Amazon Kindle Publishing Guidelines

How to make books available for the Kindle platform

version 2015.4.1

This document describes the primary ways publishers, authors, and conversion houses can make their content available on the Amazon Kindle platform. This document includes guidelines and suggestions to ensure a smooth conversion and publication process.

Copyright © 2016 Amazon.com, Inc. or its affiliates. All rights reserved.

Revision History

Revision Number	Revision Notes
2015.4.1	• Deleted 3.7.1 Table Guideline #1: Use Tables for Tabular Data Only • Deleted 3.8.2 Styling Guideline #2: Format Sidebars Correctly • Added 9.4.2 Size Images for Responsive Layouts • Updated 9.4.7 Image and Font Size Requirements for Line-Art and Text • Updated 13.7.5 Adding Images with Play Controls • Reformatted Appendix B: HTML and CSS Tags Supported in Kindle Format 8
2015.4	• Reorganized document. • Updated 2.2.2 Kindle Previewer Software • Added 2.2.3 Kindle Textbook Creator • Added 2.2.4 Kindle Comic Creator • Added 2.2.5 Kindle Kids' Book Creator • Added 3 Comparing Formats • Updated 9.3 Text Guidelines • Added 12 Creating Fixed-Format Books Without Pop-ups • Added Appendix A Enhanced Typesetting
2015.3	• Updated 2.2.1.3 Using KindleGen • Updated 3.6 Image Guidelines (and subsections 3.6.1—3.6.4) • Added 3.9.10 HTML Guideline #10: Use bi-directional hyperlinks for footnotes • Updated 6 Audio and Video Guidelines • Updated 9.1 Testing Kindle Books

Table of Contents

PART I. GETTING STARTED 8

1 Introduction 8

2 Paths to Getting Your Content on Kindle 8
2.1 Amazon's Kindle Direct Publishing Platform 8
2.2 Creating Kindle Books In-House Using Kindle Publisher Tools 8
2.2.1 KindleGen 8
2.2.2 Kindle Previewer Software 11
2.2.3 Kindle Textbook Creator 11
2.2.4 Kindle Comic Creator 12
2.2.5 Kindle Kids' Book Creator 12
2.3 Third-Party Conversion Services 12

3 Comparing Formats 13
3.1 About Kindle Format 8 15

PART II. GENERAL BEST PRACTICES 17

4 Cover Image Guidelines 17
4.1 Marketing Cover Image Is Mandatory 17
4.2 Internal Content Cover Image Is Mandatory 17

5 Navigation Guidelines 18
5.1 HTML TOC Guidelines 18
5.1.1 Using a Nested HTML TOC 19
5.2 NCX Guidelines 19
5.2.1 Creating a Logical TOC Using a toc nav Element 20
5.2.2 Creating a Logical TOC Using NCX 21
5.3 Guide Items 22
5.3.1 Defining Cover and TOC 22
5.3.2 Defining Start Reading Location 22

6 HTML and CSS Guidelines 23
6.1 Constructing Well-Formed HTML Documents (XHTML) 23
6.2 Anchors Must Be Added Before Formatting Tags 23
6.3 Avoid Using Negative Values 23
6.4 Avoid Using Scripting 23
6.5 File References Must Match Case and Spelling of Source 23
6.6 Other Encodings Are Supported 23
6.7 Use Supported Characters and Spaces 24
6.8 Design for a Good eBook Experience 24

7 External Link Guidelines 24

8 QA Standards 25
8.1.1 Testing Kindle Books 25

PART III. GUIDELINES FOR SPECIFIC TYPES OF EBOOKS 27

9 Creating Text-Heavy Fiction and Non-Fiction eBooks (Reflowable) 27
9.1 Metadata Guidelines 27
9.2 Layout Guidelines 27
9.3 Text Guidelines 27
9.3.1 Body Text Must Use Defaults 27
9.3.2 Formatting Paragraphs 28
9.3.3 Monospaced Font Is Supported 29
9.3.4 Avoid Using Fixed Values for Most Elements 29
9.3.5 Margin and Padding Formatting 29
9.3.6 Drop Caps 29
9.3.7 Use CSS for Page Breaks 30
9.3.8 Using Embedded Fonts 30
9.3.9 Customizing Font Selection 31
9.3.10 Page Number Guidelines 32
9.3.11 Footnote Guidelines 32
9.4 Image Guidelines 32
9.4.1 Use Supported Input Formats 32
9.4.2 Size Images for Responsive Layouts 33
9.4.3 KindleGen Performs Automatic Image Conversions 33
9.4.4 Use Color Images 33
9.4.5 Photographs Should Be Optimized for High-Resolution Devices 33
9.4.6 Use GIF or PNG for Line-Art and Text 33
9.4.7 Image and Font Size Requirements for Line-Art and Text 34
9.4.8 Prefer HTML to Images 36
9.4.9 Positioning Image Captions 37
9.4.10 Controlling Image Aspect Ratio 37
9.4.11 Displaying Text Correctly within SVG 37
9.4.12 Use Supported SVG Tags and Elements 38
9.5 Table Guidelines 39
9.5.1 Avoid Large Tables 39
9.5.2 Create Simple HTML Tables 40
9.5.3 Split Tables as Needed 40
9.5.4 Optimize for Maximum Table Size 41

10 Creating Children's Picture Books (Fixed-Layout with Text Pop-Ups) 41
10.1 Metadata Guidelines 41
10.2 Cover Image Guidelines: Including Back Cover for Children's Content 43
10.3 Text Guidelines: Including Specific Fonts 43
10.4 Content Requirements 43
10.4.1 Requirement #1: Using HTML File Structure 43
10.4.2 Requirement #2: Using Region Magnification (Pop-ups) 44
10.4.3 Requirement #3: Setting Images as Background Images 46
10.5 HTML and CSS Guidelines 46

10.5.1 Applying CSS Reset 46
10.5.2 Including One CSS File Per HTML Page 46
10.5.3 Optimizing Content for Full Screen 46
10.5.4 Using Large Region Magnification Tap Targets in Children's Books 46
10.5.5 Using position:absolute for Text on Image 47
10.5.6 Future-Proofing Fixed-Layout Content in Children's Books 47
10.6 Creating Children's Books with Multipage Background Images and Text 47
10.6.1 Using Side-by-Side Images to Form a Double-Page Spread When Orientation-Lock Is Set to Landscape 47
10.6.2 Positioning Text Blocks 49
10.6.3 Aligning Text 49

11 Creating Graphic Novels/Manga/Comics (Fixed-Layout) 49
11.1 Metadata Guidelines 50
11.2 Text Guidelines 51
11.3 Image Guidelines 53
11.3.1 Asset Requirements 53
11.3.2 Image Quality 53
11.4 Panel View (Region Magnification) 53
11.5 Virtual Panels in Comics and Manga 55
11.5.1 Requirement #1: Pairing Pages When Orientation-Lock Equals None 56
11.6 Optimizing Content for the Graphic Novel Experience 60
11.6.1 Optimizing Tap Targets 60
11.6.2 Optimizing View Panels 60
11.6.3 Optimizing for Wide or Tall Action Scenes 60

12 Creating Fixed-Format Books Without Pop-ups 61
12.1 Metadata Guidelines 61
12.2 Text Guidelines 62

13 Creating Kindle Edition with Audio/Video Content 63
13.1 Audio Requirements 63
13.2 Video Requirements 63
13.3 File Requirements 64
13.3.1 Multimedia Directory 64
13.3.2 Confirm Correct Mime-Type 64
13.3.3 File Size 65
13.4 Navigation Guidelines 65
13.4.1 Including an Audio and Video TOC 65
13.4.2 Including Audio and Video Assets in NCX File 65
13.5 Text Guidelines 66
13.6 Image Guidelines 66
13.7 Audio and Video Guidelines 66
13.7.1 Adding Audio 66
13.7.2 Adding Video 67
13.7.3 Audio and Video Metadata Required 67
13.7.4 Providing Descriptive Media Captions 68
13.7.5 Adding Images with Play Controls 68

13.8 Custom Sample File Required 68

14 Creating Dictionaries 69
14.1 Metadata Guidelines 69
14.2 Text Guidelines 70
14.2.1 Dictionary Entry Template 70
14.3 Basic Dictionary HTML 70
14.3.1 Format 70
14.3.2 Frameset element 70
14.3.3 Headword Index 72
14.4 Inflections for Dictionaries 73
14.4.1 Inflection Index 73
14.4.2 Exact-match parameter 75
14.5 Building a Dictionary with Kindlegen 75
14.6 QA Standards—Testing Kindle Dictionaries 75
14.6.1 Format Testing 75
14.6.2 Lookup Testing 76

PART IV. APPENDICES 77

15 Appendix A: Enhanced Typesetting 77
15.1 About Enhanced Typesetting 77
15.2 Support for Enhanced Typesetting 77

16 Appendix B: HTML and CSS Tags Supported in Kindle Format 8 78
16.1 HTML Support Table 78
16.2 CSS Support Table 81

17 Appendix C: Media Queries 85
17.1 Media Query Guidelines 85
17.1.1 Use Correct CSS Syntax 85
17.1.2 Add a CSS Comment Before Each Media Query 86
17.1.3 Always Use Non-Media Query Code To Target E-Ink Devices 86
17.1.4 Media Queries Should Appear After Non-Media Query Code 86
17.1.5 Avoid Duplicating Code 87
17.2 Using Media Queries 88
17.3 Using Media Queries for Backward Compatibility With Mobi 90
17.3.1 Submitting a Media Query 91
17.3.2 Using the display:none Property with Media Queries 93

18 Appendix D: Guidelines for Converting XMDF to KF8 97
18.1 Kindlegen Command 97
18.2 Source Issues 97
18.2.1 Poor Image Quality 97
18.2.2 Bad quality images will result in the cover or other image being displayed too small. To avoid this, provide good quality images. For quality requirements, see section 4 Cover Image Guidelines and section 9.3.12 Sidebar Guidelines 97
18.2.3 Blurred Gaiji characters 97

18.2.4 TOC Entries Hanging Style 97
18.2.5 TOC Entries Not Linked 98
18.2.6 Incorrect Number Orientation in TOC 98
18.2.7 Bold Kanji Characters 98
18.2.8 Faded Text 98
18.2.9 No Space Between Images 99
18.2.10 Images Not Shown In Separate Pages 99
18.2.11 TOC Not Shown 99
18.2.12 Large Amount of Text Is Not Centered 99
18.2.13 Duplicate IDs 99
18.2.14 File Path and File Name Requirements 99
18.3 Unsupported Features 99
18.3.1 Ignored Features 101

Part I. Getting Started

1 Introduction

There are several options for making your books available for the Amazon Kindle platform. Which option is best for you depends upon the nature of your publications (such as your source file format), your available resources and technical expertise, and your eBook sales model. To help you choose, here are examples of common publishing scenarios and recommendations:

- For self-publishers or authors who would like to take advantage of Amazon's self-service tools to create Kindle books and sell them on Amazon, see section 2.1, Amazon's Kindle Direct Publishing Platform.
- For publishers with many titles to convert and the expertise to create Kindle books in-house using Kindle Publisher Tools software, see section 2.2, Creating Kindle Books In-House Using Kindle Publisher Tools.
- For publishers who do not wish to convert titles in-house or do not have the technical expertise to do so, outsourcing to a conversion house is described in section 2.3, Third-Party Conversion Services.

2 Paths to Getting Your Content on Kindle

2.1 Amazon's Kindle Direct Publishing Platform

Self-publishers can convert books into electronic format using Amazon's self-publishing tools and sell them on Amazon Kindle with Amazon's Kindle Direct Publishing Platform (KDP). KDP is a fast, easy self-publishing system for Amazon Kindle. Upload your content, enter sales copy and pricing information, and publish in minutes. To learn more or sign up, visit http://kdp.amazon.com.

2.2 Creating Kindle Books In-House Using Kindle Publisher Tools

Publishers can create Kindle books in-house from HTML, XHTML, and EPUB files by using the Kindle Publisher tools. Amazon officially supports these tools to convert files to Kindle Format 8. Kindle files created with these tools are designed to be compatible with current and future Kindle devices and applications. Files created with third-party software may not work properly on current or future Kindle devices and applications.

2.2.1 KindleGen

Publishers can create Kindle books in-house by using a free software program called KindleGen. This is a command line tool designed for building Kindle books. KindleGen accepts source content in HTML, XHTML, or EPUB format.

The most recent version of KindleGen can be downloaded for free from www.amazon.com/kindleformat/kindlegen. Amazon periodically releases new versions of the KindleGen software. Check the site regularly for updates.

2.2.1.1 Source Files to Use with KindleGen

To create Kindle files using KindleGen, you need:

- A single HTML file that represents the entire book; or

- EPUB-compliant files. (IDPF's EPUB spec is available at http://idpf.org/EPUB/30/spec/EPUB30-overview.html)

Using the EPUB spec, you can create a Kindle book with multiple HTML files and a single OPF file that links all of them together.

2.2.1.2 Installing KindleGen

IMPORTANT: Follow these steps to run KindleGen. Double-clicking the KindleGen icon does not launch this program.

KindleGen for Windows (XP, Vista, 7)

1. Download the KindleGen zip file from www.amazon.com/kindleformat/kindlegen to the desktop.
2. Right-click the zip file, select **Extract All,** and enter the folder name as **c:\Kindlegen**.
3. Open a command prompt by selecting **Start menu > All Programs > Accessories > Command Prompt.**
4. Type **c:\KindleGen\kindlegen**. Instructions on how to run KindleGen are displayed.

Conversion Example: To convert a file called **book.html**, use change directory (cd) to go to the directory where the book is located, for example **cd desktop**, and type **c:\KindleGen\kindlegen book.html**. If the conversion was successful, a new file called **book.mobi** displays on the desktop.

KindleGen for Linux 2.6 i386 or higher

1. Download the KindleGen **tar.gz** from www.amazon.com/kindleformat/kindlegen to a location such as the home (~) directory.
2. Extract it to **~/KindleGen**.
3. Open a command prompt and type **~/KindleGen/kindlegen**. Instructions on how to run KindleGen are displayed.

Conversion Example: To convert a file called **book.html**, use change directory (cd) to go to the directory where the book is located, for example, **cd desktop**, and type **~/KindleGen/kindlegen book.html**. If the conversion was successful, a new file called **book.mobi** displays on the desktop.

KindleGen for Mac OS 10.5 and above i386

1. Download **KindleGen.zip** from www.amazon.com/kindleformat/kindlegen. By default, the file is downloaded in the **Downloads** folder
2. Unzip the file. In Safari, the zip file is automatically unzipped after download. If this setting is disabled or if another browser was used, double-click the downloaded file to unzip it.
3. Click the spotlight icon in the top right corner and type **Terminal**. Click the application to open it.
4. To view the instructions on how to run KindleGen, locate the **kindlegen** program in the **Finder** window. Click and drag it to **Terminal** window where the cursor is. The cursor writes in the path and moves to the end of the line. Press **Enter** to view the instructions.

- Alternatively, view the instructions by typing the command **cd ~/Downloads/KindleGen_Mac_i386_v2** in **Terminal** and then typing the command **kindlegen**.

Conversion Example:

1. To convert a file called **book.html**, copy **book.html** to the desktop.
2. In the **Finder** window, locate the **kindlegen** program. Click and drag it to the **Terminal** window, and drop it where the cursor is. The cursor inserts the path automatically and moves to the end of that line.
3. In the **Finder** window, locate the document. Click and drag it to the **Terminal** window, and drop it where the cursor is. The cursor writes in the path and moves to the end of the line. Press **Enter**. If the conversion was successful, a new file called **book.mobi** displays on the desktop.
    - Alternatively, convert the file by typing the command **cd ~/Downloads/KindleGen_Mac_i386_v2** in **Terminal** and then typing the command **kindlegen ~/Desktop/book.html**. If the conversion was successful, a new file called **book.mobi** displays on the desktop.

2.2.1.3 Using KindleGen

To convert an EPUB or HTML book to the Kindle Format 8, use KindleGen version 2 or higher as described below:

```
kindlegen filename.opf/.htm/.html/.epub [-c0 or -c1 or c2] [-verbose] [-
western] [-o <file name>]

Note:

   zip formats are supported for XMDF sources

   directory formats are supported for XMDF sources

Options:

   -c0: no compression

   -c1: standard DOC compression

   -c2: Kindle huffdic compression (recommended for large, text-heavy files)

   -o <file name>: Specifies the output file name. Output file will be
created in the same directory as that of input file. <file name> should not
contain directory path.

   -verbose: provides more information during ebook conversion

   -western: force build of Windows-1252 book

   -releasenotes: display release notes

   -gif: images are converted to GIF format (no JPEG in the book)

   -locale <locale option>: To display messages in the selected language.
```

```
en: English

de: German

fr: French

it: Italian

es: Spanish

zh: Chinese

ja: Japanese

pt: Portuguese

ru: Russian

nl: Dutch
```

2.2.1.4 KindleGen Messages

As conversion progresses, KindleGen displays detailed informational messages. If KindleGen encounters issues while converting a file, it displays a warning or error.

Amazon strongly recommends fixing all KindleGen warnings and errors before publishing a Kindle book.

- If KindleGen displays an error, it will not create the .mobi file. Errors impair the readability of the book in the Kindle Reader.
- If KindleGen displays a warning, it will create the .mobi file with an attempt to automatically fix the issue described in the warning. Amazon does not guarantee the results of any KindleGen automatic fix, which could lead to the Kindle book displaying differently than designed.

2.2.2 Kindle Previewer Software

Kindle Previewer is graphical user interface tool that imitates how books display on Kindle devices and applications. Kindle Previewer makes it easy to preview the layout of a book and make sure its text displays properly for any orientation or font size. To produce the highest quality Kindle books, Amazon recommends this tool in combination with KindleGen.

Kindle Previewer is available for the Windows and Mac OS X platforms. The most recent version of Kindle Previewer can be downloaded for free from www.amazon.com/kindleformat/kindlepreviewer. Installation and help documentation can be found at http://kindlepreviewer.s3.amazonaws.com/UserGuide.pdf.

You can also try the new Kindle Previewer 3 Beta, which supports preview of Amazon's recently launched Enhanced Typesetting features, such as typographically correct drop caps and improved spacing between words through hyphenation, kerning, and ligatures. Users are also able to preview books on a Kindle for iOS skin. You can download the beta version of Kindle Previewer 3 Beta from www.amazon.com/kindleformat/kindlepreviewer.

2.2.3 Kindle Textbook Creator

Kindle Textbook Creator is a free tool designed to help authors and educators create, preview, and publish Kindle textbooks and supplemental educational materials without having to know any HTML or CSS. In a few steps, you can import your PDF content into Kindle Textbook Creator and then add supplemental audio, video, and image pop-ups that enhance the student learning experience. You can

use the built in previewer to see how your book appears on a range of Kindle devices and free Kindle reading apps.

Once your book is ready, you can export your book and upload it to Kindle Direct Publishing (KDP) to make it available to customers.

Kindle Textbook Creator is available for the Windows and Mac OS X platforms. The most recent version can be downloaded for free from www.amazon.com/ktc.

2.2.4 Kindle Comic Creator

Kindle Comic Creator is a free tool for authors and publishers to turn graphic novels, comics and manga into Kindle books. Kindle Comic Creator makes it easy to import original artwork, optimize the experience of readers, and preview how a book will look on Kindle devices.

Kindle Comic Creator accepts most common types of graphic files, so authors are free to create artwork using their preferred design tools. Kindle Comic Creator accepts single or multi-page source files in .pdf, .jpg/.jpeg, .tif/.tiff, .ppm, or .png formats.

Kindle Comic Creator is available for the Windows and Mac OS X platforms. The most recent version can be downloaded for free from www.amazon.com/kc2.

2.2.5 Kindle Kids' Book Creator

Kindle Kids' Book Creator is a free tool that authors and publishers can use to turn illustrated kids' books into Kindle books. This tool makes it easy to import original artwork, optimize the experience of readers, and preview how a book will look on Kindle devices and applications.

Kindle Kids' Book Creator accepts most common types of graphic files, so authors are free to create artwork using their preferred design tools.

Kindle Kids' Book is available for the Windows and Mac OS X platforms. The most recent version can be downloaded for free from www.amazon.com/kidsbookcreator.

2.3 Third-Party Conversion Services

Publishers have the option to outsource conversion of titles from a variety of formats to eBook formats. Conversion houses offer publishers solutions and services that include taking a variety of input formats and creating eBook or print-ready output. The typical input formats are:

- Word (.DOC, .DOCX), Rich Text Format (.rtf), Text (.txt)
- PDF
- Scan of print book
- FrameMaker, InDesign, PageMaker, QuarkXPress
- XML (such as DocBook, etc.)
- HTML, XHTML
- EPUB (also known as IDPF or OEB)

The process of converting non-reflowable content (such as PDF or scans) to reflowable content is labor-intensive and requires specialized formatting knowledge.

As you explore conversion house options, Amazon recommends that you confirm which source format(s) the conversion house requires to convert files for use on Kindle.

The preferred outputs from conversion houses to be processed by Amazon are:

- Books in Amazon Kindle Format (.mobi/.prc)
- Metadata in ONIX format (XML)

Amazon can also process content in EPUB source format. KindleGen compiles the file and runs checks for common errors. Any errors or warnings will prevent the titles from becoming available. These errors must be fixed in the EPUB file before the title is published in the Kindle store. Titles in EPUB format must be tested on Amazon software and/or hardware and must abide by the publishing guidelines in this document.

