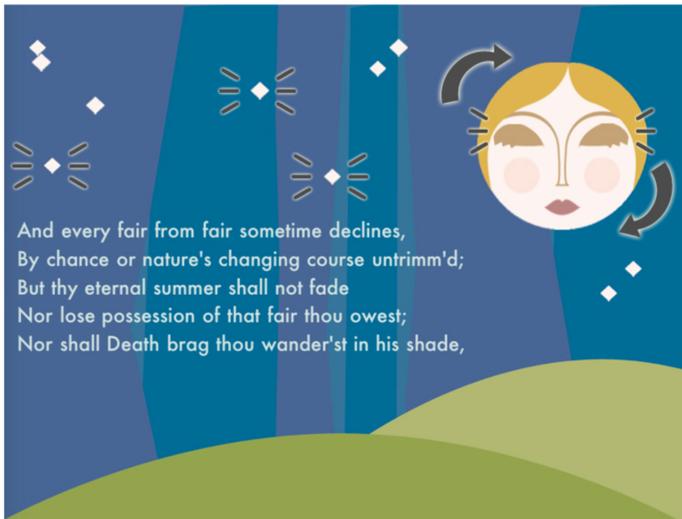
- Use bold formatting or add a glow behind words that trigger sound or animation. In the example below, touching winds triggers a wind noise.



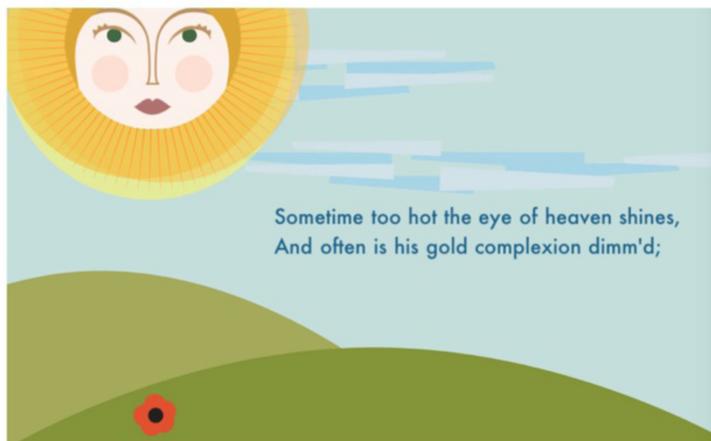
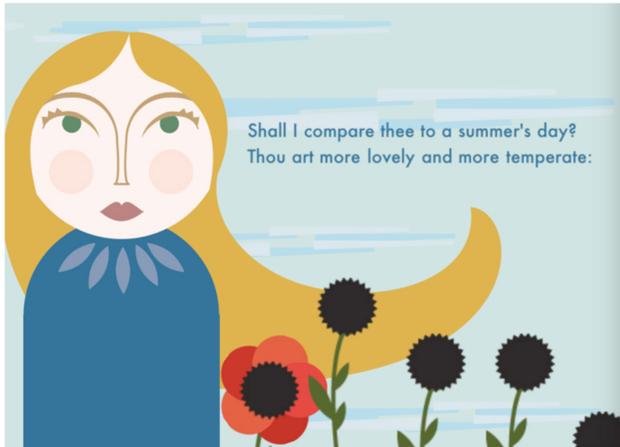
- Add a glow, pulse, or wiggle to an interactive object. In the example below, the flower on the left is glowing, the flower in the middle is pulsing, and the flower on the right is wiggling.



- 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 ([fixedlayout2-3.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 50) 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`.

Display Options for Books with JavaScript Interactivity

For the iBookstore to identify a book as containing interactivity, you must indicate in the Apple Display Options file (`com.apple.ibooks.display-options.xml`) that the book is interactive. See [“Configuring Apple Display Options”](#) (page 13) for complete details.

Using the iBook JavaScript Library

The iBook JavaScript Library (`ibook.js`) is a collection of pre-written JavaScript that make it easier to develop books with JavaScript interactivity for the iBookstore. The iBook 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 `ibook.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/ibook.js" type="text/javascript" charset="utf-8"></script>
</head>
```

When a page has loaded, `ibook.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, `ibook.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>
```

Stamp-able Elements

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

`ibook.js` relies on a SVG path to define a valid touch area; without a SVG path `ibook.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" enable-background="new 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-8.653,0-16.846,0.42-24.805,1.205v136.949h14.525,0.023l296.721,0.478l215.824,0.347  
C900.57,466.128,771.732,387.128,612.806,387.128z"/>  
</svg>  
</div>
```

Toggle-able Elements

`iBook.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 59) 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 `iBook.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  
  iBook.ACTIVE_CSS_CLASS = "active";  
  
  // CSS class name appended to body on page load  
  iBook.CSS_CLASS_ON_LOAD = "build-in";  
  
  // Delay in milliseconds before deferred events fire  
  iBook.DEFERRED_EVENT_DELAY = "1000";  
  
  // CSS selector for page  
  iBook.PAGE_CSS_SELECTOR = ".page";  
  
  // CSS class for stamped elements  
  iBook.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 iBookstore 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/elements.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

iBookstore 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 iBookstore to cut the sample automatically, you can create a separate, custom `.epub` file and deliver it for use as the sample in the iBookstore. 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 very beginning of the book. Then, the sample is cut either from the first spine item, or the percentage of the words, whichever is greater. For example, if the first spine item (as determined by text elements in the `<guide>` block of the `.opf`) 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 spine item. Any images, movies, or audio included in that percentage will be included in the sample.

Samples for Fixed Layout Books

If your book is a picture book or other Fixed Layout Book, you must indicate `"fixed-layout"` (in the `<display_options>` block in a `com.apple.ibooks.display-options.xml` file. (See ["Configuring Apple Display Options"](#) (page 13) for instructions.) The `"fixed-layout"` 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. iBookstore 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 `<guide>` block of the `.opf` to determine the start of the content of the book. For optimal samples, include `<reference type="text" ... />` to indicate the beginning of the book's content.