Conversion houses can be of service in helping publishers supply eBook retailers with metadata. Search the web for “eBook conversion” to find a list of partners to work with.

3 Comparing Formats

To determine how to convert a print book for the best Kindle experience, key elements in the source are identified and evaluated against the different conversion formats. Some formats are specifically designed for certain types of books (e.g., panel view for comics), but more complex books must be analyzed to determine the best fit for the title. Only one format can be used for each Kindle book.

This chart compares our most common Kindle book formats.

Conversion Format	Best Suited To	Key Features	Supported Devices	Limitations	Guidelines
Reflowable	Text-heavy titles	Adjustable orientation Adjustable font settings Dictionary look-up Highlighting Word search X-Ray	All Kindle devices & applications	Some complex layouts may be difficult or impossible to replicate	Section 9: Creating Text-Heavy Fiction and Non-Fiction eBooks (Reflowable)
Fixed-Layout with Text Pop-ups	Children's picture books	Fixed layout Text pop-ups	Fire tablets Kindle for Android Kindle for iOS Kindle Cloud Reader	No text selection, dictionary, or user font settings	Section 10: Creating Children’s Picture Books (Fixed-Layout with Text Pop-Ups)

Conversion Format	Best Suited To	Key Features	Supported Devices	Limitations	Guidelines
Panel View	Comics Graphic novels	Fixed layout Image pop-ups	Kindle e Ink (3rd generation & later) Fire tablets Kindle for Android Kindle for iOS Kindle Cloud Reader	No text selection, dictionary, or user font settings	Section 11: Creating Graphic Novels/Manga/Comics (Fixed-Layout)
Manga	Manga	Adjustable orientation Virtual panels Double-page spreads (in landscape) Pinch-to-zoom	Fire tablets (2nd generation & later) Kindle e Ink (touch screen models only) Kindle for Android Kindle for iOS	No text selection, dictionary, or user font settings	Section 11: Creating Graphic Novels/Manga/Comics (Fixed-Layout)
Fixed-Format without Pop-ups	Image-heavy books with large text	Fixed layout Dictionary look-up Highlighting Word search X-Ray	Kindle e Ink (text selectability features are not supported) Fire tablets Kindle for Android Kindle for iOS Kindle Cloud Reader	Only for use on books with type large enough to be read on all devices without magnification No user font settings	Section 12: Creating Fixed-Format Books Without Pop-ups

Conversion Format	Best Suited To	Key Features	Supported Devices	Limitations	Guidelines
Kindle Edition with Audio/Video	Text-heavy titles with audio and/or video content	Adjustable orientation Adjustable text settings Dictionary look-up Highlighting Word search X-Ray Inline audio & video	Fire tablets (2nd generation and later) Kindle for iOS	KF8 features are not currently supported in Kindle Edition with Audio/Video content (see sections 13.5 & 13.6)	Section 13: Creating Kindle Edition with Audio/Video Content

3.1 About Kindle Format 8

Kindle Format 8 (KF8) is our current file format (replacing Mobi 7) for Kindle books and supports additional HTML 5 and CSS 3. The following table outlines Kindle Format 8 features and device/application support:

Features & Benefits	All Kindle Devices and Apps except 1st & 2nd Generation Kindles and Kindle DX
Full support for CSS to enable publishers to control all elements of the text layout, including line spacing, alignment, justification, margin, color, style, & border.	Yes
Support for drop cap characters at the beginning of paragraphs.	Yes
Support for floating elements that includes boxed text, callouts, sidebars, & images with text wrapping.	Yes
Support for numbered and bulleted lists.	Yes
Support for nested tables and merged cells required by technical and textbooks.	Yes
Support for background images on pages and for text on background images.	Yes

Features & Benefits	**All Kindle Devices and Apps except 1st & 2nd Generation Kindles and Kindle DX**
Support for Scalable Vector Graphics (SVG) that can be zoomed without loss of fidelity.	Yes
Support for embedded fonts that allows publishers to have a custom look & feel for the book.	Yes
Support for rounded corners of boxed elements.	Yes
Support for drop shadow.	Yes
Support for outline text.	Yes
Support for multiple and repeated background images.	Yes
Support for color gradient.	Yes
Enables fine-grained control of attributes for text and other elements through CSS selectors.	Yes
Support for fixed-layout pages for specified screen sizes.	Yes
Support for Chinese, Japanese, and Korean Text	Kindle devices newer than and including the Paperwhite and Fire HD support right to left vertical scripts and Japanese Ruby script. Older e Ink and tablet devices are limited to horizontal left to right Chinese, Japanese, and Korean text.

Part II. General Best Practices

The guidelines in this section are best practices for creating Kindle books.

4 Cover Image Guidelines

4.1 Marketing Cover Image Is Mandatory

Kindle books must have a marketing cover image provided for use on the website detail page. The preferred format for the marketing cover is an image of 2560 pixels on the longest side and 1600 pixels on the shortest side with 350 ppi to ensure image clarity on Kindle HDX devices. The image file size should be 5MB or smaller.

If the marketing cover image size is smaller than the 2560 x 1600 recommendation, a reminder message is displayed at time of upload. Covers with less than 500 pixels on the shortest side are not displayed on the website.

If your cover image is smaller than the recommended size, Amazon strongly recommends that you create a new image that meets the size requirements. Do not stretch the image to meet the size requirements, because this may lower the image quality.

The content of the cover image must not:

- Infringe another publisher's or artist's copyright on the same cover.
- Mention pricing or other temporary promotional offers.

4.2 Internal Content Cover Image Is Mandatory

Kindle books must have an internal cover image provided for use within the book content. Provide a large, high-resolution cover, because Amazon quality assurance will fail the book if the cover is too small.

Do not add cover images to the content in any way other than those described in this section or the cover might appear twice in the book.

Define covers in the OPF file using either of the following methods (underlined elements are mandatory):

Method 1 (preferred):

```
<manifest>

...

<item id="cimage" media-type="image/jpeg" href="other_cover.jpg" properties="cover-image"/>

...

</manifest>
```

This syntax is part of IDPF 3.0 standard and described at http://idpf.org/epub/30/spec/epub30-publications-20111011.html#sec-item-property-values.

Method 2:

```
<metadata>

...

<meta name="cover" content="my-cover-image" />

...

</metadata>

...

<manifest>

...

<item href="MyCoverImage.jpg" id="my-cover-image" media-type="image/jpeg" />

...

</manifest>
```

This syntax is not part of the IDPF standard. However, it was designed with help from the IDPF and will validate in an IDPF validator.

5 Navigation Guidelines

Amazon strongly recommends the use of an HTML table of contents (TOC) for all books that would benefit from this navigation feature. This applies to most books, but is optional for most fixed-layout children's books (see section 10) and fixed-layout graphic novels/manga/comics (see section 11).

Amazon requires that all Kindle books include a logical TOC. The logical TOC is very important for a good reading experience, because it allows a reader to navigate between chapters easily. Users expect to see an HTML TOC when paging through a book from the beginning, while the logical TOC is an additional way for users to navigate books. The inclusion of a logical TOC is especially important for books that are longer than 20 pages.

Important: Nested anchor tags are not supported in fixed-layout books. Fixed-layout books with nested anchor tags will be suppressed.

5.1 HTML TOC Guidelines

Place the HTML TOC towards the beginning of the book and not at the end of the book. This ensures that a customer paging through the book from the beginning encounters the TOC naturally. Incorrect placement of the TOC affects the accuracy of the “Last Page Read” feature. Correct placement ensures that the TOC appears in sample downloads of the book.

The entries in the TOC must be HTML links so that users can click to go to a specific location. A table of contents that is not made of links is not useful on Kindle.

Do not create a TOC using HTML `<table>` tags. Tables are for tabular data only, not for layout.

Do not use page numbers in the TOC. Kindle books do not always map directly to page numbers in physical editions of the book.

If you are importing the document from Word, use the “Heading” styles and the "Table of Contents" feature of Microsoft Word. The TOC created by Word will be imported correctly and will convert to a TOC that follows these guidelines.

For bundled editions containing more than one individual book, include an overarching TOC at the beginning of the file.

5.1.1 Using a Nested HTML TOC

To create useful and navigable nested TOC entries, Amazon recommends using the following syntax in the HTML TOC. The examples below show two ways of writing the same sample code: style attributes and CSS classes.

Using style attributes:

```
<div>Section 1</div>
<div style="margin-left:1em;">Chapter 1</div>
<div style="margin-left:1em;">Chapter 2</div>
<div style="margin-left:1em;">Chapter 3</div>
<div style="margin-left:2em;">Subchapter 1</div>
<div style="margin-left:2em;">Subchapter 2</div>
<div style="margin-left:1em;">Chapter 4</div>
<div style="margin-left:2em;">Subchapter 1</div>
<div>Section 2</div>
...
```

Using CSS classes:

```
<style>
div.chapter { margin-left: 1em}
div.subchapter { margin-left: 2em}
</style>

<div>Section 1</div>
<div class="chapter">Chapter 1</div>
<div class="chapter">Chapter 2</div>
<div class="chapter">Chapter 3</div>
<div class="subchapter">Subchapter 1</div>
<div class="subchapter">Subchapter 2</div>
<div class="chapter">Chapter 4</div>
<div class="subchapter">Subchapter 1</div>
<div>Section 2</div>
...
```

5.2 NCX Guidelines

Logical TOCs are generated using `toc nav` elements or a navigational control file for XML application (NCX). Creating a logical TOC exposes the hierarchical structure of a Kindle book and allows the user to navigate through it. The inclusion of a logical TOC is especially important for books that are longer than 20 pages.

In logical TOC-enabled books, users can see where they are in the book because the part, chapter, or section is exposed. This progress indicator also shows relative progress through the book.

For guidance on creating a logical TOC using a `toc nav` element, see section 5.2.1.

For guidance on creating a logical TOC using NCX, see section 5.2.2.

5.2.1 Creating a Logical TOC Using a toc nav Element

The `toc nav` element is part of the IDPF 3.0 specification and is described at http://idpf.org/epub/30/spec/epub30-contentdocs-20111011.html#sec-xhtml-nav-def-model and http://idpf.org/epub/30/spec/epub30-contentdocs-20111011.html#sec-xhtml-nav-def-types-toc.

Creating a `toc nav` element provides both a logical TOC and an HTML TOC.

Example:

```
<nav epub:type="toc">

<ol>

<li><a href="Sway_body.html#preface_1">AUTHOR'S NOTE</a></li>

<li><a href="Sway_body.html#part_1">PART ONE</a>

  <ol>

  <li><a href="Sway_body.html#chapter_1">THE HOUSES, 1969</a></li>

  <li><a href="Sway_body.html#chapter_2">ROCK AND ROLL, 1962</a></li>

  <li><a href="Sway_body.html#chapter_3">THE EMPRESS, 1928–1947</a></li>

  </ol>

</li>

</ol>

</nav>
```

The example above defines the following TOC hierarchy:

AUTHOR'S NOTE
PART ONE
    THE HOUSES, 1969
    ROCK AND ROLL, 1962
    THE EMPRESS, 1928–1947

This excerpt from the OPF (publication header file) shows how to declare the `toc nav` element in the `<manifest>`:

Example:

```
<manifest>

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

Using it in the <spine> is optional if it will be used as the HTML TOC.

Example:

```
<spine>

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

5.2.2 Creating a Logical TOC Using NCX

NCX is part of the IDPF 2.0 specification and is described at http://www.niso.org/workrooms/daisy/Z39-86-2005.html#NCX.

NCX Example:

```
<navMap>

<navPoint class="titlepage" id="L1T" playOrder="1">

<navLabel><text>AUTHOR'S NOTE</text></navLabel>

<content src="Sway_body.html#preface_1" />

</navPoint>

<navPoint class="book" id="level1-book1" playOrder="2">

<navLabel><text>PART ONE</text></navLabel>

<content src="Sway_body.html#part_1" />

<navPoint class="chapter" id="level2-book1chap01" playOrder="3">

<navLabel><text>THE HOUSES, 1969</text></navLabel>

<content src="Sway_body.html#chapter_1" />

</navPoint>

<navPoint class="chapter" id="level2-book1chap02" playOrder="4">

<navLabel><text>ROCK AND ROLL, 1962</text></navLabel>

<content src="Sway_body.html#chapter_2" />

</navPoint>

<navPoint class="chapter" id="level2-book1chap03" playOrder="5">

<navLabel><text>THE EMPRESS, 1928–1947</text></navLabel>

<content src="Sway_body.html#chapter_3" />

</navPoint>

</navPoint>

</navMap>
```

The NCX example above defines the following TOC hierarchy:

AUTHOR'S NOTE
PART ONE
THE HOUSES, 1969
ROCK AND ROLL, 1962
THE EMPRESS, 1928–1947

This excerpt from the OPF (publication header file) shows how to add an NCX table of contents to a book.

Declare the NCX in the `<manifest>`:

```
<manifest>

<item id="toc" media-type="application/x-dtbncx+xml"

      href="toc.ncx"/>
```

And reference it in the `<spine>`:

```
<spine toc="toc">
```

5.3 Guide Items

Guide items are an optional feature in the EPUB format but are highly recommended. Kindle provides support for the cover, TOC, and start reading location (”Go to Beginning”) guide items. If you choose not to include guide items for the cover and TOC, these list items will still appear in the Kindle menus, but will be grayed out and not selectable.

5.3.1 Defining Cover and TOC

The Kindle platform supports both `landmarks nav` elements and guide items for defining the cover and table of contents (TOC). These elements serve to supplement the TOC and should not be used in place of one.

The `landmarks nav` elements are part of the IDPF 3.0 specification and are described at:

http://idpf.org/epub/30/spec/epub30-contentdocs-20111011.html#sec-xhtml-nav-def-model and

http://idpf.org/epub/30/spec/epub30-contentdocs-20111011.html#sec-xhtml-nav-def-types-landmarks.

Here is an example of a guide item for a TOC (underlined elements are mandatory):

```
<guide> <reference type="toc" title="Table of Contents" href="toc.html"/> </guide>
```

Here is an example of a `landmarks nav` element for a TOC (underlined elements are mandatory):

```
<nav epub:type="landmarks">

<ol><li><a epub:type="toc" href="toc.html">Table of Contents</a></li></ol>

</nav>
```

5.3.2 Defining Start Reading Location

The Kindle platform supports `landmarks nav` elements and guide items for defining the start reading location (”Go to Beginning”) in reflowable books. Do not set the start reading location to a blank page.

In Kindle fixed-format books, the OPF file should not include the start reading location guide item. Amazon now sets this guide item to the JPEG cover for Kindle fixed-format books.

6 HTML and CSS Guidelines

6.1 Constructing Well-Formed HTML Documents (XHTML)

Kindle Format 8 supports most HTML 5.0 features, although the following HTML features are not fully supported: forms, frames, and JavaScript.

When creating source HTML or XHTML for the Kindle, refer to one of the following books as a primer on constructing well-formed HTML documents:

- *HTML, XHTML, and CSS* by Elizabeth Castro (published by Peachpit Press): http://www.amazon.com/HTML-XHTML-and-CSS/dp/B000SEFC5Q
- *Beginning HTML with CSS and XHTML: Modern Guide and Reference* by David Schultz and Craig Cook (published by Apress): http://www.amazon.com/Beginning-HTML-CSS-XHTML-Reference/dp/B001D25ZPE
- *Beginning Web Programming with HTML, XHTML, and CSS* by John Duckett (published by Wrox): http://www.amazon.com/Beginning-Programming-HTML-XHTML-ebook/dp/B000VZQVVG

6.2 Anchors Must Be Added Before Formatting Tags

Correct: `<a name="Chapter1"/><h1>Chapter 1</h1>`

Incorrect: `<h1><a name="Chapter1"/>Chapter 1</h1>`

6.3 Avoid Using Negative Values

Avoid using negative values for positioning text and margins. Positioning with negative values without adding padding for compensation can cause content to display with the edge cut off. For example, if you want to use text-indent: -2em you also need to apply padding-left: 2em.

Do not use negative values for the line-height attribute. They are not supported.

6.4 Avoid Using Scripting

Scripting is not supported. All scripts are stripped from the source during conversion. SVG with animation is not supported.

6.5 File References Must Match Case and Spelling of Source

Per WC3 HTML standards, all file references (fonts, images, etc.) must match the case and spelling of the name of the source file exactly. (**Example:** “audiovideo/ThisFile.mp4” is different from “audiovideo/Thisfile.mp4”.)

To indicate a file in a directory, use “/” characters and not “\” characters. (**Example:** “multimedia/ThisFile.mp4” is valid, but “multimedia\ThisFile.mp4” is not.)

6.6 Other Encodings Are Supported

The source of a Kindle book can be encoded in many different ways. All encodings are supported, provided that:

- The encoding of the HTML files is clearly stated in the HTML; and

- The computer used for compiling the sources supports the encoding and knows how to convert it to Unicode.

Amazon recommends specifying the encoding of the HTML by using the `<meta>` tag in the `<head>` section or an XML declaration.

Method 1:

```
<html>

<head>

…

<meta http-equiv="content-type" content="text/html; charset=UTF-8">

…
```

Method 2:

```
<?xml version="1.0" encoding="UTF-8"?>
```

6.7 Use Supported Characters and Spaces

Characters should be represented using plain text UTF-8 characters, except where XML entities are strictly required or are easier for humans to read than their character equivalents. For example, instead of using the `"©"` entity, use the © character.

XML entities are strictly required for "<" (`<`), ">" (`>`), and "&" (`&`).

The only supported spaces are the normal space, the non-breaking space (` `) and the zero-width non-joiner (`‌`). Use of any other space can break the selection, dictionary lookup, and line-wrap algorithms.

Do NOT use Unicode format characters, as they may cause problems.

6.8 Design for a Good eBook Experience

Kindle supports float via CSS, but this does not guarantee that the floating of text and images will produce an exact replica of the print layout on all Kindle devices and applications. If float is not producing the desired result, Amazon recommends rethinking the design and layout to create the best possible eBook experience rather than fixating on duplicating the print experience on a device. Using fixed-layout format just to replicate print layout is not allowed in Kindle books because customers report this as a bad user experience.

7 External Link Guidelines

External links within Kindle books should be present only if they directly enhance the reader experience and the content of the title as determined by Amazon. Some examples of this include:

- Links to multimedia content directly related the content of title;
- Links to additional ancillary material (e.g., checklists, assessment forms, craft patterns, and similar printable materials);

- Links to topical websites (e.g., link to Whitehouse.gov in a Kindle book about the American government);
- Social media related to the book or author (e.g., Twitter hashtag).

Some examples of prohibited links include:

- Links to pornography;
- Links to commercial eBook store sites other than Amazon;
- Links to web forms that request customer information (e.g., email address, physical address or similar);
- Links to illegal, harmful, infringing, or offensive content;
- Links that are malicious in intent (e.g., virus, phishing, or similar).

Amazon reserves the right to remove links in its sole discretion.

8 QA Standards

Amazon strongly recommends that you verify your exported content before converting it into a Kindle book because some content creation tools format content differently when exporting to HTML.

In addition, Amazon encourages you to review the entire book for:

- missing content;
- wrong content;
- typos and complete character support;
- alignment errors;
- forced typeface throughout the entire book (reflowable only) or correct font (fixed-layout);
- image quality;
- proper paragraph spacing and indenting; and
- forced font color or background color (reflowable only).

These errors negatively affect readability and may require the Amazon team to suppress the title to protect the reader's experience.

8.1.1 Testing Kindle Books

There are three ways to test your Kindle book before adding it to the Kindle store:

1. **Use the Kindle Previewer.** You can test your EPUB file using the Kindle Previewer software, available for both Windows and Mac OS X. The Kindle Previewer allows you to select views that represent the different devices including Kindle e Ink, Fire tablet, Kindle for PC, and Kindle for IOS. The most recent version of Kindle Previewer can be downloaded for free from www.amazon.com/kindleformat/kindlepreviewer.

2. **Use Kindle devices and Kindle applications.** You can test KF8 content on a Kindle e Ink device or a Fire tablet.

3. **Use KDP.** The Kindle Direct Publishing Platform accepts a variety of book formats and provides preview capability on the website. To learn more or sign up, visit http://kdp.amazon.com.

Note: When testing Kindle Edition with Audio/Video content, the audio and video cannot be previewed.

Consider device compatibility. Keep in mind that users may wish to read your content on an extremely wide variety of devices, with very small or large screens and low or high resolution. Try to test your content on as diverse a selection of devices as possible, especially for complex Kindle books.

Once you can read your book, use this checklist to confirm that your Kindle book does not contain blatant errors.

1. Open the book for the first time or go to the cover page.
    - **Cover:** The Kindle book should have a cover.
    - **Single Cover:** From the cover, flip to the next page. There should not be another image of the cover page.
2. Go to the table of contents.
    - In the table of contents, each item should be clickable and should link to the correct location in the book. There should be no page numbers in the TOC.
3. Go to any location in the book (reflowable eBooks only).
    - **Font size:** Change the font size in the Kindle menu; the book font should change accordingly. Regular text should not be bold or italicized.
    - **Typeface:** Change the typeface in the Kindle menu; the book font should change accordingly. If you have designed your book to use only a specific font file, please make sure that you have followed guidelines in section 9.3.8 Using Embedded Fonts. Not following these guidelines could lead to the Kindle settings reverting to the customer's preferred reading font.
4. Go back to the first page and flip through every page of the book.
    - **Images:** Images should not be too small. Make sure that all text in images is legible. Large pictures should be scaled to fit the page and display in their entirety on one screen.
    - **Tables:** Tables should appear correctly. Make sure that all text in tables is legible.
    - **Material only included with physical book:** There should not be any references to material (such as a CD or DVD) that is only included with the physical book.
    - **Background settings (reflowable eBooks only):** On a Kindle tablet device or Previewer, confirm that your text is legible in all background color modes (white, black, mint, and sepia).
    - **Magnification (fixed-layout only):** Activate pop-ups/panel view and check that all text content and/or panels have corresponding magnification, no content is overflowing the edge of the screen, and magnified reading order is correct.