Samples for Books with Interactive Content

If your EPUB contains interactive content, you must indicate `"interactive"` (in the `<display_options>` block in a `com.apple.ibooks.display-options.xml` file. (See ["Configuring Apple Display Options"](#) (page 13) for instructions.)

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 EPUB is a read aloud book, you must indicate "fixed-layout" (in the `<display_options>` block in a `com.apple.ibooks.display-options.xml` file. (See "Configuring Apple Display Options" (page 13) for instructions.) Read aloud content requires a custom sample. If you do not provide a custom sample, the book won't be put in the iBookstore. 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 preview. You decide how much of your book and exactly which content to include in the sample, and then you submit the sample along with the book. We recommend you keep your sample book less than 20 MB so that it can be downloaded over 3G connections.

There are two ways of creating a sample book:

- Duplicate your book and delete chapters and pages you don't want to appear in the sample book.
- Create a new book and copy items into it from the complete book.

Important Be sure to check for broken hyperlinks and glossary terms that may link to parts of the book that you have omitted.

With the sample book open in iBooks Author, choose **File > Export**, and click **iBooks**. Give the sample a name and then click **Export**. This creates an iBooks file with the extension `.ibooks`.

Guidelines for Flowing Books

Overview

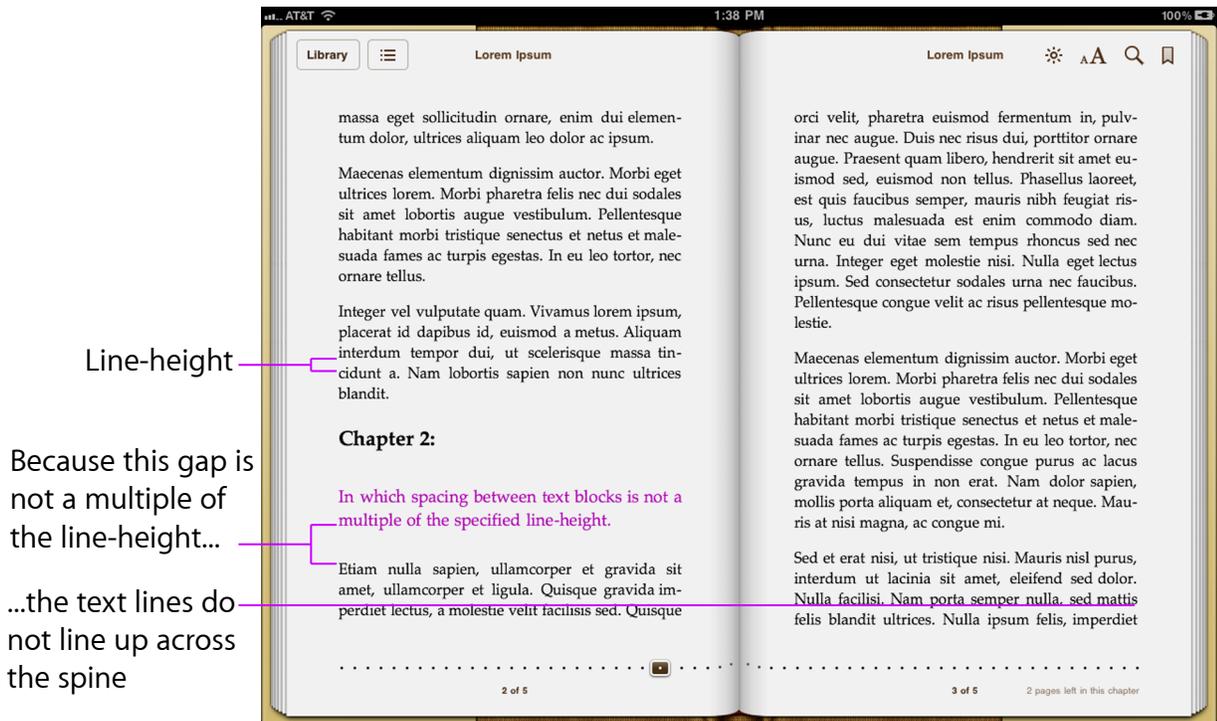
This chapter describes guidelines for creating Flowing Books. Refer to [“Digital Book Essentials”](#) (page 6) for best practices that apply to all types of books.

Content Structure

- In Flowing books, supplying the `<guide>` block of the OPF file 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.

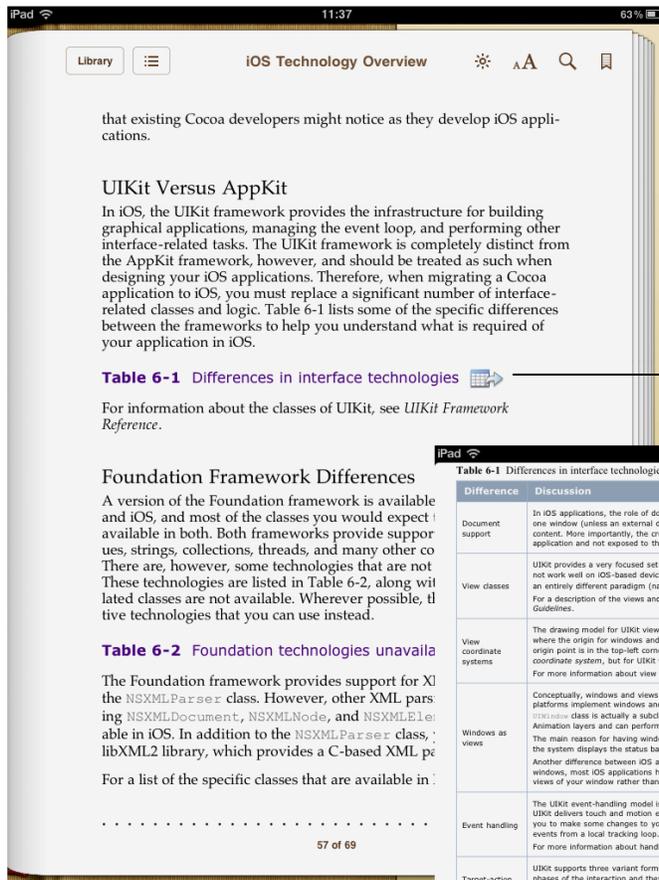


Nonlinear Content

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, `See Answer`). That link is a standard anchor element and can be associated with both text or an image.

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

that existing Cocoa developers might notice as they develop iOS applications.

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 and iOS, and most of the classes you would expect available in both. Both frameworks provide support for arrays, strings, collections, threads, and many other common technologies that are not available in iOS. These technologies are listed in Table 6-2, along with the Foundation Framework classes that are not available in iOS. Wherever possible, alternative technologies that you can use instead.