Part III. Guidelines for Specific Types of eBooks

9 Creating Text-Heavy Fiction and Non-Fiction eBooks (Reflowable)

Amazon refers to text-heavy fiction and non-fiction eBooks as "reflowable" because this type of content reflows when an eBook's text settings are changed. In general, a book can be converted as a reflowable eBook when the body text can be easily separated from the background images without losing any context or important layout design.

The reflowable format supports a number of features that allow readers to interact with and customize the way the text appears on their devices. These features include dictionary, X-Ray (when available), text-to-speech (when available), and the ability to change text and formatting settings.

9.1 Metadata Guidelines

eBooks are reflowable by default. Reflowable eBooks do not need to specify `meta name="book-type"` in the OPF file.

9.2 Layout Guidelines

Create the content using single column layout and avoid using `position:absolute` for alignments.

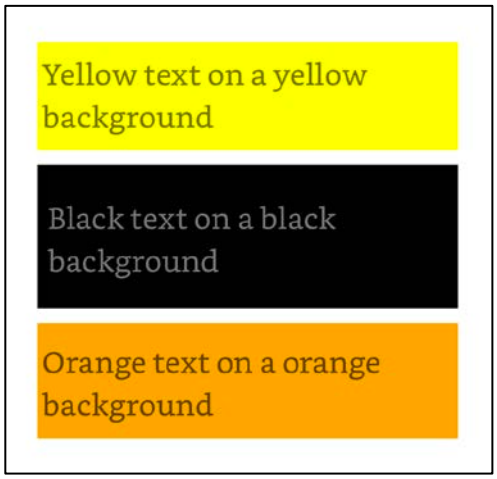
9.3 Text Guidelines

9.3.1 Body Text Must Use Defaults

The body text in a reflowable Kindle book (fiction and non-fiction) must be all defaults. Amazon encourages content creators to use creative styles for headings, special paragraphs, footnotes, tables of contents, etc., but not for body text. The reason for this is that any styling on body text in the HTML will override the user's preferred default reading settings. Users report such behavior as a poor reading experience. Here are the most important points:

- Body text must use the default font size (1em) and line height. Body text should not use the `<font size="…">` tag or the font-size and line-height attributes in CSS.
- Body text should not be primarily bold or italicized. Selected parts of the text can be bold or italicized for emphasis. This guideline only prohibits a book from being entirely bold, for example.
- Body text should not have an imposed font color throughout the book. If you prefer to use imposed font color in some sections of your book, please do not use too light or too dark a color. Light colors will not display with enough contrast on devices set to white backgrounds or on e Ink devices. Dark colors will not display well on devices set to black backgrounds. Please see the W3C recommendation described here for maintaining a readable contrast ratio between text and background colors. For grays, use colors within the hex value range of #666 to #999.
    - To determine if a color falls within this range, convert your color to RGB values using a tool such as http://hex-color.com/. Plug the resulting three numbers into the following formula: Y = 0.2126*R + 0.7152*G + 0.0722*B. If the value of Y falls within a range of 102 and 153, this color will create a good customer experience across Kindle devices and applications.

In a book with Enhanced Typesetting enabled, Kindle readers can be sure that any text of any color that appears above any background color will be legible. Font colors will adapt automatically to provide sufficient contrast with either the device color theme chosen by the reader or to the background color of any element. In the below example, the same colors (“yellow”, “black”, and “orange”) were applied to both the font and their background container; notice how the font color changes to provide readable contrast with the background. To learn more about Enhanced Typesetting, see section 15.



- Body text must not have a black or white background color. Customers report this as a bad user experience because it can create an awkward, boxy reading experience when the device background is set to a different color and because the text can become invisible when a user changes the background color setting on their device and the font color automatically inverts.
- Body text should not have a forced font face. Make sure that you have followed guidelines in section 9.3.8 Using Embedded Fonts. Not following these guidelines could lead to customers not having the ability to change their preferred reading font.
- Body text must not use non-breaking spaces in place of normal spaces in between words in paragraphs.
- Body text must not have an imposed left/right margin or padding throughout the book. If there are paragraphs that do require left/right margin to differentiate them visually from body text, like a recipe list or a block quote, margins applied to these sections should be specified as percentages rather than ems or point values.

9.3.2 Formatting Paragraphs

KindleGen automatically indents the first line of every paragraph by default. To change this behavior, use the text-indent style on the <p> tag. For example:

- `<p style="text-indent:0">` - no indentation of the first line
- `<p style="text-indent:2em">` - positive indent, 2 ems

We recommend using no more than 4 ems of text indent for body paragraphs.

To change the space before or after each paragraph, use the `margin-top` or `margin-bottom` styles respectively on the <p> tag. We recommend using em values for these attributes.

Never use the height property to control the size of elements containing text or instances of overlapping text may occur in your book. The height property should only ever be applied to images in reflowable books.

9.3.3 Monospaced Font Is Supported

Kindle uses a default font for content if none is specified; it also supports a monospaced font.

Kindle uses the monospaced font to render content in the following tags: `<pre>`, `<code>`, `<samp>`, `<kbd>`, `<tt>`, `<font face="courier">`, `<font face="monospace">`.

With the exception of `<pre>`, the tags listed above do not change the text alignment. If the content in these tags should be left-aligned, wrap the tags listed above in a `<div>` styled with CSS using `text-align:left`.

Publishers can include their own font for their content. Amazon has a quality assurance process to ensure that these fonts display well on e Ink-based devices and do not impair the reading experience.

9.3.4 Avoid Using Fixed Values for Most Elements

Avoid using fixed values such as points and pixels for CSS properties such as `font-size`, `width`, `height`, `margin`, `padding`, `text-indent`, and `line-height`. To enable rendering across various screen sizes and resolutions, specify these values in ems or percentages.

To ensure pagination, the Kindle Reader does not honor line-height value less than 1.2 em or 120%.

9.3.5 Margin and Padding Formatting

When using left or right `margin` and `padding` CSS properties, specify the values in percentage (%) instead of em units. This ensures that the margins do not grow too wide with large font sizes and impair reading. Margins should be assigned values of 0 or greater to keep content from falling off the edge of the screen or overlapping other content. Always set left and right margins to 0 for normal body text to allow users the full range of margin selection using device defaults. Top/bottom margins should be specified in ems so that spacing between paragraphs is easily distinguishable at any font or device size.

9.3.6 Drop Caps

Elements such as drop caps should be specified using percentages or relative units (positive or negative) instead of fixed values such as points and pixels. The top of the drop cap should be aligned with the body text. To create drop caps, Amazon recommends using the following sample CSS:

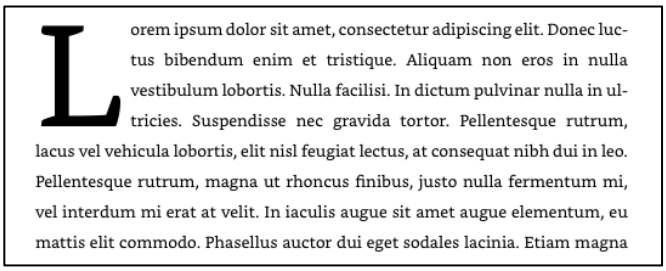
Example

```
p.para {
    font-size: 1em;
    margin-bottom: 0;
    margin-top: 0;
    text-align: justify;
    text-indent: 0;
}
@media  amzn-kf8
{
    span.dropcaps
    {
```

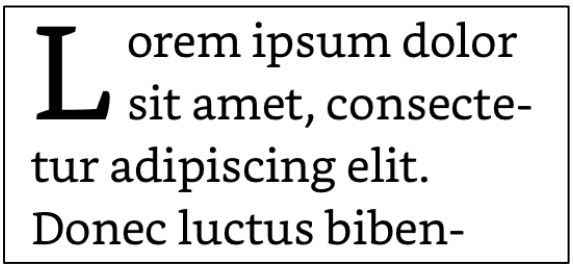
```
        font-weight:normal;
        font-size:320%;
        float:left;
        margin-top:-0.3225em;
        margin-bottom:-0.3245em;
    }
}
@media  amzn-mobi
{
    span.dropcaps
    {
        font-size:3em;
        font-weight: bold;
    }
}
<p class="para"><span class="dropcaps">T</span>his is a sample
```

To verify that the drop caps display as intended, test the book as described in section 8.1.1, Testing Kindle Books.

The following is an example of a drop cap formatted using this method in a book with Enhanced Typesetting enabled (to learn more about Enhanced Typesetting, see section 15):



Small font setting



Large font setting

9.3.7 Use CSS for Page Breaks

Do not insert blank lines of text to create page breaks. Use the CSS `page-break-before` and `page-break-after` attributes. Alternatively, place each section of content that should appear after a page break in a new HTML document.

9.3.8 Using Embedded Fonts

Kindle Format 8 supports embedded fonts within the eBook. These fonts can be either Open Type (OTF) or True Type (TTF). Kindle does not recommend the use of Type 1 (Postscript) fonts. To provide Kindle customers with the best possible reading experience, reflowable books that use Type 1 fonts are rendered using Kindle fonts by default. On KF8-enabled devices and apps, customers have the option to turn publisher-provided fonts on or off.

The font files within the book are intentionally obfuscated to reduce the probability of reuse, but it is the responsibility of the publisher to secure the appropriate license rights for fonts. Unless embedded fonts are necessary to convey intent, Amazon recommends using the default set of fonts installed on Kindle devices and apps because they have been tuned for high quality rendering.

Only embed fonts that are not currently available on devices and apps. When selecting a font, consider usability for visually impaired readers and select a simple, clear font which will contrast well against all tablet and e Ink backgrounds.

9.3.9 Customizing Font Selection

The primary or main font in a book should be set at the <body> level. If you prefer to use additional text styling such as bold or italics, ensure that the styles are set on the text rather than the font so that any font that the customer selects correctly displays these styling elements. Below are examples of both correct and incorrect implementation of customizing fonts in a Kindle book.

Incorrect HTML Code	Correct HTML Code
<html> <body> <p **style="font-family:PrimaryFont"**> Primary font content Secondary font content <p **style="font-family:PrimaryFont"**> Primary font content <p **style="font-family:PrimaryFont"**> Primary font content </body> </html>	<html> <body style="font-family:PrimaryFont"> Primary font content Secondary font content Primary font content Primary font content </body> </html>

The same behavior can be achieved by using CSS classes as shown below.

Incorrect CSS Code	Correct CSS Code

.indent {
font-size: asize;
font-family: PrimaryFont;
}
 	
.sidebar-text {
font-family: SecondaryFont;
font-weight: bold;
}
	body {
	font-family: PrimaryFont;
	}
	
	.sidebar-text {
	font-family: SecondaryFont;
	font-weight: bold;
	}

9.3.10 Page Number Guidelines

Kindle books do not always map directly to page numbers in physical editions of the book. Even if the Kindle Real Page Numbers feature is activated in the Go To menu, references within the eBook to page numbers should be handled as follows:

- Table of contents: If there are page numbers in the print source's TOC, they should be removed in the digital conversion. The name of the section should be retained and hyperlinked to the relevant location in the eBook. For example, if a print source TOC displays the entry "Chapter 1 ... P. 36", then the eBook should only display "Chapter 1" hyperlinked to the correct digital location.

- Internal links: If there is text that refers to another page in the eBook, such as "see page XX", this text should be linked to the relevant paragraph within the eBook.

- Index: Every page number in the index should be linked to the relevant paragraph in the eBook (or the relevant illustration, table, or chart).

- Links within index: If there is an entry that references another section of the index, such as "see also XXX", this text should be linked to the relevant section within the index.

9.3.11 Footnote Guidelines

Amazon recommends formatting footnotes with bi-directional hyperlinks (the text is linked to the footnote and the footnote is linked back to the text). This makes it easier for customers to return to the text after viewing the footnote. On some Kindle devices, such as Kindle Paperwhite, footnotes with bi-directional hyperlinks are displayed in a pop-up.

Example:

```
This sample text has a footnote.<sup><a href="footnotes.html#fn1"
id="r1">[1]</a></sup>

<p id="fn1"><a href="chapter01.html#r1">1.</a> This is the footnote text.
```

9.4 Image Guidelines

These guidelines apply to reflowable fiction and non-fiction books, but are not applicable to image-intensive fixed-layout children's books (see section 10) and fixed-layout graphic novels/manga/comics (see section 11). For cover image guidelines, see section 4.

9.4.1 Use Supported Input Formats

The Kindle platform supports GIF, BMP, JPEG, non-transparent PNG, and Scalable Vector Graphics (SVG) images.

When using images for schemas, charts, tables, maps, or anything that includes text, pay special attention to the legibility of the final image.

Add images to the source using the standard HTML `<img>` tag.

Use a resolution of at least 300 ppi for all images.

Important: Use RGB or sRGB as the color profile when saving your files. Kindle does not support CMYK.

9.4.2 Size Images for Responsive Layouts

Amazon recommends that block and float images be styled using a percentage value for the `width` style attribute. This will ensure that images always occupy the same percentage of space on the screen irrespective of device resolution.

Inline images should be sized in em units so that they scale in relation to the text around them when users adjusts the font size of their reading system.

9.4.3 KindleGen Performs Automatic Image Conversions

The maximum size of a mobi is 650 MB. KindleGen performs automatic image conversions to optimize the content for Kindle. You can obtain the best results by optimizing your images to the highest quality possible within the file size limit before inputting your files into KindleGen.

9.4.4 Use Color Images

For the best user experience across Kindle devices and applications, use color images whenever possible and relevant. If an image is a photograph, it should be formatted as a JPEG.

Even in marketplaces where only e Ink devices are currently available, use color images when possible for future compatibility.

9.4.5 Photographs Should Be Optimized for High-Resolution Devices

Photographs should use the JPEG format and be saved at high quality. (In Adobe Photoshop, choose **Save As**, select **JPEG**, and click **Save**. In the following **JPEG Options** dialog box, choose a **Quality level** of **10** or higher.) Photographs should use the highest resolution available within the file size limit.

Photographs of less than 300 x 400 pixels create a poor customer experience on high-resolution devices.

Amazon recommends that images display clearly at 2X magnification across devices. This means that if an image is intended to display at full width on device, its width should be captured at 3200 px (this is twice the width of our highest resolution device, the Kindle Fire HDX 8.9”). Smaller images can be resized accordingly.

If the photographs are in GIF format or are too small, converting them to JPEG or artificially increasing the size does not improve the quality. Go back to the original source to create a JPEG image with sufficient resolution.

Some images, such as historical photographs, may not be available at 300 ppi or greater. In these situations, provide the best image quality that you have.

9.4.6 Use GIF or PNG for Line-Art and Text

Line-art images are graphics drawn with a limited number of solid colors (such as images drawn by Illustrator, Paint, or Power Point, including black-and-white drawings). Text, graphics, charts, and tables are examples of images that are line-art.

Line-art should be in GIF or PNG format. The JPEG algorithm tries to blend parts of the image together, and blurs the sharp edges of the line-art, causing the image (and any text it contains) to be blurry.

Text appearing in line-art images should be sharp and legible.

Optimize line-art GIFs before submitting them to KindleGen. Resizing or JPEG compression introduces blurriness or unwanted artifacts in line-art images, which is why Amazon insists on GIF or PNG file formats for line-art.

To optimize GIFs and PNGs and make them fit the image size limit, try the following tips:

- Try reducing the number of colors used. This can often be done without altering the quality of the image. Line-art images that appear to be black and white might actually be in color because of certain anti-aliasing algorithms. Here is an example (notice the shades of red and blue around the "A" in the left picture):

- Remove white margins around the image, if any exist. When cropping, consider how the image will look on devices set to white, sepia, and black backgrounds.

- Resize the image, if necessary, but pay close attention to the legibility of text (see section 9.4.7, Image and Font Size Requirements for Line-Art and Text).

9.4.7 Image and Font Size Requirements for Line-Art and Text

An image containing text should not be significantly larger than a Kindle screen. The Kindle e Ink devices offer the possibility to rotate an image to use more screen real estate. The Fire tablets and the Kindle for iPhone application allow zooming and panning.

- The MINIMUM size of text is 6 pixels for the height of a lowercase "a."The image itself will need to be larger than 6 pixels in height if there is any extra space above or below the "a." For an image that contains only a single line of text, such as the example below, the image should be at least 45 pixels in height so that it displays proportional to surrounding text content.

$$p_t = d_t + \delta_1 d_{t+1} + \delta_1 \delta_2 d_{t+2} + \delta_1 \delta_2 \delta_3 d_{t+3} + \cdots ,$$

Example Images

Description	Image
A table with line-art/text content rendered as an image. This GIF image is 317 x 233 pixels and 6 KB in size. The text is sharp and legible. The font size requirement is met ("a" is 7 pixels high).	

Description	Image
The same image with JPEG compression. Compression artifacts appear, making the text blurry although it remains legible. The size has increased to 17 KB.	
Bad quality: The image is blurry because of resizing and JPEG compression. The text is not legible. This will be rejected.	

Fruit	#/week	Sales	Remarks	Sales
Apple	4	$ 250,000	Sold well	$ 250,000
Peach	2	$ 150,000		$ 150,000
Banana	5	$ 670,000		$ 670,000
Pear	3	$ 560,000	Need more marketing	$ 560,000
Plum	2	$ 432,000		$ 432,000
Walnut	1	$ 35,000		$ 35,000
Pineapple	15	$14,000	Prospective sales	$14,000
Grapefruit	5	$1,345,000		$1,345,000
Hazelnut	3	$ 25,000		$ 25,000
Total	16	$3,679,000		$3,679,000

An example of the largest acceptable table rendered as an image is given below. The size is 500 x 600 pixels, which is the maximum. The font uses a body size (height of an “a”) of 7 pixels, which is just above the 6 pixel minimum. The size of the GIF is 33 KB.

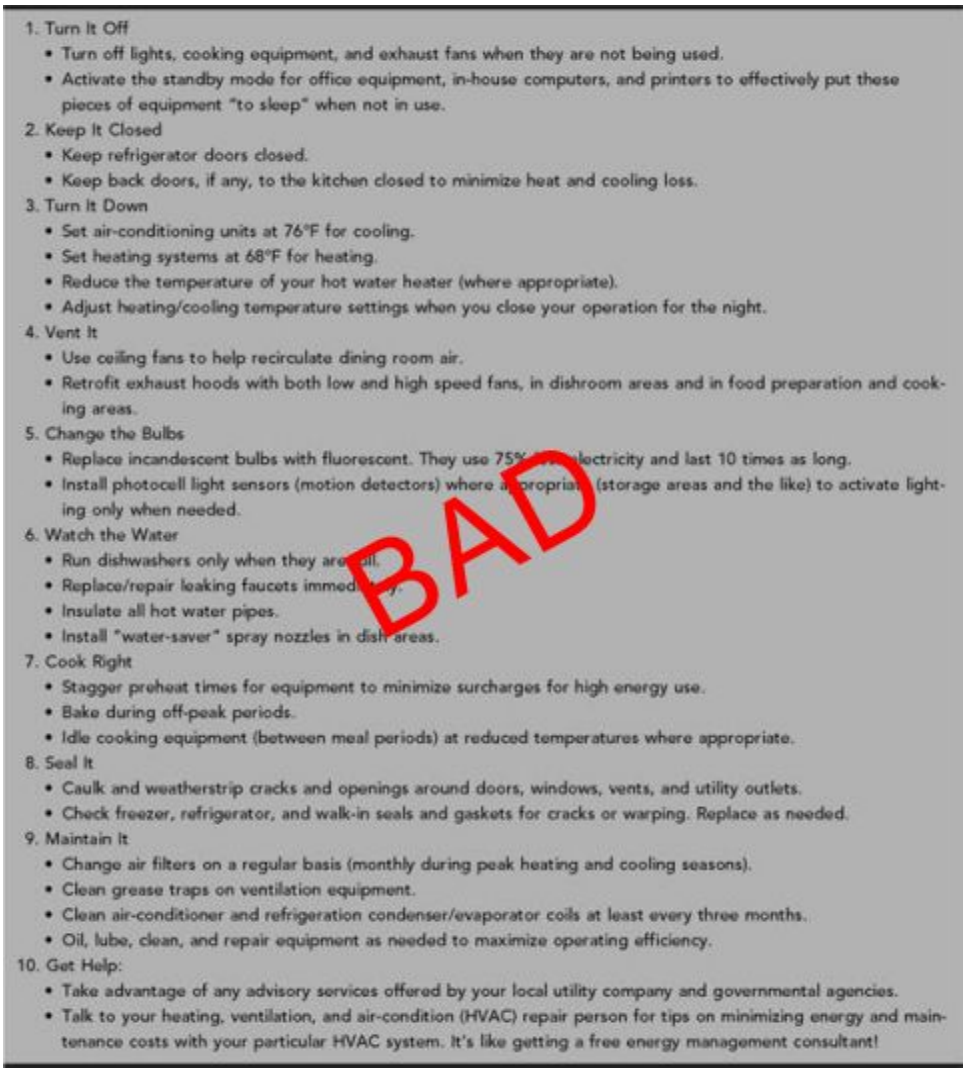
Fruit	#/week	Sales	Remarks	Sales	Quality	Country of origin
Apple	4	$ 250,000	Sold well	$ 250,000	1st choice	U.K.
Peach	2	$ 150,000		$ 150,000	1st choice	Holland
Banana	5	$ 670,000		$ 670,000	2nd choice	Germany
Pear	3	$ 560,000	Need more marketing	$ 560,000	3rd choice	Mexico
Plum	2	$ 432,000		$ 432,000	1st choice	Argentina
Walnut	**1**	**$ 35,000**		**$ 35,000**	**3rd choice**	**Uruguay**
Pineapple	15	$14,000	Prospective sales	$14,000	2nd choice	Ethiopia
Grapefruit	5	$1,345,000		$1,345,000	3rd choice	Iran
Hazelnut	3	$ 25,000		$ 25,000	2nd choice	Japan
Apple	4	$ 250,000	Different batch	$ 250,000	1st choice	U.K.
Peach	**2**	**$ 150,000**		**$ 150,000**	**1st choice**	**Holland**
Banana	5	$ 670,000		$ 670,000	2nd choice	Germany
Pear	3	$ 560,000	Negotiated a good price	$ 560,000	3rd choice	Mexico
Plum	2	$ 432,000		$ 432,000	1st choice	Argentina
Walnut	1	$ 35,000		$ 35,000	3rd choice	Uruguay
Grapefruit	5	$1,345,000		$1,345,000	3rd choice	Iran
Hazelnut	3	$ 25,000		$ 25,000	2nd choice	Japan
Apple	4	$ 250,000	Sold well	$ 250,000	1st choice	U.K.
Peach	2	$ 150,000		$ 150,000	1st choice	Holland
Banana	**5**	**$ 670,000**		**$ 670,000**	**2nd choice**	**Germany**
Pear	3	$ 560,000	Find alternative suppliers	$ 560,000	3rd choice	Mexico
Plum	2	$ 432,000		$ 432,000	1st choice	Argentina
Walnut	1	$ 35,000		$ 35,000	3rd choice	Uruguay
Pineapple	15	$14,000	Prospective sales	$14,000	2nd choice	Ethiopia
Grapefruit	5	$1,345,000		$1,345,000	3rd choice	Iran
Hazelnut	3	$ 25,000		$ 25,000	2nd choice	Japan
Total	**16**	**$3,679,000**		**$3,679,000**		

9.4.8 Prefer HTML to Images

Do not render large chunks of text as images. If an image contains whole paragraphs of text, it should not be an image. Instead, it should be HTML.