Table 6-2 Foundation technologies unavailable in iOS

The Foundation framework provides support for the `NSXMLParser` class. However, other XML parsing classes, such as `NSXMLDocument`, `NSXMLNode`, and `NSXMLElement`, are not available in iOS. In addition to the `NSXMLParser` class, the `libXML2` library, which provides a C-based XML parser, is available in iOS.

For a list of the specific classes that are available in iOS, see the *Foundation Framework Reference*.

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>NSDocument</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 coordinate system</i> , 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>NSResponder</code> , but in UIKit, the <code>UINavigationController</code> class is actually a subclass of <code>UIViewController</code> . This change in inheritance means that windows in UIKit are backed by <code>UIViewController</code> objects 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 these 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>NSTableView</code> and <code>NSOutlineView</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 (`flowing2-2.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.

Tables

Specifying tables in the `spine` using `linear=no` is no longer necessary; iBooks now 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. iBooks' new table treatment achieves a similar result to specifying `linear=no` for tables, but eliminates much of the development work required with a nonlinear treatment.

Fonts

- Font sizes should be defined in `em` or `px`, not by name, for example, `small`, `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 6) 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 28) in Chapter 2.

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 (`fixedlayout2-3.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:

- Each page must be a separate XHTML document. A two-page spread consists of two XHTML documents.
- A book's first page must begin on the right-side page of the first two-page spread. Generally, this first page is the cover page. When laying out spreads, remember that even-numbered pages will be on the left-side and odd-numbered pages will be on the right. Subsequent spreads are displayed across pages 2 and 3, 4 and 5, 6 and 7, and so on.
- 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.
- A Fixed Layout book must have the Apple Display Option `fixed-layout=true`.

- A Fixed Layout book must include a `<reference>` of `type="text"` in the `<guide>` block of the `.opf` file. The target of the `<reference>` has to be an actual HTML file and not an anchor within a file. This is to ensure proper behavior of the automatically-generated sample (automatically-generated samples are not supported for Read Aloud books).

```
<guide>
  <!-- text reference is required for fixed layout books and is used to determine
  the start of the book's main content for sample cutting purposes -->
  <reference type="cover" title="Cover" href="page001.xhtml"/>
  <reference type="text" title="Main Content" href="page002.xhtml"/>
</guide>
```

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 `toc.ncx` file (required for all EPUBs). 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, include only one `navPoint` in the `toc.ncx<navMap>` block—the thumbnail table of contents will still be available. The following shows a thumbnail table of contents:



In 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 `i, ii, iii` or `a, b, c` then `1, 2, 3` for page numbers. Using `<pageList>` in Fixed Layout books requires iBooks 2.0.

The following shows an example of `<pageList>` within an `<ncx>` block (for the sake of brevity, only three `<pageTarget>` examples are shown):

```
<?xml version="1.0" encoding="UTF-8"?>
<ncx xmlns="http://www.daisy.org/z3986/2005/ncx/" version="2005-1">
  <head>
    <meta name="dtb:uid" content="http://www.hxa7241.org/articles/content/
epup-guide_hxa7241_2007_1.epub"/>
  </head>
  <docTitle>
    <text>Fixed Layout Example 2.3</text>
  </docTitle>

<!--TOC-->
  <navMap>
    <navPoint id="p001" playOrder="1">
      <navLabel>
        <text>Animated text</text>
      </navLabel>
      <content src="page001.xhtml#word1"/>
    </navPoint>
    <navPoint id="p002" playOrder="2">
      <navLabel>
        <text>Embedded read aloud controls</text>
      </navLabel>
      <content src="page001.xhtml#readAloud"/>
    </navPoint>
    <navPoint id="p003" playOrder="3">
      <navLabel>
        <text>Looping animation</text>
      </navLabel>
      <content src="page002.xhtml#loopingAnimation"/>
    </navPoint>
    ...
  </navMap>