The following is an example of a text-heavy image that should be HTML.

Note: The image would be shrunk to fit the screen and become unreadable, while an HTML version would be paginated.



9.4.9 Positioning Image Captions

Amazon recommends placing a caption below the related image, so that the reader views the image before the caption. Define a line break between the image and the caption (for example, using a `<br />` tag) to make sure the caption text is not positioned adjacent to the image.

9.4.10 Controlling Image Aspect Ratio

To preserve aspect ratio of images, width and height cannot both be set to a fixed percentage. Either width or height can be set to the fixed percentage (such as 100%), but then the other property should be set to “auto” to preserve the aspect ratio.

9.4.11 Displaying Text Correctly within SVG

To display text correctly within an SVG, use the font-size attribute for <text> inside the SVG.

Example

```
<html>

<body>

<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
```

```
<text x="20" y="20" font-size=20 fill="red">svg text sample</text>

</svg>

</body>

</html>
```

9.4.12 Use Supported SVG Tags and Elements

A publisher can reference the SVG files from within an HTML file using inline <svg>, <img>, <embed>, or <object> tags. Please refer to the SVG specification http://www.w3.org/TR/SVG/ for details about SVG.

Example

```
<html>

<body>

<svg xmlns="http://www.w3.org/2000/svg"><!—Inline SVG--></svg>

<img src="svgfile1.svg"/>

<embed src="svgfile2.svg"/>

<object src="svgfile3.svg"/>

</body>

</html>
```

Supported SVG Elements

<circle>	<feFlood>	<feSpotLight>
<clipPath>	<feFuncA>	<feTile>
<defs>	<feFuncB>	<feTurbulence>
<ellipse>	<feFuncG>	<filter>
<feBlend>	<feFuncR>	<font-face>
<feColorMatrix>	<feGaussianBlur>	<font-face-name>
<feComponentTransfer>	<feMerge>	<font-face-src>
<feComposite>	<feMergeNode>	<line>
<feConvolveMatrix>	<feMorphology>	<linearGradient>
<feDiffuseLighting>	<feOffset>	<marker>
<feDisplacementMap>	<fePointLight>	<mask>
<feDistantLight>	<feSpecularLighting>	<metadata>

`<path>`	`<rect>`	`<text>`
`<pattern>`	`<stop>`	`<textPath>`
`<polygon>`	`<style>`	`<tref>`
`<polyline>`	`<svg>`	`<tspan>`
`<radialGradient>`	`<symbol>`	`<use>`

Reference: http://www.webkit.org/projects/svg/status.xml.

9.5 Table Guidelines

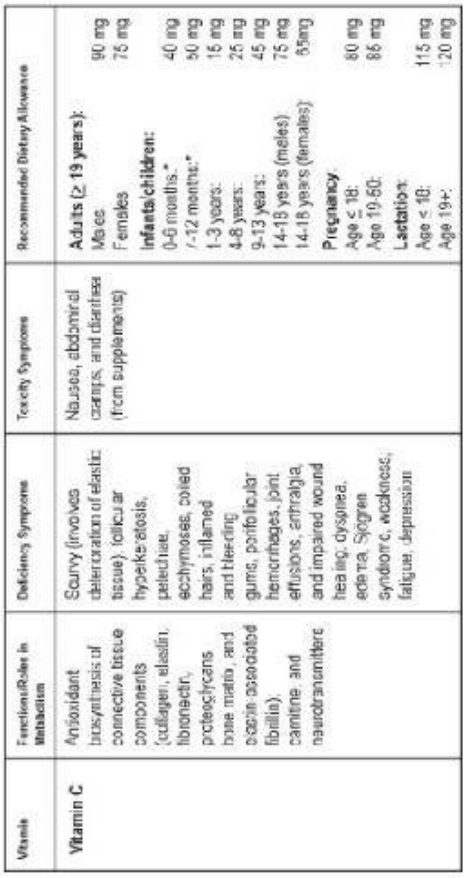
9.5.1 Avoid Large Tables

A table rendered as an image cannot be paginated because the entire image is displayed on one screen. If the table is rendered using HTML `<table>` tags, pagination is available and users can cursor through the cells in the table. If the table is significantly wider than the screen and forces panning, it creates a poor user experience.

For the best user experience, tables should not contain whole paragraphs of text or large pictures in a cell.

If a table is too large or contains too much text in its cells, consider reformatting it in a way that maintains legible text. Single columns of HTML text provide the best reading experience for customers using a variety of font sizes.

In the example below, rotating the table to better fit the screen does not help. To preserve the format and layout of the data, it would be better to have the paragraph text as plain HTML and only the right-most column rendered as an image or HTML table.

Large table rendered as an image: illegible text, unacceptable quality	The same content, reformatted as HTML and resulting in a legible, good-quality table
	Vitamin C Functions/Roles in Metabolism Antioxidant; biosynthesis of connective tissue components (collagen, elastin, fibronectin, proteoglycans, bone matrix, and elastin–associated fibrillin); carnitine, and neurotransmitters Deficiency Symptoms Scurvy (involves deterioration of elastic tissue); follicular hyperkeratosis, petechiae, ecchymoses, coiled hairs, inflamed and bleeding gums, perifollicular hemorrhages, joint effusions, arthralgia, and impaired wound healing; dyspnea, edema, Sjögren syndrome, weakness, fatigue, depression Toxicity Symptoms Nausea, abdominal cramps, and diarrhea (from supplements)

9.5.2 Create Simple HTML Tables

Use the `<table>` tags to create simple tables with standard rows and columns that can be displayed on Kindle devices and applications. KF8 has support for nested tables and merged cells, but Amazon recommends that publishers use this judiciously and only when necessary.

`Colspan` and `rowspan` attributes should be less than or equal to the total number of columns or rows (as appropriate) in the table.

9.5.3 Split Tables as Needed

There are times when it may be necessary to format a table as an image, but the image is still too large to be legible on one Kindle screen. In this case, it is a good idea to split the image. The following example is a guideline to use when splitting a 2-page table. This logic can be extended for multiple-page table images.

Example: Split the image in half horizontally 60% of the way down the image, then split the header, copy it to the bottom half of the image, and stitch these into a new image. The final two images should then be the same size, with table headers.

Revise the source image, not the converted GIF; otherwise, the image will be converted into GIF format twice, which might result in lower quality.

9.5.4 Optimize for Maximum Table Size

Optimize tables to be no larger than 10 Kindle screens. A Kindle screen is approximately 24 rows of 60 characters, although the Kindle DX shows more characters. The character limit is the maximum number of characters in any one row. There are a limited number of combinations for a table that looks like this (see below). If a table has more characters than the maximum number specified below (given the number of rows), split the table into smaller tables or images, as described in section 9.5.3, Split Tables as Needed. In this case, characters are non-formatting characters (the actual text that a user sees when looking at the contents of a table).

Number of Rows	Maximum Characters (Per Row)
1 – 24	600
25 – 48	300
49 – 72	180
72 – 120	120
121 – 240	60

10 Creating Children's Picture Books (Fixed-Layout with Text Pop-Ups)

Certain books have elements with fixed dimensions and orientation that do not allow fonts to be resized or text to be reflowed. For example, children's books have full-page images with text set precisely in relation to the background art.

Fixed-layout books do not support reflowable text and should only be used when the entire book is a fixed-layout format; books cannot be partially reflowable or partially fixed-layout.

To demonstrate best practices in creating fixed-layout books, Amazon provides a children's book example at www.amazon.com/kindleformat (under the **KindleGen Examples** heading). This example is a demonstration of how to create content to take advantage of fixed-layout with Region Magnification. It is not intended to be an HTML tutorial.

10.1 Metadata Guidelines

The OPF file specifies metadata necessary for fixed-layout books. For a demonstration, see the children's book example at www.amazon.com/kindleformat (under the **KindleGen Examples** heading).

Metadata	Description

Layout can be specified using one of the following metadata fields:
 	
`1) <meta property="rendition:layout">pre-paginated</meta>`
 	
`2) <meta name="fixed-layout" content="true"/>`
	Required. Identifies the book as having a fixed layout.
	
	Valid values for `rendition:layout` metadata are `reflowable` or `pre-paginated`. The default value is `reflowable`.
	
	Valid values for `fixed-layout` metadata are `true` or `false`. The default value is `false`.

`<meta name="original-resolution" content="1024x600"/>`	Required. Identifies the original design resolution of the content (“1024x600” is only an example). The pixel dimensions can have any positive integer value. These values must equal the overall aspect ratio of the original content.

Orientation can be specified using one of the following metadata fields:
 	
`1) <meta property="rendition:orientation">landscape</meta>`
 	
`2) <meta name="orientation-lock" content="landscape"/>`
	Optional.
	
	Valid values for `rendition:layout` metadata are `portrait`, `landscape`, or `auto`. Locks the orientation of the book to either portrait or landscape. If the value is `auto`, both portrait and landscape modes are supported. The default value is `auto`.
	
	Valid values for `orientation-lock` metadata are `portrait`, `landscape` or `none`. Locks the orientation of the content to either portrait or landscape. If the value is `none`, both portrait and landscape modes are supported. The default value is `none`.

`<meta name="RegionMagnification" content="true"/>`	An optional tag for enabling the Kindle Panel View and Kindle Text Pop-Up features that are required for comics and children's books. Valid values are `true` or `false`. The default value is `false`. Enabling this feature requires additional CSS instructions as specified in section 10.4.2.
`<meta name="primary-writing-mode" content="horizontal-rl"/>`	Optional. Defines page rendering order, reading mode, and reader navigation (including Kindle Text Pop-Up, Kindle Panel View, and Kindle Virtual Panels). Valid values are `horizontal-lr`, `horizontal-rl`, `vertical-lr`, and `vertical-rl`. The default value is `horizontal-lr`.

Metadata	Description
`<meta name="book-type" content="children"/>`	Optional for children's books. Removes reader functionality (e.g., share) which may not be relevant for certain books such as children's. Valid values are `children` or `comic`.

10.2 Cover Image Guidelines: Including Back Cover for Children's Content

While Kindle books in reflowable format do not use back covers, it provides a sense of closure to the narrative for children's content. It is best to include a back cover as part of the fixed-format children's book design. Remove barcodes, price listings, and promotional content from the back cover image. Do not include popups for any text on the back cover unless the back cover includes story text.

10.3 Text Guidelines: Including Specific Fonts

Fixed-format titles do not allow users to choose and vary fonts. Using CSS `@font-face` and packaging fonts with the title guarantees book design look-and-feel to be consistent across all devices and screens. This not only ensures that the exact fonts used for the source are used in the fixed-format title, but that HTML text has more fluid rendering between the page view and the Region Magnification view.

Example:

```
@font-face {

  font-family: "Arial"; /* assigns the name of the font to use */

  src: url(../fonts/arial.otf); /* includes the file for the correct font */

}
```

10.4 Content Requirements

10.4.1 Requirement #1: Using HTML File Structure

Fixed-layout content must have a single HTML file for each page represented on a Kindle device. Publishers may use the OPF metadata to create double-page spread reading experiences.

Portrait orientation lock:

1 print page = 1 HTML file

Example:

Landscape orientation lock:

2 print pages (1 two-page spread) = 1 HTML file

Example:

10.4.2 Requirement #2: Using Region Magnification (Pop-ups)

Fixed-layout content does not allow the user to change the font size; allowing font sizes to change could obfuscate content relevant to storytelling. Kindle uses Region Magnification (pop-ups) to enlarge fixed-layout text without altering the original layout. For an example of Region Magnification, see the example below.

The user activates Region Magnification by double tapping an “active area” on touch screen devices. (On non-touch screen devices, clicking the up arrow on the 5-way controller selects the region and clicking the center button activates Kindle Text Pop-Up or Kindle Panel View.) During Region Magnification, the active area (source element) is hidden and the magnification area (target element) is displayed.

To support Region Magnification, the following steps are required:

1. Set the active area by creating a well-defined HTML anchor (`<a>`) element around the text to be enlarged. The anchor must specify the `app-amzn-magnify` class. The anchor should also have the following attributes stored in a JSON object (http://www.w3schools.com/json/json_syntax.asp) as part of the `data-app-amzn-magnify` value:

    a. `"targetId":"<string:elementId>"` = unique element id of the magnification area (position and font size are set in CSS file)

    b. `"sourceId":"<string:elementId>"` = unique element id of the source that will be magnified

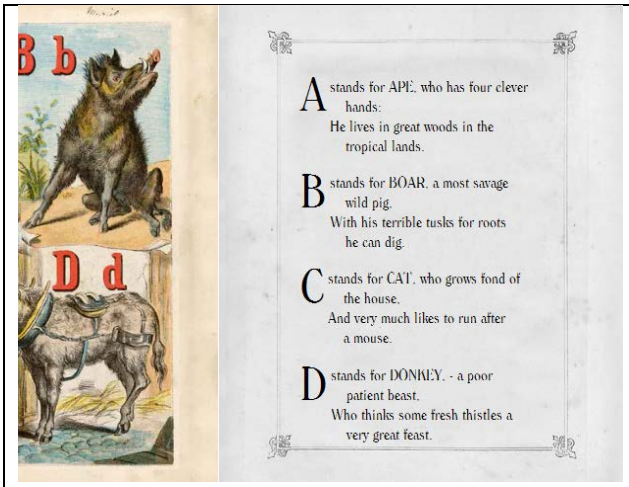
    c. `"ordinal":<integer:reading order>"` = reading order of the magnification areas (the order in which panels appear as part of the reading flow). This is required for all text that uses Region Magnification.

2. Create a target `<div>` element that is aligned to completely cover the text being magnified and positioned to minimize covering the background art of the page. This ensures that when a user activates Region Magnification, the source text will not disappear from the page view. When magnification is activated, the source text is no longer displayed. It is also important to not position a popup directly abutting the right or bottom edges of the screen. Differences between device types can create content overflow error if popups are too close to these edges. Check content on as many different kinds of devices as possible before publishing.

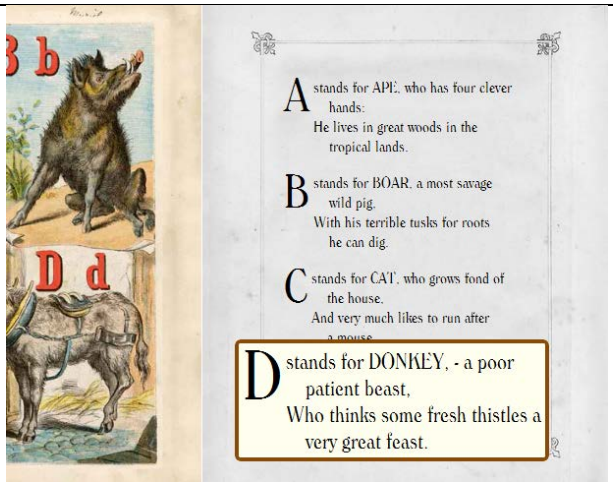
3. The font size of text in the Region Magnification `<div>` should be set to 150% of the regular font size on the page. There are several exceptions to this:

    - One exception is when the text on the page is so large that magnifying it to 150% would make it harder to read instead of improving readability.

    - Another exception is when the text on the page needs to be increased by more than 150% to improve readability in the Region Magnification `<div>`. For example, if the font size of the text on the page is 45%, the font size of the text in the Region Magnification `<div>` may need to be magnified to 225% to be readable.



Fixed-layout sample page



Same sample page with Region Magnification activated

Example:

```
<div id="fs1-4-org" class="txt fs1-txt4">

<a class="app-amzn-magnify" data-app-amzn-magnify='{"targetId":"fs1-txt4-magTarget",
"sourceId":"fs1-4-txt", "ordinal":4}'>

<p id="fs1-4-txt"><span class="dropcap">D</span>stands for DONKEY, - a poor patient
beast<br/>Who thinks some fresh thistles a very great feast.</p></a></div>

…

<div id="fs1-txt4-magTarget" class="target-mag fs1-txt4"></div>
```

10.4.3 Requirement #3: Setting Images as Background Images

The images in fixed-layout books must be set as background images using the CSS `background-image` property (instead of using HTML `<img>` tags). This is important for children's content and comic books because HTML images interfere with Region Magnification if they are not set as background images. If background images are correctly set, the images will not open in the image viewer when double tapped.

10.5 HTML and CSS Guidelines

10.5.1 Applying CSS Reset

Apply a CSS reset to fixed-layout books. A CSS reset removes the inconsistent styles that browsers automatically apply, such as font sizes, margins, etc. Adding a CSS reset, such as the YUI reset (http://yuilibrary.com/yui/docs/cssreset), removes these inconsistencies, allowing designers to build on a dependable styling template.

10.5.2 Including One CSS File Per HTML Page

To increase page-turn performance on fixed-layout books, include one short, relevant CSS file per HTML page. CSS files should only contain information that is directly referenced by the associated HTML files.

10.5.3 Optimizing Content for Full Screen

Kindle books are read across a wide variety of devices (e.g., Fire tablets and other manufacturers' smartphones and tablets) and a wide variety of screen dimensions. The 2013 Kindle Fire HD 8.9" has a resolution of 1920 x 1200 pixels. Design the content to maintain this aspect ratio, if possible.

For the best user experience, Amazon strongly encourages publishers to design fixed-layout content to maximize the available space of the screen dimensions. If the content has a different aspect ratio or size, the Kindle devices and apps display it scaled to fit the screen, centered, and surrounded by a white margin (letterbox).

Fixed-layout and other image-heavy content is more likely to be magnified because customers prefer to read with Kindle Panel View or on devices with large screens. Amazon recommends submitting images scaled to support at least 2X magnification with high quality. For example, if planning for the 2013 Kindle Fire HD 8.9", the image pixel dimensions should be at least 3840 x 2400 (this matches the aspect ratio and would support 2X zoom). Always use Kindle Previewer to validate the quality of the content.

10.5.4 Using Large Region Magnification Tap Targets in Children's Books

The primary purpose of Region Magnification is to aid accessibility and is more effective when the tap target is larger than the area being magnified. To enable a larger area, consider adding a padding of 20 to 40 pixels to the `app-amzn-magnify` anchor elements, but do not let the tap targets overlap.

10.5.5 Using position:absolute for Text on Image

For text on an image that needs to be positioned exactly, use the `position:absolute` attribute. Only use this attribute for books that need a fixed layout, such as children's picture books with text specifically positioned in relation to background image elements.

10.5.6 Future-Proofing Fixed-Layout Content in Children's Books

By definition, fixed-layout is designed for a single screen size. To future-proof your content, Amazon recommends using percentage or em values instead of pixels or points for all positioning of text blocks and Region Magnification pop-ups. Specifying text position or font size with pixels prevents the content from scaling to new devices.