<!--PageList-->
  <pageList>
    <pageTarget type="normal" value="1">
      <navLabel>
        <text></text>
      </navLabel>
      <content src="page001.xhtml"/>
    </pageTarget>
```

```
<pageTarget type="normal" value="2">
  <navLabel>
    <text>i</text>
  </navLabel>
  <content src="page002.xhtml" />
</pageTarget>
<pageTarget type="normal" value="3">
  <navLabel>
    <text>ii</text>
  </navLabel>
  <content src="page003.xhtml" />
</pageTarget>
...
</pageList>
</ncx>
```

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 20), 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.

Optimizing Images in Fixed Layout Books

In addition to the image recommendations in “[Images](#)” (page 18), the following apply to working with images in Fixed Layout books:

- Images may be in PNG or JPEG 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.
- The maximum size is 2 million pixels per image.

Layout

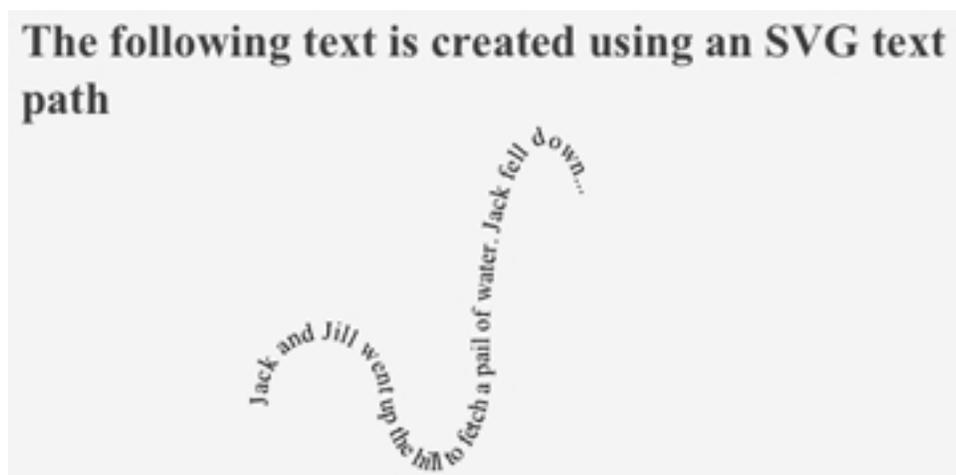
- Because iBooks scales the book to fit within the viewport, both landscape-dominant and portrait-dominant books will maintain their intended layout in either orientation of the device, even after an orientation change.

- 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 may be locked. See [“Configuring Display Options”](#) (page 61).
- 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 (`<dc:date>`) will circumvent iBooks' caching. For example:

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns="http://www.idpf.org/2007/opf">
  <dc:title>Book's Title</dc:title>
  <dc:identifier id="bookid">1234567890</dc:identifier>
  <dc:language>en</dc:language>
  <dc:date>2011-09-01</dc:date>
</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 42).

Note iBooks supports read aloud content within version 2 EPUBs. iBooks does not support EPUB 3 at this time.

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.



- Volume slider
- Turn ambient soundtrack on or off
- Page-turning options
- Start or stop read aloud

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 57).
- 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.

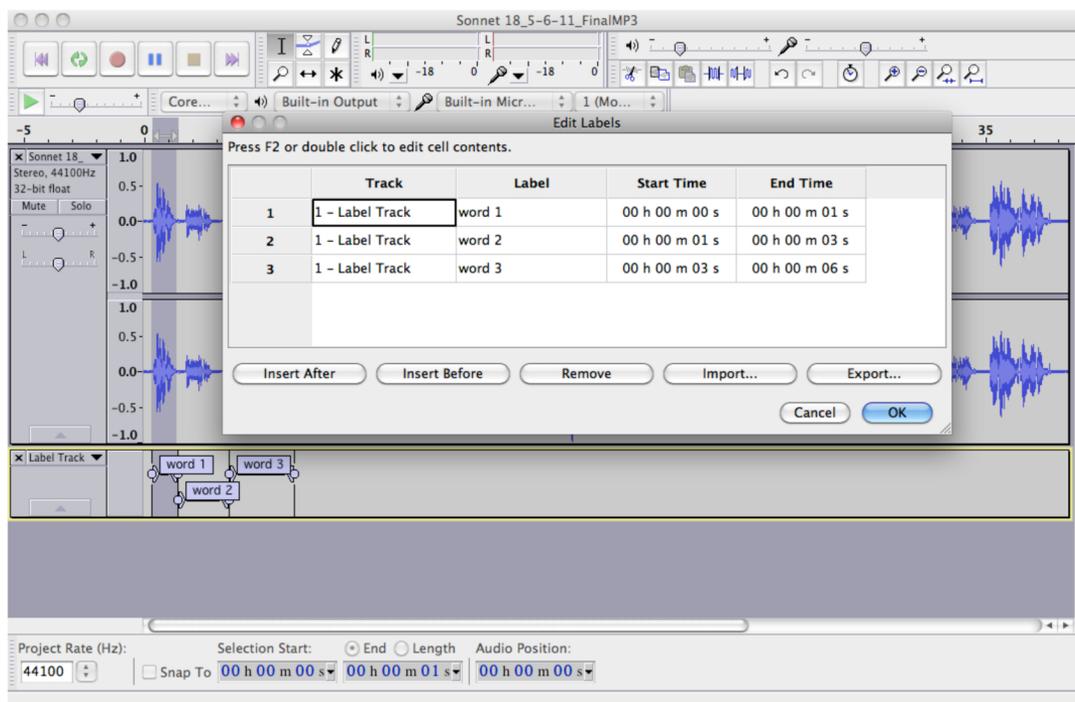
Note The following instructions show how to mark start and end times in an audio file using Audacity, a free, cross-platform audio editing tool. You can use another audio file tool to mark the start and end times in an audio file, however steps to do this will vary depending on the audio editing tool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

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 the namespace `xmlns:epub="http://www.idpf.org/2007/ops"` in `<html>`.

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">
...
  <audio epub:type="ibooks:soundtrack" src="audio/sonata14.m4a"/>
...
</html>
```