For example, the children's book example at www.amazon.com/kindleformat (under the **KindleGen Examples** heading) includes a sample style sheet (style-150.css) that demonstrates the minimal set of changes required to scale to a device that is 150% larger. These changes are limited to a base font-size and changes in the container height and widths (approximately 5 updates).

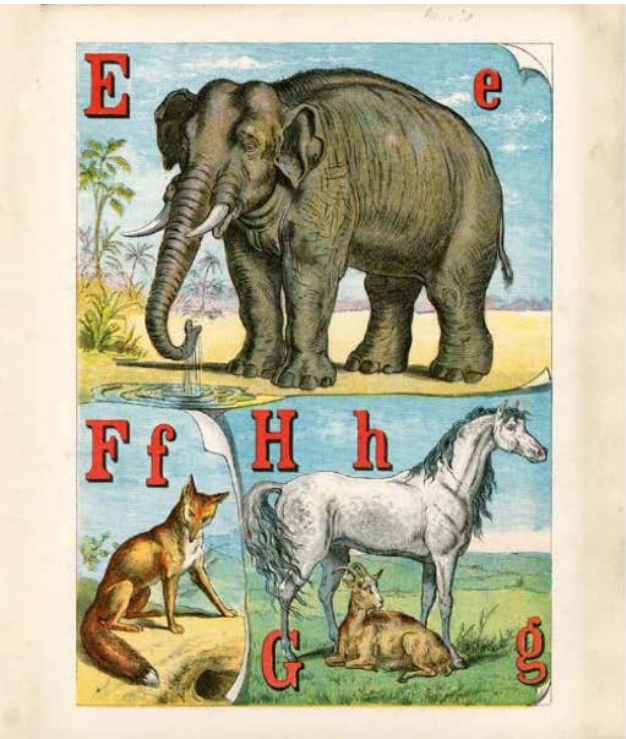
10.6 Creating Children's Books with Multipage Background Images and Text

This section explains the proper way to create pages that contain a single background image and text. While there are many potential solutions, Amazon's goal is to ensure that markup is easily portable with minimal effort. The provided template meets this goal by updating the CSS rules without changing the HTML.

10.6.1 Using Side-by-Side Images to Form a Double-Page Spread When Orientation-Lock Is Set to Landscape

Many books have two-page spread that consists of a single image. Other books have a two-page spread that consists of two side-by-side images.

In the example below, the double-page spread is 1024 x 600 pixels, which is full-screen resolution for the Kindle Fire (1st generation). The images for each page should have dimensions exactly half the width of the full screen: 512 x 600 pixels. The unique parts of each element are labeled using CSS IDs; the common parts use CSS classes. The left image displays on the left side of the spread. The right image is shifted to the right side of the page by defining a `margin-left` style set to the width of the left side image.



E stands for ELEPHANT, might in size
But kind to all children, and gentle and wise.

F stands for FOX, who lives down under ground,
And often is chased by the huntsman and hound.

G stands for GOAT, - o'er the Welsh hills it flies,,
And very rich milk to the children supplies.

H stands for HORSE, - one of man's greatest friends:
On our wants and our pleasures he gladly attends.

HTML

```
<div class="fs">
<div id="fs1-left" class="leftPage"></div>
<div id="fs1-right" class="rightPage"></div>
</div>
```

CSS

```
/* Region sized for both pages */

div.fs {

  height: 600px;

  width: 1024px; /* 2 x page width */

  position: relative;

}

div.leftPage {

  position: absolute;

  background-repeat: no-repeat;

  height: 600px;
```

```
  width: 512px; /* 1 x screen width */
}
div.rightPage {
  position: absolute;
  background-repeat: no-repeat;
  height: 600px;
  width: 512px; /* 1 x screen width /
  margin-left: 512px; /* this value equals the value of the left side image width */
}
```

10.6.2 Positioning Text Blocks

Specify the proper position and font size using percentages. This allows the position to scale consistently at different resolutions, ensuring compatibility across a wider range of devices and screens. Each paragraph should be grouped within a single `<div>` element, with multiple lines broken by `<br/>` elements. If custom line spacing is required, assign this via CSS style declarations instead of adding extra markup such as multiple `<div>` containers or extra line break tags.

The example in section 10.4.2, Requirement #2: Using Region Magnification (Pop-ups), expands on the two-page spread example and illustrates how to place text on top of a background image: text is positioned within a fixed spread block, uses percentages for the margin attribute, and is aligned and spaced via CSS.

10.6.3 Aligning Text

By default, text aligns to the upper left corner of the containing HTML element. Many books may have text that is right-aligned, bottom-aligned, or justified. The easiest way to identify the alignment is to imagine an outline around the text and identify which edges of the paragraph are associated with a margin (top, left, right, bottom). If the alignment is unclear, use the default upper left, although this makes positioning of the magnified element more difficult.

Never use non-breaking space (` `) characters for text alignment. Instead, use CSS top, right, bottom and left to position `<div>` elements that contain absolutely positioned text. Use two adjacent sides to position each `<div>` element. For example, top and left but never top, left, and bottom. CSS `text-indent` and `line-height` are useful in aligning text within HTML block elements.

11 Creating Graphic Novels/Manga/Comics (Fixed-Layout)

Graphic novels, manga, and comics (hereafter referred to as graphic novels) are similar to children's books, but present a unique challenge because they tend to be longer and have more complex content.

Graphic novels include a large amount of detail in images that displays on a 1920 x 1200 screen. To overcome this and other accessibility concerns, Amazon encourages the use of customized content and our Kindle Panel View feature, which optimizes the content for a high-resolution reading experience.

11.1 Metadata Guidelines

The OPF file specifies metadata necessary for fixed-layout books. For a demonstration, see the graphic novel example at www.amazon.com/kindleformat (under the **KindleGen Examples** heading).

Metadata	Description

Layout can be specified using one of the following metadata fields:
 	
`1) <meta property="rendition:layout">pre-paginated</meta>`
 	
`2) <meta name="fixed-layout" content="true"/>`
	Required. Identifies the book as having a fixed layout.
	
	Valid values for `rendition:layout` metadata are `reflowable` or `pre-paginated`. The default value is `reflowable`.
	
	Valid values for `fixed-layout` metadata are `true` or `false`. The default value is `false`.

`<meta name="original-resolution" content="1024x600"/>`	Required. Identifies the original design resolution of the content (“1024x600” is only an example). The pixel dimensions can have any positive integer value. These values must equal the overall aspect ratio of the original content.

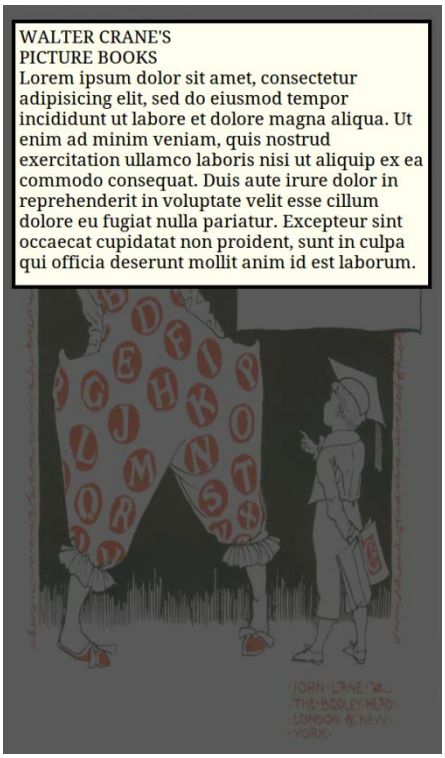
Orientation can be specified using one of the following metadata fields:
 	
`1) <meta property="rendition:orientation">landscape</meta>`
 	
`2) <meta name="orientation-lock" content="landscape"/>`
	Optional.
	
	Valid values for `rendition:orientation` metadata are `portrait`, `landscape`, or `auto`. Locks the orientation of the book to either portrait or landscape. If the value is `auto`, both portrait and landscape modes are supported. The default value is `auto`.
	
	Valid values for `orientation-lock` metadata are `portrait`, `landscape` or `none`. Locks the orientation of the content to either portrait or landscape. If the value is `none`, both portrait and landscape modes are supported. The default value is `none`.

`<meta name="RegionMagnification" content="true"/>`	Optional. Enables Kindle Panel View and Kindle Text Pop-Up. Valid values are `true` or `false`. The default value is `false`. Enabling this feature requires additional CSS instructions as specified in section 11.4.

Metadata	Description
`<meta name="primary-writing-mode" content="horizontal-rl"/>`	Required for Virtual Panels in Manga with right-to-left reading order; optional for other fixed-layout books. Defines page rendering order, reading mode, and reader navigation (including Kindle Text Pop-Up, Kindle Panel View, and Kindle Virtual Panels). Valid values are `horizontal-lr`, `horizontal-rl`, `vertical-lr`, and `vertical-rl`. The default value is `horizontal-lr`.
`<itemref idref="page-id" properties="page-spread-left"/>`	Required for Virtual Panels in Comics and Manga; optional for other fixed-layout books. Allows publishers to specify page layouts (double-pages, facing pages) at the page level and can vary throughout the book. The page properties should be specified in the `itemref` elements (child of `<spine>` element in the OPF file). Valid values are `page-spread-left`, `page-spread-right`, `facing-page-left`, `facing-page-right`, and `layout-blank`. The value `layout-blank` can be used independently or in conjunction with other valid values. Default value is `layout-blank`.
`<meta name="book-type" content="comic"/>`	Required for all graphic-novels; optional for other fixed-layout books. Removes reader functionality (e.g., share) which may not be relevant for certain books such as children's. Valid values are `children` or `comic`.

11.2 Text Guidelines

To display large amounts of text, Amazon suggests a hybrid text treatment that mixes the experiences of the graphic novels and children's books. Amazon recommends limiting the use of the hybrid text treatment to sections of text that are too wide to be magnified effectively. The hybrid text treatment should mimic the formatting of the text it represents in line-height, italic and/or boldface, and general appearance. This provides a better user experience.



Hybrid Text HTML Example:

```
<div id="pageXXX-magTargetParent" class="target-mag-parent">

<div class="target-mag-lb"></div>

<div id="pageXXX-magTarget" class="target-mag">

<div class="text">

<p>WALTER CRANE'S <br />PICTURE BOOKS <br /></p>

<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>

</div>

</div>

</div>
```

Hybrid Text CSS Example:

```
div.target-mag div.text{

  height: 100%;
```

```
  padding: 5px;

  background-color: #FFFFEF;

  font-size: 150%;

  font-family: "Georgia";

}
```

11.3 Image Guidelines

11.3.1 Asset Requirements

When optimized for the 2013 Kindle Fire HD 8.9", graphic novels should maintain a 1920 x 1200 aspect ratio. The image resolution will differ depending on the zoom factor required for Kindle Panel View. However, Amazon recommends planning for a high quality reading experience at 2X magnification. Images must be in the JPEG format. Amazon recommends using an image resolution of 300 ppi.

There are four standard zoom factors:

Zoom Factor	When to Use	Required Image Resolution
100%	Avoid using this zoom factor. It offers no magnification and poses an accessibility challenge for users.	1920 x 1200 pixels
125%	Only use this zoom factor when it is absolutely necessary to enlarge a very large panel. This allows the user to see a large action scene, but with the downside of limited enlargement.	2400 x 1500 pixels
150%	This is the default and preferred zoom factor. Use this zoom factor whenever possible.	2880 x 1800 pixels
250%	Only use this zoom factor on a two page spread image (two physical pages are displayed at once, and content appears especially small as a result). The downside is that the enlarged panel only represents a small portion of the original page.	4800 x 3000 pixels

11.3.2 Image Quality

Image quality for graphic novels requires that images follow the resolution standards listed in section 11.3.1, Asset Requirements, and maintain a consistent aspect ratio. Most importantly, optimize images for clarity of background art as well as readability of text. These two factors guarantee the highest quality for the graphic novel format.

11.4 Panel View (Region Magnification)

Panel View for graphic novels offers a unique reading experience. It addresses accessibility and lets users experience the flow of action on each page in a high-resolution, easy-to-use manner. Users can

dismiss Panel View at any time to view the entire page. For an example of Panel View, see the following images.

The user activates Panel View by double tapping a “tap target”. The active area (source element) is hidden and the Panel View (target element) is displayed.

To support Panel View, the following steps are required:

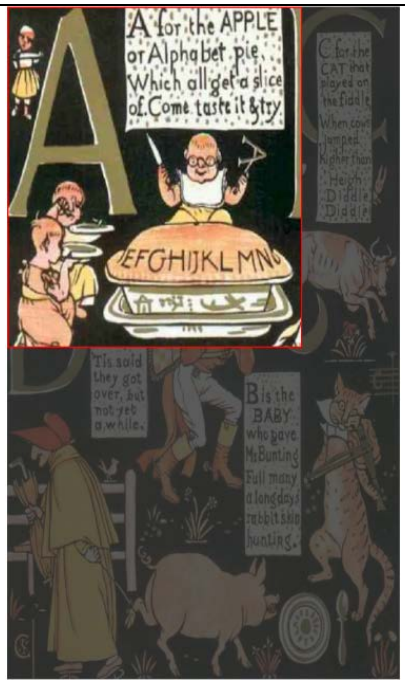
1. Set the tap target by creating a well-defined container ( ) element that contains an anchor (<a>) element. The provides the size and position of the tap target. The <a> is sized to fill the and must specify the `app-amzn-magnify` class. The anchor should also have the following attributes stored in a JSON object as part of the `data-app-amzn-magnify` value:

    a. `"targetId":"<string:elementId>"` = unique element id of the Panel View HTML element that represents the enlarged region

    b. `"ordinal":<integer:reading order>"` = reading order of the magnification areas (the order in which panels appear as part of the reading flow)

2. Create a target view panel element that is sized and positioned to display the action that best reflects the tap target.

Snapshot of fixed layout content



Snapshot of the same content with Panel View activated

Example:

```
<div>

<img src="images/hij.jpg" alt="Comic Book Images" class="singlePage" />

</div>
```

```
<div id="pagehij-1">

<a class="app-amzn-magnify" data-app-amzn-magnify='{"targetId":"pagehij-1-
magTargetParent", "ordinal":1}'></a>

</div>

…

<div id="pagehij-1-magTargetParent" class="target-mag-parent">

<div class="target-mag-lb">

</div>

<div id="pagehij-1-magTarget" class="target-mag">

<img  src="images/hij.jpg" alt="Comic Book Images"/>

</div>

</div>
```

11.5 Virtual Panels in Comics and Manga

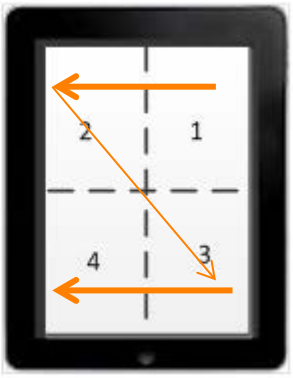
The Virtual Panels feature is activated for comics and manga books in the absence of publisher-provided panels. The `RegionMagnification` metadata is used to identify whether the publisher has provided panel information. If the publisher has designed the content with panels, the Kindle Virtual Panel view is not enabled.

By default, every page is divided into four panels based on the `primary-writing-mode` value. The examples below indicate the order of the panels.

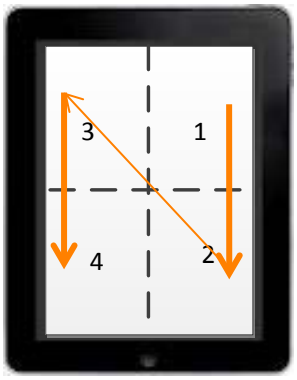
Example:

Portrait mode:

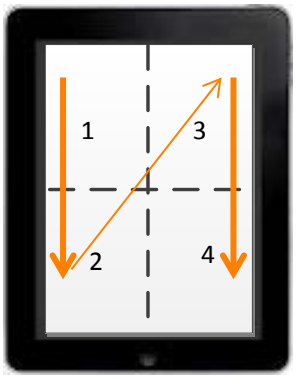
Horizontal-lr



Horizontal-rl

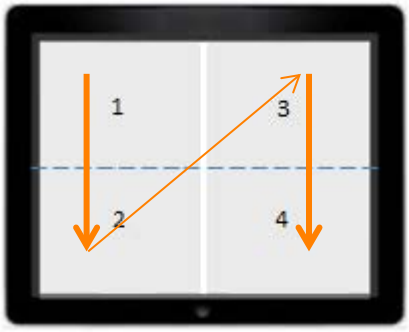


Vertical-rl

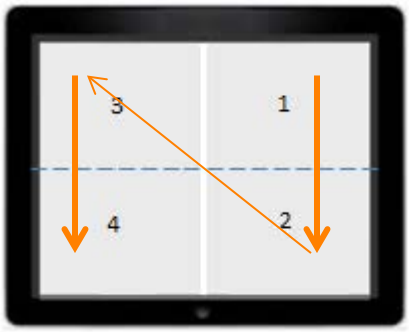


Vertical-lr

Landscape mode:



Vertical-lr and Horizontal-lr



Horizontal-rl and Vertical-rl

11.5.1 Requirement #1: Pairing Pages When Orientation-Lock Equals None

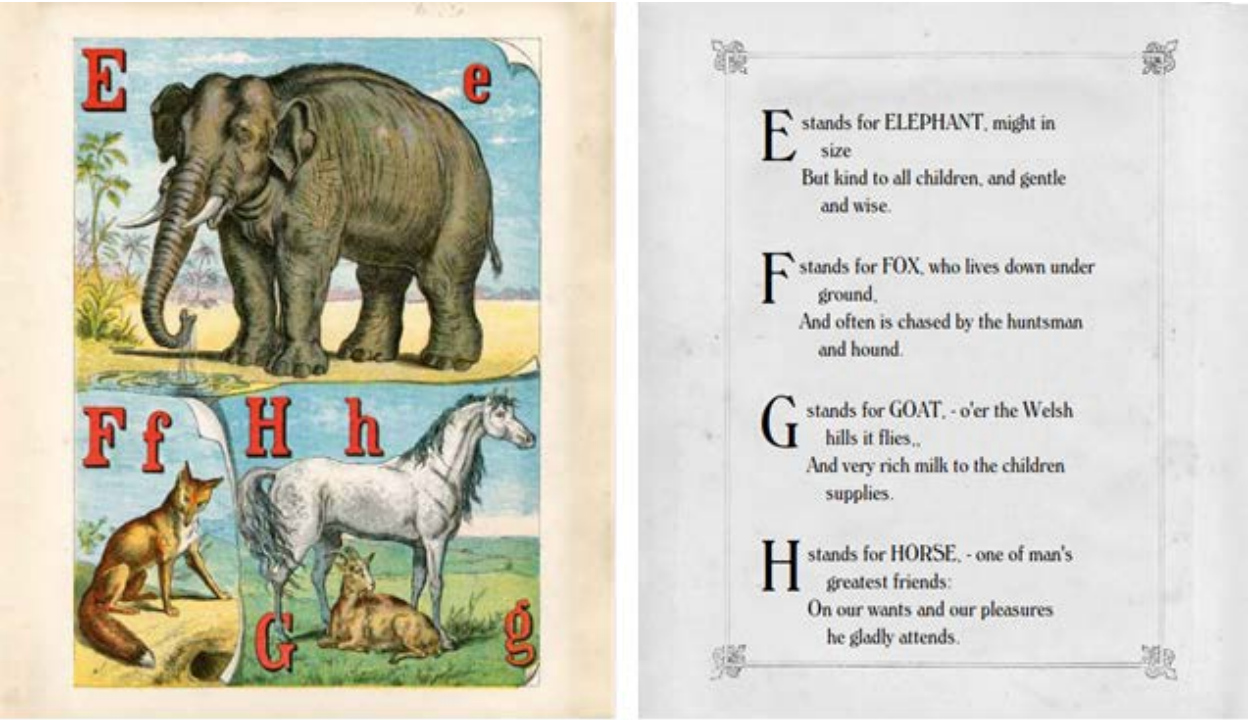
If orientation is not locked, content should be designed for both portrait and landscape mode. Also, every page is expected to have a defined definitive pair to support landscape orientation. In portrait orientation, the pairs are ignored.

All single pages should be tagged with the properties `facing-page-left` or `facing-page-right`. In this case, the renderer will add the spine while stitching two pages together.

Example:

The following example shows a double-page spread with visual separator in landscape mode.

```
<spine>

<itemref idref="page1" properties="facing-page-left"/>

<itemref idref="page2" properties="facing-page-right"/>

</spine>
```

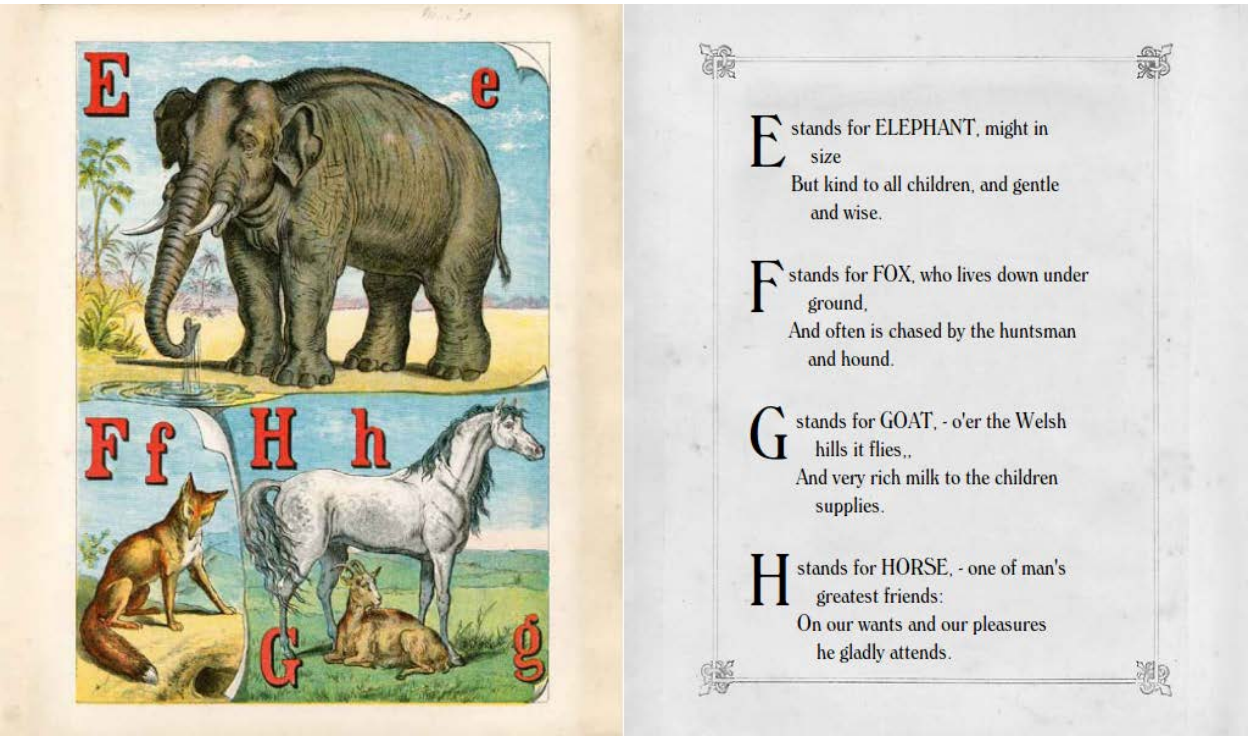


Double-page spreads should be tagged with the properties `page-spread-left` or `page-spread-right`. Every left page should have a right page associated and vice-versa. In this case, the renderer will not add the spine while stitching two pages together.

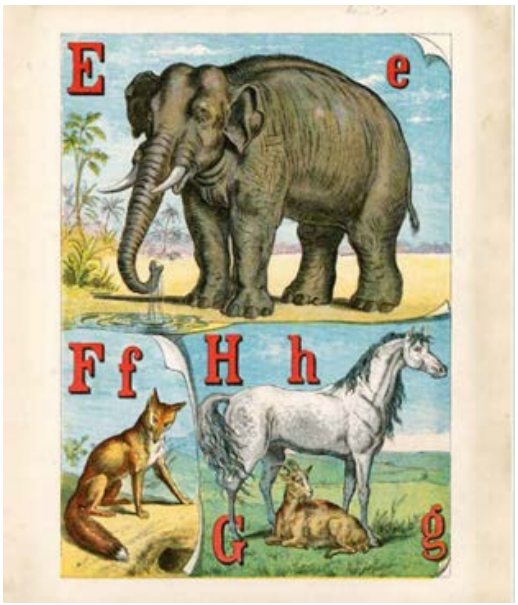
Example:

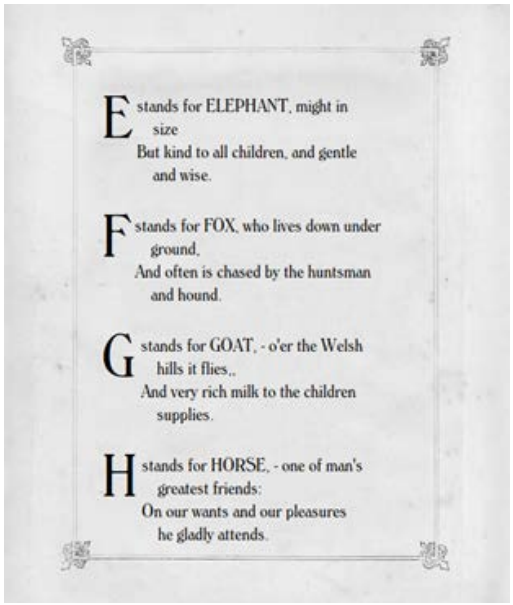
The following example shows a double-page spread without visual separator in landscape mode.

```
<spine>

<itemref idref="page1" properties="page-spread-left"/>

<itemref idref="page2" properties="page-spread-right"/>

</spine>
```



In portrait mode, the two pages will be rendered separately as shown below:





If none of the properties are specified, Kindle assumes `facing-page-left` and `facing-page-right` for alternate pages based on the book's writing mode.

The following example assumes `primary-writing-mode` equals `horizontal-lr` or `vertical-lr`.

Example:

```
<spine>

<itemref idref="page1" /> <!—assumed to be properties="facing-page-left" -->

<itemref idref="page2" /> <!—assumed to be properties="facing-page-right" -->

<itemref idref="page3" properties="page-spread-left"/> <!—double page spread’s left
viewport -->

<itemref idref="page4" properties="page-spread-right"/> <!—double page spread’s right
viewport -->

</spine>
```

The following example assumes `primary-writing-mode` equals `horizontal-rl` or `vertical-rl`.

Example:

```
<spine>

<itemref idref="page1" /> <!—assumed to be properties="facing-page-right" -->

<itemref idref="page2" /> <!—assumed to be properties="facing-page-left" -->

<itemref idref="page3" properties="page-spread-right"/> <!—double page spread’s right
viewport -->

<itemref idref="page4" properties="page-spread-left"/> <!—double page spread’s left
viewport -->

</spine>
```

In cases where a left page does not have an equivalent right page (or vice-versa), the publisher should insert a blank HTML page and add the property `layout-blank` to the page, unless it is the last page. Optionally, the blank page can include the book title and watermark by design.

Pages with the `layout-blank` property are only rendered in landscape mode and are ignored in portrait mode.

In some cases, the publisher may wish to insert a blank page that always renders in both portrait and landscape modes. In this instance, do not use the `layout-blank` property. Use the same facing (or double-page spread) rules as noted above and reference an image file that contains a “blank” jpeg.

The following example assumes `primary-writing-mode` equals `horizontal-lr` or `vertical-lr`.

Example:

```
<spine>

<itemref idref="page1" /> <!—assumed to be properties="facing-page-left" -->

<itemref idref="blank-page"  properties="layout-blank"/> <!—assumed to be
properties="facing-page-right". Ignored in portrait mode. -->
```

```
<itemref idref="page2" properties="page-spread-left"/> <!—double page spread’s left
viewport -->

<itemref idref="page3" properties="page-spread-right"/> <!—double page spread’s right
viewport -->

</spine>
```

11.6 Optimizing Content for the Graphic Novel Experience

11.6.1 Optimizing Tap Targets

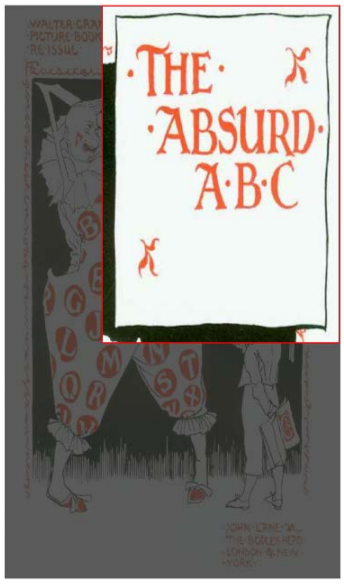
Tap targets should effectively cover 100% of the screen. This ensures that the user gets a magnified experience whenever the user double taps the graphic novel.

11.6.2 Optimizing View Panels

View panels should be 150% of the tap target by default. It is acceptable to use different size view panels to emphasize a specific action scene within the tap target.

Position view panels so that they convey where the primary action scene occurred on the original page. Typically, view panels are horizontally aligned to the left, center, or right edges and are vertically aligned to the top, center, or bottom edges.

When preserving context across multiple panels, it is acceptable to overlap slightly with other view panels.

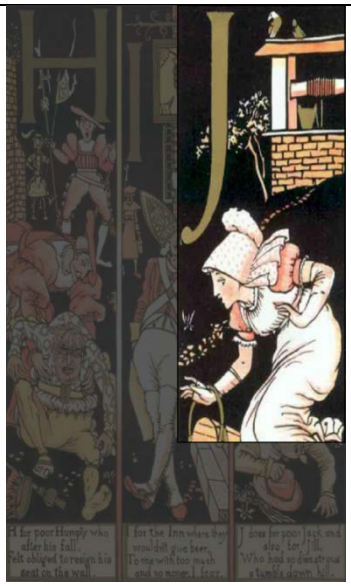


11.6.3 Optimizing for Wide or Tall Action Scenes

To use the default 150% default zoom factor, an action scene often needs to be divided into two view panels (typically a left and right or top and bottom). This is a better user experience than using a smaller zoom factor, because it preserves accessibility and gives the user a higher resolution reading experience.

Split the tap targets so that the first tap target is between 50 and 75% the width of the entire area, and the second tap target is the remaining amount necessary to reach 100%. This ensures when a user double taps an area near the middle of the action panel, they experience the first view panel first, and then the second view panel when they move forward.

To preserve the flow of the action, view panels should display a small amount of overlapped action.



First view panel of an action scene divided into two view panels

Second view panel of an action scene divided into two view panels

12 Creating Fixed-Format Books Without Pop-ups

In cases where a fixed format is needed to maintain the layout of the original book, but where text selection, dictionary, and highlighting are still required, fixed-format without pop-ups may be used. This conversion technique is ideal for content that has text large enough to be easily read without magnification, which can also be accomplished by redesigning the eBook to optimize the font size (on device, capital letters in body text must be at least 4 millimeters high).

The HTML and CSS of this format follows children's fixed-format specifications, including the CSS reset and embedded fonts. Fonts must be embedded, as is the standard for all fixed-format books.

12.1 Metadata Guidelines

The OPF file specifies metadata necessary for fixed-layout books. Unlike other fixed-layout books, this format does not include `meta name="book-type"` or `meta name="RegionMagnification"`.

Metadata	Description

Layout can be specified using one of the following metadata fields:
 	
`1) <meta property="rendition:layout">pre-paginated</meta>`
 	
`2) <meta name="fixed-layout" content="true"/>`
	Required. Identifies the book as having a fixed layout.
	
	Valid values for `rendition:layout` metadata are `reflowable` or `pre-paginated`. The default value is `reflowable`.
	
	Valid values for `fixed-layout` metadata are `true` or `false`. The default value is `false`.

Metadata	Description
`<meta name="original-resolution" content="1024x600"/>`	Required. Identifies the original design resolution of the content (“1024x600” is only an example). The pixel dimensions can have any positive integer value. These values must equal the overall aspect ratio of the original content.

Orientation can be specified using one of the following metadata fields:
 	
`1) <meta property="rendition:orientation">landscape</meta>`
 	
`2) <meta name="orientation-lock" content="landscape"/>`
	Optional.
	
	Valid values for `rendition:layout` metadata are `portrait`, `landscape`, or `auto`. Locks the orientation of the book to either portrait or landscape. If the value is `auto`, both portrait and landscape modes are supported. The default value is `auto`.
	
	Valid values for `orientation-lock` metadata are `portrait`, `landscape` or `none`. Locks the orientation of the content to either portrait or landscape. If the value is `none`, both portrait and landscape modes are supported. The default value is `none`.
	
	Amazon recommends only unlocking orientation when text is readable in both portrait and landscape.

`<meta name="primary-writing-mode" content="horizontal-rl"/>`	Optional. Defines page rendering order, reading mode, and reader navigation (including Kindle Text Pop-Up, Kindle Panel View, and Kindle Virtual Panels). Valid values are `horizontal-lr`, `horizontal-rl`, `vertical-lr`, and `vertical-rl`. The default value is `horizontal-lr`.

12.2 Text Guidelines

In this format, each page contains a background image that is larger than the screen size, so that the page renders a high-quality, readable image. The text can then be created with either of the two following methods and both methods can be used in the same book. With either method, capital letters in body text must be at least 4 millimeters high on device.

Live HTML Text

In this method, the background image is stripped of text and the majority of text in the book is rendered as live HTML text. The text can be repositioned as needed for a good eBook experience. This method also allows search and dictionary functionality.

Invisible Text Overlay

For text that is part of an image, or is irregularly sized, angled, or curved, you can capture that text as part of the background and use invisible overlay text (using the `opacity:0` property) to enable selection, dictionary lookup, and search. The invisible overlay text should be positioned directly over the corresponding text in the base image, and should be sized so that the highlight area of the invisible text corresponds to the text in the base image.

13 Creating Kindle Edition with Audio/Video Content

Currently, Kindle Edition with Audio/Video content is available on Fire tablets (2nd generation and later), iPad, iPhone, and iPod Touch. Audio and video content is not supported on Kindle e Ink devices (customers can read the book, but any audio or video is replaced with a message that it is not supported on this device).

KF8 features are not currently supported in Kindle Edition with Audio/Video content. The file delivered to Amazon should be a Mobi 7 EPUB with self-contained audio and video or a Mobi 7 .prc file with self-contained audio and video. (If delivering a .prc, make sure the file was created using the latest version of KindleGen available from www.amazon.com/kindleformat).

Amazon's Kindle Direct Publishing Platform (KDP) does not currently accept Kindle Edition with Audio/Video content.

Amazon does not currently accept any audio or video books with read-along content, which is defined as someone reading the full text or multiple pages of text from the book in either audio or video format.

When testing Kindle Edition with Audio/Video content, the audio and video cannot be previewed on Kindle Previewer or on Kindle devices and applications. To add audio and/or video content to your Kindle book, follow the guidelines and examples below.

13.1 Audio Requirements

Amazon recommends using stereo channels in the MP3 source where possible, because Kindle supports playing back audio in stereo. Use as high a bitrate as you need to hear the audio content appropriately; this is a judgment call. For good results, consider bitrates between 128 kbps and 256 kbps (kilobits per second). The maximum supported by Kindle is 320 kbps at variable bit rate.

13.2 Video Requirements

Since audio content can be part of the video content, Amazon recommends using stereo channels in your audio source where possible. Kindle supports playing back audio in stereo.

This is the **ideal** source spec:

Attribute	Setting
Dimensions	Widescreen: 704 x 396 (or any other widescreen ratio); Fullscreen: 640 x 480
Interlacing	Progressive
Color Space	4:2:0 YUV

Attribute	Setting
Video Codec	H.264 (recommended), MPEG-2
Video Mode	VBR (recommended) or CBR
Video Bit Rate	2500 kbps or higher recommended
Key Frame Interval	2 or 4 seconds recommended
Audio Codec	MP3
Audio Bit Rate	256 kbps or higher recommended
Audio Sample Rate	48 kHz (recommended), 44.1 kHz

The following container formats are acceptable:

Container	File Extensions	Mime Type	RFC
MP4	.mp4	video/h264	RFC3984
MPEG-2 video file	.mpg, .mpeg	video/mpeg	RFC2045, RFC2046
MPEG-2 program stream	.ps	video/mp2p	RFC3555
MPEG-2 transport stream	.ts	video/mp2t	RFC3555

Will not work: any other video codec (such as Windows Media, Apple ProRes), AC3 audio, audio >2 channels

13.3 File Requirements

13.3.1 Multimedia Directory

When adding audio and video files to an eBook, create an “audiovideo” directory for storing these files. When referring to the audio or video file, include the directory name (**Example:** “audiovideo/filename”) in the HTML.

13.3.2 Confirm Correct Mime-Type

When specifying video and audio files in the OPF, make sure that they have the correct mime-types, depending on the extensions used. (**Example:** MP4 video files should have a mime-type of “video/mp4” and not “audio/mpeg”.)

13.3.3 File Size

Limit the combined file sizes of all audio and video files to 600 MB or less for each title. If the files are larger than 600 MB, manually transcode them to reduce the file size(s). (The total maximum audio/video file size that can be converted from EPUB via KindleGen is 650 MB.)

Limit the number of individual audio and video files within each title to 1,000 or fewer.

13.4 Navigation Guidelines

13.4.1 Including an Audio and Video TOC

All Kindle Edition with Audio/Video content must have a TOC that begins with "List of Audio and Video." This line should be bold. On the next line, begin an indented list of hyperlinks to each audio and video file. The text of the link should include the file description, with the file duration in parentheses.

Use these guidelines for all audio and video files longer than 10 seconds that a user might want to see listed.

Here is an example of how the code below would display in the TOC:

List of Audio and Video

This is my video (5:01)

This is my audio (1:10)

This is the corresponding code for the example above:

```
<video id="video_1" src=" audiovideo//movie.mp4" controls poster="start.jpg"
title="This is my video (5:01)">

<br/><br/><br/>"There is video content at this location that is not currently supported for your device. The caption for this content is displayed below."<br/><br/><br/>

</video>

<br>This is my video (5:01)</br>

<audio id="audio_1" src=" audiovideo//audio.mp3" controls title="This is my audio
(1:10)">

<br/><br/><br/>"There is audio content at this location that is not currently supported for your device. The caption for this content is displayed below." <br/><br/><br/>

</audio>

<strong>This is my audio (1:10)</strong>
```

13.4.2 Including Audio and Video Assets in NCX File

When creating eBooks with audio and video content, Amazon requires the creation of an NCX file that points to the audio and video assets. This file should list all video and audio files in reading order, with links to where they occur in the book. For descriptions of the audio and video files, reuse the same audio and video metadata. (**Example:** A link to the video clip in section 13.7.3 would say “How to create Kindle content (5:01)”.) This information should be embedded in the NavList portion of the NCX file.

13.5 Text Guidelines

KF8 features are not currently supported in Kindle Edition with Audio/Video content. This includes, but is not limited to:

- Drop caps
- Drop shadow
- Embedded fonts
- Nested tables and merged cells
- Floating elements
- Borders

13.6 Image Guidelines

KF8 features are not currently supported in Kindle Edition with Audio/Video content. This includes, but is not limited to:

- Background images
- Scalable Vector Graphics (SVG)
- Floating images

13.7 Audio and Video Guidelines

13.7.1 Adding Audio

Streaming audio is not supported at this time. Use embedded audio instead.

To embed an audio file inside a Kindle book, add a standard HTML 5 tag such as the following:

Example:

```
<audio id="audio_1" src="audio.mp3" controls title="Audio about ...">

<br/><br/><br/>"There is audio content at this location that is not currently supported for your device. The caption for this content is displayed below."<br/><br/><br/>

</audio>
```

- **src tag:** (Required) Identifies the embedded audio file, which must be in MP3 format.
- **title tag:** (Optional) Identifies the description of the audio.
- **controls tag:** (Required, unless you provide an image for use in starting the video playback) Tells the Kindle application to display controls for the embedded audio.
- **text content:** (Required) Devices that do not support audio content display the text between the `<audio>` and `</audio>` tags. If users view this eBook on a device that does not support audio, they see this text instead. (**Example:** “There is content at this location that is not currently supported for your device. The caption for this content is displayed below.”)

- **id tag:** (Optional) Must be unique to the document if it is used.

13.7.2 Adding Video

Streaming video is not supported at this time. Use embedded video instead.

To embed a video inside a Kindle book, add a standard HTML 5 tag such as the following:

Example:

```
<video id="video_1" src="movie.mp4" controls poster="start.jpg" title="Video about
...">

<br/><br/><br/>"There is video content at this location that is not currently supported for your device. The caption for this content is displayed below."<br/><br/><br/>

</video>
```

- **src tag:** (Required) Identifies the embedded video file.
- **title tag:** (Required) Identifies the description of the video.
- **poster tag:** (Required) Identifies the placeholder image file. Users see the placeholder in the eBook before the video is played. The placeholder could be the first frame of the video or a representative frame, depending on your preference. (If this file is not specified, a blank black image is displayed.)
- **controls tag:** (Required, unless you provide an image for use in starting the video playback) Tells the Kindle application to display controls for the embedded video.

  Note: The Kindle application may render a play button on top of the poster frame. It appears in the middle of the frame.
- **text content:** (Required) Devices that do not support video content display the text between the `<video>` and `</video>` tags. If users view this eBook on a device that does not support video, they see this text instead. (**Example:** “There is content at this location that is not currently supported for your device. The caption for this content is displayed below.”)
- **id tag:** (Optional) Must be unique to the document if it is used.

13.7.3 Audio and Video Metadata Required

Amazon requires that publishers (or their conversion houses) provide a description of the audio and video file, and the duration of the file in minutes and seconds, in the HTML immediately after the audio and video file is specified.

Example:

```
<p align="center" style="text-indent:0px">

<video id="video_1" src="movie.mp4" controls poster="start.jpg" title="How to create
Kindle content (5:01)">

<br/><br/><br/>"There is video content at this location that is not currently supported for your device. The caption for this content is displayed below."<br/><br/><br/>
```

```
</video>

<strong>How to create Kindle content (5:01)</strong>

</p>
```

13.7.4 Providing Descriptive Media Captions

Media captions describe the audio and video files to the user. Here are some general guidelines:

- Captions should not be generic. They should describe the media content they are referencing.

  These media captions are not a good user experience:

  1. Media 1
  2. Track 1
  3. Audio 1
  4. Video 1

  These media captions describe the content:

  1. Introduction by the Author
  2. The Making of the Movie

- Media captions cannot include file extensions (.mp3, .mp4, etc.).