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 in `<html>`. In iBooks 1.5, you can also pause the read aloud narration.

iBooks Namespace

```
xmlns:ibooks="http://apple.com/ibooks/html-extensions"
```

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.

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://apple.com/ibooks/html-extensions`; this is typically done on the document's HTML tag.

- If you are using our iBook 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 iBook JS library:

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

Additionally, when using iBook 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.

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>
```

Configuring Display Options

You can customize the iBooks reading experience by using the display options in the `com.apple.ibooks.display-options.xml` file, within the META-INF folder. If your book has a fixed layout, you must indicate that it has a fixed layout in the `com.apple.ibooks.display-options.xml` file. You can also set display options for a specific platform, for example, when a book is opened on an iPhone, you can set it so the book always opens in Landscape mode. See [“Configuring Apple Display Options”](#) (page 13) for more information.

Guidelines for Multi-Touch Books

Overview

This chapter describes guidelines for audio and video asset requirements for Multi-Touch books. Refer to “[Digital Book Essentials](#)” (page 6) for best practices that apply to all types of 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. (For the sake of brevity in this document, the term “Multi-Touch books” includes all books created using the iBooks Author application including Multi-Touch textbooks. The term “Multi-Touch textbooks” is used when referring only to textbooks. Note that you must have an additional contract in place to deliver paid Multi-Touch textbooks to the iBookstore.) iBookstore only accepts Multi-Touch books made with the latest version of iBooks Author.

The iBooks Author support site (<http://www.apple.com/support/ibooksauthor/>) includes best practices for fonts, 3D objects, accessibility, and more.

Note Multi-Touch books require a custom sample for previews. See “[Samples for Multi-Touch Books](#)” (page 42) for details.

Audio and Video Asset Requirements

In iBooks Author, you can embed audio and video inside a book using a Media widget and an HTML widget. This section describes asset requirements for video and audio for both types of widgets. It also describes how to encode the audio and video used in a Media widget.

Important When sending books with embedded audio or video content, keep in mind that the maximum file size for the .iBooks file is 2GB. For usability's sake, the maximum recommended size is 1GB. Larger files take longer to download and may become unwieldy on older devices.

Media Widget Video and Audio Asset Requirements

- H.264 video (.m4v)
- AAC audio (.m4a) audio in movies or standalone audio files

Note Video and audio media in Media widgets can be DRM protected when submitted to the iBookstore using a Paid Books Account.

Encoding Media Widget Audio

Use iTunes to create a compatible audio file:

1. Open iTunes and choose **iTunes > Preferences**.
2. In the General pane, click the **Import Settings** button.
3. Choose **AAC Encoder** from the Import Using pop-up menu, and choose a quality level from the Setting pop-up menu. iTunes Plus is recommended.
4. Do one of the following:
 - If the audio file isn't already in iTunes: Drag the file from the Finder to the iTunes window (or in iTunes, choose **File > Add to Library**).
 - If the audio file is already in iTunes: Select the file and choose **Advanced > Create AAC Version**.

This option produces a .m4a audio file.

Encoding Media Widget Video

Encode videos for use in the Media widget using Compressor 4 or later:

1. Create a new batch in Compressor 4 or later.
2. Add your video file to a job in the batch.
3. From the Apple Devices Settings, drag one of the following settings to job:
 - HD for Apple Devices (5 Mbps) for HD video
 - SD for Apple Devices for SD video (or HD video if you are trying to reduce the book's file size)

Each of these options produces a .m4v video file.

4. Click **Submit**.

If you want to reduce the movie's file size, you can reduce the bit rate in the Encoder pane of the Inspector to a lower value. For example, lowering the bit rate from 5 Mbps to 2.5 Mbps reduces the movie's file size by half while maintaining good image quality. For more information, see the *Compressor User Guide*.

HTML Widget Video and Audio Asset Requirements

In iBooks Author, video and audio files used in HTML widgets are not DRM (digital rights management) protected. If your HTML Dashboard file includes video and audio, keep in mind that HTML widgets do not support M4V (.m4v) video files or M4P (.m4p) audio files.

- H.264 video (.mp4)
- AAC audio (.m4a) audio in movies or standalone audio files

When you use audio and video included in a Keynote widget, the Keynote HTML export process transcodes any video and audio in the presentation to H.264 (.m4v) movies.

Keynote Best Practices

- You must use the latest version of Keynote (5.1.1 at time of publication).
- Where possible, use other widget types when appropriate instead of Keynote files. For example, instead of embedding a video or audio in a Keynote file, use the Media Widget; instead of producing a slideshow in Keynote, use an image gallery if appropriate.

Revision History

Previous Spec Revisions

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

Date/Version	Summary
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 <code><spine></code> 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.
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.

Date/Version	Summary
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 the iBookstore.

Changes in the iBookstore 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 the iBookstore, use `itms-books://` instead of `http://` at the beginning of the iBookstore URL. This takes the readers directly to the book in iBookstore without first displaying a message that they're leaving the iBooks app. See “[Links](#)” (page 26) 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 42).

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 iBookstore 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 43).

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 [“Images”](#) (page 18).

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 iBook 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 59) 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 39) for details.

Changes in the iBookstore 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 52) for more information.

Changes in the iBookstore Asset Guide 4.7 Revision 3

Corrections

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

Changes in the iBookstore 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 iBookstore 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 44) 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.

Change in iBookstore 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 in the iBookstore.

Changes in iBookstore 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 [“Configuring Apple Display Options”](#) (page 13) 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 17).

Changes in iBookstore 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 the iBookstore.



Apple Inc.

© 2012 Apple Inc.

All rights reserved.

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.

The Apple logo is a trademark of Apple Inc.

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

iBookstore is a service mark of Apple Inc.

iTunes Plus is a registered service mark of Apple Inc.

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

Multi-Touch is a trademark of Apple Inc.

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

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

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.

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.

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.