13.7.5 Adding Images with Play Controls

It is possible to tag images so that they can be played by clicking on them. The minimum pixel width and height for such images is 45 pixels by 45 pixels.

To add play controls to the image, superimpose the Amazon PLAY icon onto the lower right-hand side of any image via Photoshop or similar program. Then add the following tag to the HTML (in this example, the audio file has an `id` attribute of "`audio1`" and no `controls` tag):

Example:

```
<a onclick="play(this);" data-AmznAudioTag="audio1"><img src="play.jpg"/></a>
```

13.8 Custom Sample File Required

Amazon requires that publishers create and supply a custom sample for each Kindle Edition with Audio/Video. The sample file should include a full TOC and an audio/video list, with live links to only the content in the sample file.

The sample file should include at least one of each type of media available in the full file, including both audio and video, if applicable.

The sample file must have a “Buy It Now” link added to the end, or where appropriate.

14 Creating Dictionaries

A dictionary is a Kindle eBook (.mobi file) with extra tags added to support search and lookup functionality. Dictionary eBooks:

- Contain a primary index: a list of words or sentences that are sorted in alphabetical order. Readers can search quickly in this list by typing the beginning of the word and selecting the desired entry.
- Are marked as dictionaries. The input and output languages of the dictionary must be defined properly so that Kindle devices can use the dictionary for in-book lookup.

For example, an English (monolingual) dictionary lists English as both the input and output language. A French-English dictionary lists French as the input language and English as the output language. To build a bidirectional bilingual dictionary (example: Spanish-French and French-Spanish), you must create two separate eBooks: one for Spanish-French and one for French-Spanish.

A Kindle dictionary should have all the same components as a normal Kindle eBook. There should be an OPF file and HTML files with CSS. Every dictionary should have:

- A cover image
- A copyright page
- Any relevant front or back matter (explanations of symbols, appendices, etc.)
- Definitions of words (this is the bulk of the file)

14.1 Metadata Guidelines

The OPF file of a dictionary is similar to that of other Kindle books, except that it contains specialized metadata tags in the `<x-metadata>` section. These extra tags in the OPF file set the source language and the target language for the dictionary. If the dictionary has multiple indices, the OPF file also specifies the name of the primary lookup index.

- The `<DictionaryInLanguage>` element contains the ISO 639 language code for the language of the books this dictionary is designed to be used on. For a Spanish-French dictionary, the input language is Spanish.
- The `<DictionaryOutLanguage>` element contains the ISO 639 language code for the language of the definitions returned by the dictionary. For a Spanish-French dictionary, the output language is French.
- The `<DefaultLookupIndex>` element indicates the index that will open first when the dictionary is used for lookup from another eBook. The default index must be specified if the dictionary has more than one index. The index name that is wrapped in the `<DefaultLookupIndex>` tags in the OPF file also should appear as the value of the `name` attribute in the `<idx:entry>` elements in the content of the dictionary (see section 14.3.3).

As an example, for a Spanish-French dictionary, the input language code would be **`es`**; the output language code would be **`fr`**, and the primary index might be named **`Spanish`**.

Example: (Bilingual Dictionary Metadata)

```
<x-metadata>

<DictionaryInLanguage>es</DictionaryInLanguage>
```

```
<DictionaryOutLanguage>fr</DictionaryOutLanguage>

<DefaultLookupIndex>Spanish</DefaultLookupIndex>

...

</x-metadata>
```

For a monolingual dictionary, the same language code must appear twice: once to identify the input language, and again to identify the same language as the output language. To identify a regional variant for the source and/or target languages, a regional suffix may be appended to the ISO 639 code. For example, **en-gb** indicates British English, while **en-us** indicates US English.

Example: (Monolingual Dictionary Metadata, Regional Variant)

```
<x-metadata>

<DictionaryInLanguage>en-us</DictionaryInLanguage>

<DictionaryOutLanguage>en-us</DictionaryOutLanguage>

<DefaultLookupIndex>headword</DefaultLookupIndex>

...

</x-metadata>
```

14.2 Text Guidelines

14.2.1 Dictionary Entry Template

A simple, clean format works best for in-book lookup. Amazon recommends these dictionary content and formatting features for a high-quality user experience:

- The headword (word being defined) should come first in the entry, and should be distinguished from surrounding content (on its own line, left flush, in bold).
- Every dictionary entry should contain a definition (or translation, for bilingual dictionaries).
- Horizontal rules should appear between each entry.
- Each alphabet letter section should begin on a new page.
- Images should be avoided (see section 9.4 for image constraints).
- Tables should not be used (see section 9.5 for table constraints).
- Font color, size and typeface should not be forced (see section 9.3 for text guidelines).

14.3 Basic Dictionary HTML

14.3.1 Format

Dictionaries for Kindle must be in Mobi 7 format, not in KF8. For this reason, the dictionary layout should use a single-column format. Multiple columns and sidebars are not supported in Mobi 7 format.

14.3.2 Frameset element

All dictionaries must have an **<mbp:frameset>** element as the first child of the <body> element. This frameset element contains all of the **<idx:entry>** elements of the dictionary.

The namespace for this **<mbp:frameset>** element is xmlns:mbp="http://www.kreutzfeldt.de/mmc/mbp" and it must be declared in the root <html> element of the XHTML document.

Example:

```
<html xmlns:math="http://exslt.org/math" xmlns:svg="http://www.w3.org/2000/svg"
xmlns:tl="http://www.kreutzfeldt.de/tl"

xmlns:saxon="http://saxon.sf.net/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:cx="http://www.kreutzfeldt.de/mmc/cx"
xmlns:dc="http://purl.org/dc/elements/1.1/"

xmlns:mbp="http://www.kreutzfeldt.de/mmc/mbp"
xmlns:mmc="http://www.kreutzfeldt.de/mmc/mmc"
xmlns:idx="http://www.mobipocket.com/idx">

<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head>

<body>

<mbp:frameset>

<idx:entry name="english" scriptable="yes" spell="yes">

<idx:short><a id="1"></a>

       <idx:orth value="aardvark"><b>aard•vark</b>

       <idx:infl>

       <idx:iform value="aardvarks"></idx:iform>

       <idx:iform value="aardvark's"></idx:iform>

       <idx:iform value="aardvarks'"></idx:iform>

</idx:infl>

</idx:orth>

<p> A nocturnal burrowing mammal native to sub-Saharan Africa that feeds exclusively
on ants and termites. </p>

</idx:short>

</idx:entry>

[...other entries…]

</mbp:frameset>

</body>

</html>
```

14.3.3 Headword Index

In order to make an alphabetical index of headwords, it is necessary to use special tags that are not standard HTML. The source still will be valid XHTML with these added <idx> mark-ups.

<idx:entry>..</idx:entry>

The <idx:entry> tag marks the scope of each entry to be indexed. In a dictionary, each headword with its definition(s) should be placed between <idx:entry> and </idx:entry>. Any type of HTML may be placed within this tag.

The <idx:entry> tag can carry the name, scriptable, and spell attributes. The name attribute indicates the index to which the headword belongs. The value of the name attribute should be the same as the Default Lookup Index name listed in the OPF. The scriptable attribute makes the entry accessible from the index. The only possible value for the scriptable attribute is "yes". The spell attribute enables wildcard search and spell correction during word lookup. The only possible value for the spell attribute is "yes".

Example:

```
<idx:entry name="english" scriptable="yes" spell="yes">
```

The <idx:entry> tag also may carry an id attribute with the sequential id number of the entry. This number should match the value of the id attribute in an anchor tag used for cross-reference linking:

Example:

```
<idx:entry name="japanese" scriptable="yes" spell="yes" id="12345">

<a id="12345"></a>
```

The entry id number is not used for in-book lookup; instead, the wordform entity to be indexed for lookup must be contained in the <idx:orth> element as follows.

<idx:orth>..</idx:orth>

The <idx:orth> tag is used to delimit the label that will appear in the index list and that will be searchable as a lookup headword. This is the text that users can enter in the search box to find an entry.

Example:

```
<idx:orth>Label of entry in Index</idx:orth>
```

Here is an example of an extremely simple entry that could be part of an English dictionary. From this example code, the word "chair" would appear in the index list and would be searchable by users.

Example:

```
<idx:entry>

<idx:orth>chair</idx:orth>

A seat for one person, which has a back, usually four legs, and sometimes two arms.

<idx:entry>
```

The `value` attribute can be used on the `<idx:orth>` tag to include a hidden label in the entry. This attribute maintains lookup functionality in the presence of the special formatting that commonly appears on headwords in dictionaries.

Example:

```
<idx:orth value="Hidden Label of entry in Index">Display format</orth>
```

If the headword should be displayed in the dictionary with a superscripted number to indicate homographs, with a registered trademark symbol, with middle dots to separate syllables, or with any other added symbols, this special formatting should appear on the text between the `<idx:orth>` tags, but not on the text in the `value` attribute. The text in the `value` attribute should match exactly the form to be used for lookup. If a `value` attribute is not supplied, then the entity between the `<idx:orth>` tags will be indexed for lookup. If middle dots, superscripted numbers, or any other symbols are included in the text between the `<idx:orth>` tags, then *in-book lookup will fail* unless a hidden label with the lookup form is supplied in the `value` attribute.

Example:

```
<idx:orth value="Amazon">A•ma•zon®<sup>3</sup></orth>
```

If the dictionary uses more than one orthographic script, then the `format` attribute on the `<orth>` tag can be used to identify each script for building the index.

Example:

```
<idx:orth format="script name">
```

Along with this primary index of headwords for all entries in the dictionary, in-book lookup also requires a supplementary index of inflected forms for each headword. To build the hidden inflection index, additional data should be nested within the `<idx:orth>` tag as follows.

14.4 Inflections for Dictionaries

Dictionaries should be built so that multiple inflected forms of a single root word all access the same entry. A complete list of inflected wordforms should be provided for every headword. If an entry uses multiple orthographies, then separate inflections must be provided for each orthography.

14.4.1 Inflection Index

To construct the hidden inflection index, the inflected wordform data should be wrapped within `<idx:infl>` and `<idx:iform />` tags nested inside the `<idx:orth>` element. This index will not be directly searchable by the user, but instead will be used for in-book lookup.

<idx:infl>..</idx:infl>

The `<idx:infl>` element may contain multiple `<idx:iform />` elements. The `<idx:iform />` elements are always empty elements, and are used only to carry attributes, not visible content. The `value` attribute indicates the inflected forms that make up the inflection index.

Example:

```
<idx:orth>record

<idx:infl>
```

```
<idx:iform value="records" />

<idx:iform value="recording" />

<idx:iform value="recorded" />

</idx:infl>

</idx:orth>
```

The <idx:infl> tag, the <idx:iform /> tag, and the value attribute are mandatory. The <idx:infl> element also may carry an optional inflgrp attribute to denote part of speech, and the <idx:iform /> element may carry an optional name attribute to indicate the inflection paradigm category. For languages that use extensive inflection, including these optional categories will expand the size of the inflection index, and may result in slower performance during word lookup.

Example:

```
<idx:orth>record

<idx:infl inflgrp="noun">

<idx:iform name="plural" value="records" />

</idx:infl>

<idx:infl inflgrp="verb">

<idx:iform name="present participle" value="recording" />

<idx:iform name="past participle" value="recorded" />

<idx:iform name="present 3ps" value="records" />

</idx:infl>

</idx:orth>
```

The values listed as attributes of the <idx:iform /> tag will be invisible to the user, but rather will provide the information needed to redirect from inflected forms to the associated headwords during in-book lookup. To inform the user about parts of speech or inflection paradigms, additional text should be included in the body of the entry (i.e., alongside the definition and examples).

<idx:key>..</idx:key> (DEPRECATED)

Like the <idx:infl> tag, the <idx:key> tag is designed to enable search for an entry in the index by means of an alternative lookup wordform. However, the presence of <idx:key> tags in a Kindle dictionary can create instability in the lookup functionality, and can interfere with the operation of the exact-match parameter (see section 14.4.2). For these reasons, the use of <idx:key> tags in Kindle dictionaries is deprecated. Instead, <idx:infl> and <idx:iform /> tags should be used to wrap the alternative lookup forms.

14.4.2 Exact-match parameter

By default, the Kindle device uses a fuzzy algorithm for matching diacritics during word lookup. Languages that use contrastive diacritics to distinguish between distinct word forms should use the `exact="yes"` attribute in the `<idx:iform />` tag to force exact match of diacritics during lookup.

Example:

```
<idx:entry name="spanish" scriptable="yes" spell="yes">

<a id="12345"></a>

<idx:orth value="uña"><b>uña</b>

<idx:infl>

<idx:iform value="uñas" exact="yes" />

</idx:infl>

</idx:orth>
```

Setting the `exact` parameter to `"yes"` forces the device to match *uñas* to the headword *uña* (‘fingernail’), and prohibits a match to *una* (‘one’).

14.5 Building a Dictionary with Kindlegen

When building a dictionary with Kindlegen via the command line, use the following syntax:

```
kindlegen.exe [filename.opf] -c2 –verbose -dont_append_source
```

If the dictionary entries are contained in a single, very large XHTML file, then Kindlegen may not be able to build the dictionary. If the dictionary fails to build, this problem may be resolved by splitting the dictionary content into two or more XHTML files.

For more guidance on using Kindlegen, please see section 2.2.1.

14.6 QA Standards—Testing Kindle Dictionaries

14.6.1 Format Testing

Amazon recommends verifying that the converted dictionary is properly formatted to provide a good visual experience for the user. Check the formatting of the definitions by paging through the dictionary and reading several definitions. (The format of the dictionary may be checked using Kindle Previewer or any Kindle device; however, lookup testing requires the use of an e Ink device.)

- Check words for unsupported characters, broken or joined words, proper display of accented characters, symbols, pronunciation guide, etc.
- Check that there are no typos.
- Check that links (if present) are working correctly. (Links will be disabled in the in-book lookup window, but links should function inside the dictionary itself.)
- If any images are used, check that these images are clear and readable.
- Check that the font color and typeface are not forced.

14.6.2 Lookup Testing

Amazon recommends verifying that definitions return correctly when the dictionary is used to look up words in other books. This component of testing can be done only with e Ink devices (not including Previewer), because only e Ink devices allow the user to set the default dictionary for lookup.

- Sideload the dictionary onto the e Ink device. To do this, connect the Kindle to your computer with a USB-to-mini-USB cord. Your computer should detect the device. In the window that pops up, you should see a folder called Documents. Put the dictionary file into this folder, and then eject your Kindle from the computer.

- Set the test dictionary as the default dictionary for lookup:
    - Kindle Paperwhite: Go to **Home > Menu > Settings > Device Options > Language and Dictionaries > Dictionaries > [Source Language]**

- Look up a variety of words to see what definition is returned. Open a title other than the dictionary, select a word, and note the definition returned in the lookup window. If lookup fails entirely, check for errors in the HTML tagging.

  Suggestions of words to look up include:
    - Conjugations of regular and irregular verbs.
        - Example: walk, walks, walked, walking; go, goes, went, gone, going.
    - Nouns, adjectives, adverbs and their conjugations/declensions.
        - Example: desk, desks; wolf, wolves; hot, hotter, hottest.
    - Grammatical and punctuation conventions commonly used in the language.
        - Example: contractions, elisions, verbs with clitic pronouns.

- Check the index view of the dictionary. To do this, open the dictionary and start typing a word in the Search box. An alphabetized list of headwords should appear and should update dynamically based on which letters are typed. Selecting a headword from the index list should redirect the user to the dictionary entry for that headword.

Part IV. Appendices

15 Appendix A: Enhanced Typesetting

15.1 About Enhanced Typesetting

Enhanced Typesetting is a series of typographical and layout features that are automatically enabled on Kindle books. These enhancements improve readability and enforce more consistent display behavior across Kindle reading platforms, including Kindle devices, Fire tablets, and free Kindle reading apps for Android and iOS. Some Enhanced Typesetting features include:

- Drop caps that dynamically adjust with font size
- Hyphenation and smoother word spacing
- Kerning and ligature improvements

15.2 Support for Enhanced Typesetting

If Enhanced Typesetting is enabled for your book, you'll see "Enhanced Typesetting: Enabled" on that book's detail page. We are continuously working to make Enhanced Typesetting compatible with more titles and will automatically enable Enhanced Typesetting enhancements for your book when possible.

Read more about Enhanced Typesetting at http://www.amazon.com/betterreading.

16 Appendix B: HTML and CSS Tags Supported in Kindle Format 8

16.1 HTML Support Table

HTML Tag	Supported on KF8-Enabled Devices & Apps
<!--...-->	Yes
<!DOCTYPE>	Yes (not on e Ink)
<?xml?>	Yes
<a>	Yes
<address>	Yes
<article>	Yes
<aside>	Yes
<b>	Yes
<big>	Yes (deprecated – recommend using CSS instead)
<blockquote>	Yes
<body>	Yes
 	Yes
<caption>	Yes
	Yes (deprecated – recommend using CSS style text-align:center instead)
<cite>	Yes
<code>	Yes
<col>	Yes
<dd>	Yes
<del>	Yes
<dfn>	Yes
	Yes
<dl>	Yes
<dt>	Yes
	Yes
<font>	Yes (deprecated – recommend using CSS instead)
<figcaption>	Yes

HTML Tag	Supported on KF8-Enabled Devices & Apps
<figure>	Yes
<footer>	Yes
<h1>	Yes
<h2>	Yes
<h3>	Yes
<h4>	Yes
<h5>	Yes
<h6>	Yes
<head>	Yes
<header>	Yes
<hgroup>	Yes
<hr>	Yes
<html>	Yes
<i>	Yes
<img>	Yes
<ins>	Yes
<kbd>	Yes
<li>	Yes
<link>	Yes
<mark>	Yes
<menu>	Yes
<ol>	Yes
<output>	Yes
<p>	Yes
<pre>	Yes
<q>	Yes
<rp>	Yes
<rt>	Yes
<samp>	Yes
<section>	Yes

HTML Tag	Supported on KF8-Enabled Devices & Apps
<small>	Yes
<source>	Yes
	Yes
	Yes
<style>	Yes
<strike>	Yes
<sub>	Yes
<sup>	Yes
<table>	Yes
<tbody>	Yes
<td>	Yes
<tfoot>	Yes
<th>	Yes
<thead>	Yes
<time>	Yes
<title>	Yes
<tr>	Yes
<u>	Yes
<ul>	Yes
<var>	Yes
<wbr>	Yes
<nav>	Yes
<summary>	Yes
<video>	No
<audio>	No

The following HTML tags are not supported in the Kindle format:

- Canvas
- Command
- Datalist
- Script (reserved for Amazon use only)

- Base
- Form
- Eventsource
- KeyGen
- Input
- Embed (Only SVG is supported)
- Object (Only SVG is supported)
- Param
- Noscript
- IFrame
- Marquee

16.2 CSS Support Table

CSS Attribute	Supported on KF8-Enabled Devices & Apps
`/*Comment*/`	Yes
`@import`	Yes
`@charset`	Yes
`@font-face`	Yes
`*`	Yes
`E`	Yes
`E.class`	Yes
`E#id`	Yes
`E:link`	Yes
`E:visited`	No
`margin`	Yes
`padding`	Yes
`width`	Yes
`height`	Yes
`float`	Yes
`clear`	Yes
`display`	Yes
`min-width`	Yes

CSS Attribute	Supported on KF8-Enabled Devices & Apps
max-width	No
min-height	Yes
max-height	No
clip	Yes
visibility	Yes
border	Yes
border-color	Yes
border-style	Yes
border-width	Yes
border-top	Yes
border-right	Yes
border-bottom	Yes
border-left	Yes
border-radius	Yes
line-height	Yes
vertical-align	Yes
position	Yes
top	Yes
right	Yes
bottom	Yes
left	Yes
z-index	Yes
list-style	Yes
list-style-image	Yes
list-style-position	Yes
list-style-type	Yes
opacity	Yes
background	Yes
background-attachment	Yes
background-color	Yes

CSS Attribute	Supported on KF8-Enabled Devices & Apps
background-image	Yes
background-position	Yes
background-repeat	Yes
background-clip	Yes
background-origin	Yes
background-size	Yes
font	Yes
font-family	Yes
font-size	Yes
font-style	Yes
font-variant	Yes
font-weight	Yes
text-align	Yes
text-decoration	Yes
text-indent	Yes
text-transform	Yes
letter-spacing	Yes
word-spacing	Yes
white-space	Yes
text-shadow	Yes
text-overflow	Yes
word-wrap	Yes
direction	Yes
border-collapse	Yes
border-spacing	Yes
caption-side	Yes
empty-cells	Yes
outline	No
outline-color	No
outline-style	No

CSS Attribute	Supported on KF8-Enabled Devices & Apps
`outline-width`	No
`outline-offset`	Yes
`width`	Yes
`height`	Yes
`device-width`	Yes
`device-height`	Yes
`device-aspect-ratio`	Yes
`color`	Yes
`color-index`	Yes
`monochrome`	Yes

The following CSS selectors, attributes, and properties are not supported in the Kindle format:

- E + F (Direct adjacent)
- E ~ F (Indirect adjacent)
- E:first-child
- E:first-of-type
- E:last-child
- E:last-of-type
- E:only-child
- E:only-of-type
- E:nth-child
- E:nth-last-child
- E:nth-of-type
- E:nth-last-of-type
- E::first-letter
- E::first-line
- E::before
- E::after
- Counter-increment
- Counter-reset

17 Appendix C: Media Queries

Media queries are blocks of CSS code that allow content creators to apply different styles to specific Kindle devices (or a group of devices) using only one style sheet. Amazon has implemented media queries as a way to help content creators build a better customer experience across devices.

Amazon recommends that content creators only use media queries if they solve a problem or provide a better customer experience. For example, you can use media queries to:

- Create custom drop cap solutions for specific devices or groups of devices.
- Change light-colored text (yellow, baby blue, pink, etc.) to darker colors that provide better contrast on e Ink while retaining the original color on tablet devices.
- Increase the font size for fixed-format text pop-ups on e Ink devices separately from tablets to accommodate the difference in screen sizes.
- Display colored borders on e Ink and colored backgrounds on tablets independently from each other, allowing you to better replicate the print experience on tablets without sacrificing the reading experience on e Ink devices.

This section describes ways that you can use media queries to customize the reading experience on Kindle e Ink devices, Fire tablets, and iPads. You can also use these same principles across all Kindle platforms for devices of all aspect ratios.

Media queries are part of the W3 standard. For more information, visit http://www.w3.org/TR/css3-mediaqueries/

17.1 Media Query Guidelines

Support for two new media types enables content creators to use specific CSS based on the Mobi or KF8 file format: `amzn-mobi` and `amzn-kf8`.

- For KF8 CSS styles, use the media query `@media amzn-kf8`. This is only applied for the KF8 format.
- For Mobi CSS styles, use the media query `@media amzn-mobi`. This is only applied for the Mobi format.

The `@media screen` and `@media all` styles continue to apply to both KF8 and Mobi.

17.1.1 Use Correct CSS Syntax

Media queries consist of two parts: (1) the selector, which specifies the conditions of the media query; and (2) the declaration block, which is rendered when the conditions of the media query are met.

In the following example, the blue background color is only applied if the format of the book is KF8 and the aspect ratio of the device is 1280 x 800.

Example:

```
/* Kindle Fire (All) Formatting. */
```

```
@media amzn-kf8 and (device-aspect-ratio:1280/800) {

  .blue_background {

    background-color: blue;

  }

}
```

17.1.2 Add a CSS Comment Before Each Media Query

Amazon recommends that you add a CSS comment before each media query to clarify which device you are targeting. (A CSS comment starts with /* and ends with */.)

Example:

```
/* Kindle Fire (All) Formatting */
```

CSS comments are invisible to the customer, but they make the code much easier to navigate and troubleshoot for anyone who works on the file.

17.1.3 Always Use Non-Media Query Code To Target E-Ink Devices

Always optimize your non-media query ("default") code for Kindle e Ink devices (including Kindle Voyage and Kindle Paperwhite). The default code contains the CSS values that will appear on an E-reader when none of the media queries match that particular device.

17.1.4 Media Queries Should Appear After Non-Media Query Code

Since CSS is applied in the order that it appears, code that targets multiple devices (such as the `device-aspect-ratio` code that targets all Fire tablets) should appear after any non-media query code.

In the following example, the default code creates a black border on every device for any element that uses the `blue_background` class. The media queries that follow it remove the border and display a blue background on Fire tablets and iPad for any element that uses the `blue_background` class. All other devices will only display the black border.

Example:

```
/* Default Formatting. Use this for Kindle e-Ink. No media queries required. */

.blue_background {

  border: 1px solid black;

}

/* Kindle Fire (All) Formatting. Use this to target all Kindle Fires. */

@media amzn-kf8 and (device-aspect-ratio:1280/800) {

  .blue_background {

    background-color: blue;
```

```
    border: none;
  }
}
/* iPad (3, Air, Mini) Formatting. Use this for any iPad. */
@media (device-width: 768px) {
  .blue_background {
    background-color: blue;
    border: none;
  }
}
```

In the example above, the black border defined in the default code for the `blue_background` class will still appear on Fire tablets if the `border` property is not overridden. Setting the `border` to `none` in the media queries for Fire tablets and iPad ensures that the default values for these properties are overridden. This is helpful if you are using non-media query code to target Kindle e-Ink devices and do not want to transfer a colored border to a Fire tablet.

17.1.5 Avoid Duplicating Code

When writing media queries, only include the CSS classes and code that you need to change for that particular device. Any non-media query code that you use will automatically appear on all devices unless it is overwritten with a media query, so you do not need to repeat code that you want to apply to all devices.

In the following example, the goal is to to override a colored border on Fire tablets and replace it with a colored background, while keeping the text red for all devices. The example on the left is incorrect because repeating the .red_font class in the media query code is not necessary.

Example:

Incorrect Media Query	Correct Media Query

`/* Default formatting. */`
 	
`.blue_background {`
`  border: 1px solid blue;`
`}`
 	
`.red_font {`
`  color: red;`
`}`
 	
`/* Kindle Fire (All) Formatting */`
 	
`@media amzn-kf8 and (device-aspect-ratio:1280/800) {`
 	
`  .blue_background {`
`  background-color: blue;`
`  border: none;`
`  }`
 	
`  .red_font {`
`  color: red;`
`  }`
`}`
	`/* Default formatting. */`
	
	`.blue_background {`
	`  border: 1px solid blue;`
	`}`
	
	`.red_font {`
	`  color: red;`
	`}`
	
	`/* Kindle Fire (All) Formatting */`
	
	`@media amzn-kf8 and (device-aspect-ratio:1280/800) {`
	
	`  .blue_background {`
	`  background-color: blue;`
	`  border: none;`
	`  }`
	`}`

17.2 Using Media Queries

The following table outlines examples of supported media queries and the CSS applied to KF8, Mobi, and other readers:

Media Queries in CSS	CSS Applied to KF8	CSS Applied to Mobi	CSS Applied to Other Readers

`@media amzn-mobi`
`{`
`.class1`
`{`
`font=size:3em;`
`font-weight: bold;`
`}`
`}`
	-	`font-size:3em;`
		`font-weight: bold;`
			-

Media Queries in CSS	CSS Applied to KF8	CSS Applied to Mobi	CSS Applied to Other Readers

.class1
{
  font-style: italic;
  font-size:2em;
}
 	
@media amzn-mobi
{
.class1
{
  font-size:3em;
  font-weight: bold;
}
}
	font-style: italic;
	font-size: 2em;
		font-style: italic;
		font-size: 3em;
		font-weight: bold;
			font-style: italic;
			font-size:2em;

@media amzn-mobi
{
.class1
{
  font-size:3em !important;
  font-weight: bold !important;
}
}
 	
.class1
{
  font-style: italic;
  font-size:2em;
}
	font-style: italic;
	font-size:2em;
		font-style: italic;
		font-size:3em;
		font-weight: bold;
			font-style: italic;
			font-size:2em;

Media Queries in CSS	CSS Applied to KF8	CSS Applied to Mobi	CSS Applied to Other Readers

`@media not amzn-mobi`
`{`
`.firstletter`
`{`
`  float: left;`
`  font-size: 3em;`
`  line-height: 1;`
`  font-weight: bold;`
`  padding-right: .2em;`
`  margin: 10px`
`}`
`}`
 	
`@media amzn-mobi`
`{`
`.firstletter {`
`  font-size: 3em;`
`}`
`}`
	`.firstletter`
	`{`
	`  float: left;`
	`  font-size: 3em;`
	`  line-height: 1;`
	`  font-weight: bold;`
	`  padding-right: .2em;`
	`  margin: 10px`
	`}`
		`.firstletter`
		`{`
		`font-size: 3em;`
		`}`
			`.firstletter`
			`{`
			`  float: left;`
			`  font-size: 3em;`
			`  line-height: 1;`
			`  font-weight: bold;`
			`  padding-right: .2em;`
			`  margin: 10px`
			`}`
			`}`

`@media amzn-kf8`
`{`
`  p {`
`  color: red;`
`  }`
`}`
	`p {`
	`  color: red;`
	`}`		

17.3 Using Media Queries for Backward Compatibility With Mobi

Media queries allow one CSS file to supply complex CSS for KF8 and basic CSS for the Mobi format. Some guidelines:

- Complex CSS can be overridden for the Mobi format by redefining the same class inside the `@media amzn-mobi` media query.
- Per the W3C standard, media queries should either be:
    - Individual queries specified after the common CSS; or

      Example:

      ```
      class1 {font-size: 2em;}

      @media amzn-mobi {.class1 {font-size: 3em;}}
      ```

    - Include `!important` with each property to enforce precedence.

      Example:

      ```
      @media amzn-mobi {.class1 {font-size: 3em !important;}}
      ```

```
.class1 {font-size: 2em;}
```

CSS	CSS Styles Applied to Mobi	CSS Styles Applied to KF8

p {
  font-style: normal;
}
 	
h {
  font-weight: bold;
}
 	
div.example {
  margin: 10px
}
 	
ul {
  margin: 20px;
  padding-left: 30px;
}
 	
.firstletter {
  float: left;
  font-size: 3em;
  line-height: 1;
  font-weight: bold;
  padding-right: .2em;
}
 	
@media amzn-mobi {
  .firstletter {
  float: 0;
  font-size: 3em;
  line-height: 0;
  font-weight: bold;
  padding-right: 0;
  }
}
	p {
	  font-style: normal;
	}
	
	h {
	  font-weight: bold;
	}
	
	div.example {
	  margin: 10px
	}
	
	ul {
	  margin: 20px;
	  padding-left: 30px;
	}
	
	.firstletter {
	  float: 0;
	  font-size: 3em;
	  line-height: 0;
	  font-weight: bold;
	  padding-right: 0;
	)
		p {
		  font-style: normal;
		}
		
		h {
		  font-weight: bold;
		}
		
		div.example {
		  margin: 10px
		}
		
		ul {
		  margin: 20px;
		  padding-left: 30px;
		}
		
		.firstletter {
		  float: left;
		  font-size: 3em;
		  line-height: 1;
		  font-weight: bold;
		  padding-right: .2em;
		}

17.3.1 Submitting a Media Query

There are four options for submitting media queries:

- One CSS file;
- Different CSS files;
- Style tags; and
- @import.

17.3.1.1 Option 1: Using One CSS File.

Media queries can specify different CSS for Mobi and KF8 formats in the same CSS file. In the example below, a different **.class1** class is specified for the Mobi format than for the other formats in the same CSS file.

Example:

```
.class1 {
   font-style: italic;
   font-size:2em;
}
@media amzn-mobi {
  .class1 {
    font-size:3em;
    font-weight: bold;
  }
}
```

17.3.1.2 Option 2: Using Different CSS Files

Media queries can specify different CSS for Mobi and KF8 formats in different CSS files. In the example below, the Mobi and KF8 formats utilize different CSS style sheets and the common CSS styles apply to all media.

Example:

```
<link href="common.css" rel="stylesheet" type="text/css">
<link href="kf8.css" media="amzn-kf8" rel="stylesheet" type="text/css">
<link href="mobi.css" media="amzn-mobi" rel="stylesheet" type="text/css">
```

17.3.1.3 Option 3: Using Style tags

Media queries can specify different CSS for Mobi and KF8 formats directly using `<style>` tags.

Example:

```
<style type="text/css">
<style type="text/css" media="amzn-kf8">
<style type="text/css" media="amzn-mobi">
```

17.3.1.4 Option 4: Using @import

Media queries can specify different CSS for Mobi and KF8 formats directly using `@import` to include different CSS files.

Example:

```
@import

@import url(common.css);

@import url(kf8.css) amzn-kf8;

@import url(mobi7.css) amzn-mobi;
```

17.3.2 Using the display:none Property with Media Queries

To specify different CSS for the content in Mobi 7 and KF8 format, use the `display:none` property with media queries. Support for the `display:none` property in the Mobi 7 format is available in Kindlegen 2.4 and later versions.

Example:

```
.defaultcontent {

  display: block;

}

.mobicontent {

  display: none;

}

@media amzn-mobi {

  .defaultcontent {

    display: none;

  }

  .mobicontent {

    display: block;

  }

}
```

17.3.2.1 Using the display:none Property with Complex Tables

Tables have extensive support in KF8, but complex tables do not render well in Mobi 7. With the `display:none` property, you can use an HTML-based table for the KF8 content and an image-based table for Mobi 7, as shown in the example below.

Example:

```
.defaultcontent {

  display: block;
```

```
}
.mobicontent {
  display: none;
}
@media amzn-mobi {
  .defaultcontent {
    display: none;
  }
  .mobicontent {
    display: block;
  }
}
<table class="defaultcontent" bordercolor="#E66C2C" border="1" cellpadding="4">
  <tr>
    <th>Heading</th>
    <th>Heading</th>
    <th>Heading</th>
  </tr>
  <tr>
    <td>Cell</td>
    <td>
      <table bordercolor="#003399" border="1" cellpadding="4">
        <tr>
          <td>Nested</td>
          <td>Nested</td>
        </tr>
        <tr>
          <td>Nested</td>
          <td>Nested</td>
        </tr>
```

```
      </table>
    </td>
    <td>Cell</td>
  </tr>
</table>
<img class="mobicontent" src="tableimage.jpg"></img>
```

17.3.2.2 Using the display:none Property with SVG Images

The SVG image format is supported in KF8, but not in Mobi 7. With the `display:none` property, you can use an SVG image for the KF8 content and a JPEG image for the Mobi 7 content, as shown in the example below.

Example:

```
.defaultcontent {
  display: block;
}
.mobicontent {
  display: none;
}
@media amzn-mobi {
  .defaultcontent {
    display: none;
  }
  .mobicontent {
    display: block;
  }
}
<svg class="defaultcontent" xmlns="http://www.w3.org/2000/svg" version="1.1">
  <circle cx="100" cy="50" r="40" stroke="black" fill="red" />
</svg>
<img class="mobicontent" src="circleimage.jpg"></img>
```

17.3.2.3 Limitation on Using the display:none Property

Kindle limits usage of the `display:none` property for content blocks beyond 10000 characters. If the `display:none` property is applied to a content block that is bigger than 10000 characters, Kindlegen returns an error.

18 Appendix D: Guidelines for Converting XMDF to KF8

18.1 Kindlegen Command

Kindlegen accepts the folder containing the XMDF source file as input. Use the command:

```
kindlegen <folder name>
```

You can use an additional command line option (`-intermediate_only`) to generate intermediate OPF/HTML files:

```
kindlegen -intermediate_only <folder name>
```

This option generates intermediate files as OPF/HTML files in a new folder next to the input folder. Kindlegen names this folder with the input folder name followed by **_dump**. To change anything in the intermediate files, use these intermediate HTML or OPF files.

The OPF file is named **current_content.opf** and can be found at the same location as the **main.xml** file.

Use the following command to generate the KF8 from OPF/HTML files:

```
kindlegen <OPF File>
```

18.2 Source Issues

An XMDF source can have multiple issues that degrade the reading experience on Kindle. You can correct these issues to improve the reading experience. The following sections list a few of the top source issues found in our testing,

18.2.1 Poor Image Quality

Bad quality images will result in the cover or other image being displayed too small. To avoid this, provide good quality images. For quality requirements, see section 4 Cover Image Guidelines and section 9.4 Image Guidelines.

18.2.2 Blurred Gaiji characters

Replace Gaiji characters with corresponding code characters or provide high quality images of at least 64 x 64 to avoid blurring.

Requirements for Gaiji image files:

- Image format: PNG (8-bit) format or JPEG format (8-bit transparent image recommended)
- Size: 128 x 128 pixels or greater is recommended

18.2.3 TOC Entries Hanging Style

Use the proper style of positive margin and negative indent to create TOC text with the proper indentation.

Correct:

```
<p top_line_indent = "-1em" top = "1em">

<char_id char_id = "CR0002">Part 1</char_id><br/>
```

```
<char_id char_id = "CR0004">Sub-Part 1</char_id><br/>

<char_id char_id = "CR0006">Sub-Part 2</char_id><br/>

</p>
```

Incorrect:

```
<char_id char_id = "CR0002">Part 1</char_id><br/>

<char_id char_id = "CR0004">Sub-Part 1</char_id><br/>

<char_id char_id = "CR0006">Sub-Part 2</char_id><br/>
```

18.2.4 TOC Entries Not Linked

All TOC items must be linked to the corresponding chapter.

18.2.5 Incorrect Number Orientation in TOC

Apply tate-chu-yoko style for TOC index numbers to avoid incorrect number orientation.

Correct:

```
<char_id char_id = "CR0020">

<yoko>10</yoko>

</char_id>

CHAPTER X
```

Incorrect:

```
<char_id char_id = "CR0020">10</char_id>

CHAPTER X
```

18.2.6 Bold Kanji Characters

Avoid bold style for Kanji characters. This sort of styling makes the characters look blurred.

Correct:

遺

Incorrect:

```
<font bold = "yes">遺</font>
```

18.2.7 Faded Text

Leave the text color unspecified. Using light colors results in faded text.

Some Kindle devices allow the reader to change the background color to black. If text is forced black, it will not be readable in this mode and the book will be suppressed.

18.2.8 No Space Between Images

Use line breaks (<br/>) to avoid images rendered without space.

Example:

```
<object type="image/png" src="image1.png"/>

<br/>

<object type="image/png" src="image2.png"/>
```

18.2.9 Images Not Shown In Separate Pages

To display images on separate pages, use separate chapters for each image.

18.2.10 TOC Not Shown

To ensure that the TOC is shown, include all of the important links in <special_page_link>.

Example:

```
<special_page_link>

<special_page title="Chapter 1">PG1111</special_page>

<special_page title="Chapter 2">PG1112</special_page>

...

</special_page_link>
```

18.2.11 Large Amount of Text Is Not Centered

Do not use the valign="middle" style for large amounts of text.

18.2.12 Duplicate IDs

Do not use the same Flow ID/Page ID mentioned in the source. Use unique ID names.

18.2.13 File Path and File Name Requirements

Do not use the backward slash (\) for specifying file path; always use the forward slash (/) instead. Do not include special characters (such as !, @, #, $) in the file path or file name.

18.3 Unsupported Features

Some XMDF features are not supported in Kindle. If the file has such features, Kindlegen displays an error message and fails the conversion. These are the unsupported features.

Serial No.	Feature	Tag	Example
1	Image animations	flip_animation	<flip_animation renewal_time="500ms" > <flip_animation_source src="aaa9.jpg" type="image/jpeg"/> <flip_animation_source src="aaa2.jpg" type="image/jpeg"/> ... </flip_animation>

Serial No.	Feature	Tag	Example
2	Comic books	comic_object_entry	<parts_module> <object_table> ... <comic_object_entry src="comic9.xml" type="application/x-bvf-comic" object_id="OB0001"/> </object_table> </parts_module>
3	Sound media	sound_object_entry	<parts_module> <object_table> ... <sound_object_entry src="movie9.3g2" type="video/3gpp2" object_id="OBmv00"/> </object_table> </parts_module>
4	Movie media	movie_object_entry	<parts_module> <object_table> ... <movie_object_entry src="movie9.3g2" type="video/3gpp2" object_id="OBmv00"/> </object_table> </parts_module>
5	Search pages	search_page_object_entry	<parts_module> <object_table> ... <search_page_object_entry src="spage9.xml" type="text/x-bvf-search-page" object_id="OBSP01" /> </object_table> </parts_module>
6	Clickable area in images	pointer_region	<event> <trigger> <trigger_pointer id="OB003k/CR0001" action_flag="click"> <pointer_region> <vertex position="(0,0)"/> <vertex position="(100,0)"/> <vertex position="(100,100)"/> <vertex position="(0,100)"/> </pointer_region> </trigger_pointer> </trigger> <action> <action_page_jump page_id="PG0043"/> </action> </event>
7	Trigger action pointing to a URL with ampersand	action_page_jump	<event> <trigger> <trigger_pointer id="OB0006/CR0015" action_flag="click"/>

Serial No.	Feature	Tag	Example
	HTML entity		</trigger> <action> <action_page_jump book="http://www.japanvisitor.com/index.php?cID=359&pID=986" book_type="text/html"/> </action> </event>

18.3.1 Ignored Features

For some features that are not supported, Kindlegen does not error out. Instead, Kindlegen ignores them and proceeds with the conversion. These are the ignored features.

Serial No.	Feature	Tag/Attribute	Example
1	Pronunciation	Reading attribute	<title reading="PI">π</title>
2	Alternative code and set for Gaiji	Attributes alt_set and alt_code of tag external_char	<external_char alt_set="sh_extchars" alt_code="0x2345" alt="間"/>
3	Permission information	permission_info	<permission_info> <print_permission permission="authorized"/> </permission_info>
4	Line breaking method	line_breaking_method	<line_breaking_method method="word_wrap">
5	Play back	action_play	<event> <trigger> <trigger_pointer id="OB0006/CR0015" action_flag="click"/> </trigger> <action> <action_play object_id="OBkj23"/> ... </action> </event>
6	Opacity for font, background and others	Opacity attribute	<font color="#FF0000" opacity="100"/>
7	Background music	text_default_background_music	<text_default_attribute> <text_default_background_music src="9.mp3" type="application/x-smaf" loop="yes"/> ... </text_default_attribute>
8	Drop cap	drop_cap	Alice was...

Serial No.	Feature	Tag/Attribute	Example
9	Scrolling text content	scrolling_text	<scrolling_text> This text will be scrolling over and over </scrolling_text>
10	Content masking	Mask	<mask>42 </mask>
11	Key entries for search	key_entry	<key_entry> <key_item search_word="color" table_id="ST0001"> Color </key_item> <key_item search_word="colour" table_id="ST0002"> Colour </key_item> </key_entry>