amazonkindle

Amazon Kindle Publishing Guidelines

How to make books available for the Kindle platform

version 2012.5

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.

Table of Contents

1 Getting Started

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 the 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 Adobe InDesign content, 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 Kindle Plugin for Adobe InDesign

Publishers can create Kindle books in-house from Adobe InDesign content by using a free software program called *Kindle Plugin for Adobe InDesign.* This plug-in allows the publisher to convert content from InDesign into Kindle format. The current version of the *Kindle Plugin for Adobe InDesign (v0.96*) supports Kindle Format 8*.* Details of the options currently supported in *Kindle Plugin for Adobe InDesign* are available at www.amazon.com/kindleformat.

The plug-in is available from www.amazon.com/kindleformat. Installation and help documentation for *Kindle Plugin for Adobe InDesign* are available at www.amazon.com/kindleformat.

2.2.2 KindleGen

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

The most recent version of KindleGen can be downloaded for free from www.amazon.com/kindleformat/kindlegen. To create books for Kindle Format 8, use KindleGen 2 or later versions.

Amazon periodically releases new versions of the KindleGen software. Visit www.amazon.com/kindleformat/kindlegen to check for updates.

2.2.2.1 Source Files to Use with KindleGen

To create Amazon 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.

2.2.2.2 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**, go to the directory where the book is located, such as **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**, go to the directory where the book is located, such as **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.

2.2.2.3 Using KindleGen

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

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

Options:

   -c0: no compression

   -c1: standard DOC compression

   -c2: Kindle huffdic compression

   -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
```

2.2.2.4 KindleGen Messages

If KindleGen encounters issues while converting a file, it displays a warning or error.

- Errors impair the readability of the book in the Kindle Reader. It is critical to address errors before the book can be converted and published.
- Warnings cause loss of non-essential functionality when the file is converted. KindleGen will introduce a work-around that will not impair the reading experience.

As conversion progresses, KindleGen displays detailed informational messages.

2.2.3 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.

2.2.3.1 Installing Kindle Previewer

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

Kindle Previewer for Windows (XP, Vista, 7)

1. Download **Kindle Previewer for Windows** from www.amazon.com/kindleformat.
2. Store the executable (**KindlePreviewer.exe**) to the local disk.
3. Execute **KindlePreviewer.exe** by double-clicking it.
4. Accept the EULA from the dialog box with details to install Kindle Previewer.
5. Kindle Previewer appears in **Start > Programs > Amazon > Kindle Previewer** after successful installation.

Kindle Previewer for Mac OS 10.5 and above i386

1. Download **Kindle Previewer for Intel Mac** from www.amazon.com/kindleformat.
2. Save the zip file (**KindlePreviewer.zip**) to the local disk.
3. Double-click the zip file to unzip **Kindle Previewer**.
4. Drag **Kindle Previewer** from **Downloads** folder to **Application** folder.
5. Start **Kindle Previewer**.

Use F1 or the Help menu to find the Kindle Previewer User's Guide.

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 General Formatting Guidelines

Kindle Format 8 (KF8) is the next generation file format (replacing Mobi 7) for Kindle books and supports 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 character 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
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

There are important differences between writing HTML for websites and for Kindle books. To provide a good reading experience, many website design practices should be avoided when creating Kindle books. Refer to the following sections for more information.

Support for Chinese, Japanese, and Korean Text

The Kindle family of devices has limited support for Chinese, Japanese, and Korean text. The Kindle Readers for e Ink devices and applications can render horizontal left to right Chinese, Japanese, and Korean text. They have limited font support for the CJK characters. There is no support for right to left vertical scripts and no support for Japanese Ruby script.

3.1 Text Guidelines

3.1.1 Text Guideline #1: Body Text Must Use Defaults

The body text in a reflowable Kindle book 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 not have a forced alignment (such as left aligned or justified).
- Body text must use the default font size. Body text should not use the `<font size="…">` tag or its equivalent in CSS.
- Body text should not be bold or italicized. Selected parts of the text can be bold or italicized. This guideline only prohibits a book that would be entirely bold, for example.
- Body text should not have an imposed font color.
- Body text must not have a white font color. Customers report this as a bad user experience.
- Body text must not have a black background color. Customers report this as a bad user experience.

3.1.2 Text Guideline #2: Use CSS for Page Breaks

Do not insert blank lines of text to create empty pages. Use the CSS `page-break-before` and `page-break-after` attributes.

3.1.3 Text Guideline #3: 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:10%">` - positive indent, 10% of the width of the page
- `<p style="text-indent:5em">` - positive indent, 5 em

To change the space before each paragraph, use the `margin-top` style on the `<p>` tag.

3.1.4 Text Guideline #4: 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 files in the HTML itself by using the `<meta>` tag in the `<head>` section.

Example

```
<html>

<head>

…

<meta http-equiv="content-type" content="text/html; charset=iso-

       8859-1">

…
```

3.1.5 Text Guideline #5: Spaces and Unicode Characters

The only supported spaces are the normal space, the non-breakable 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.

3.1.6 Text Guideline #6: 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 align="left">` block.

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. Do not include the Charis font; it is replaced with a higher quality font in the Kindle Readers.

3.1.7 Text Guideline #7: CSS Support

The earlier Kindle platform offered very basic support for Cascading Style Sheets (CSS). This has been significantly enhanced in KF8 with support for CSS 2/CSS 3. (See section 11.2 for the list of supported CSS attributes/selectors). To verify that your use of CSS elements displays the way you intended, preview your Kindle book on different devices before publishing it.

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 percentages.

When using the `margin` and `padding` CSS properties, specify the values in percentage (%) instead of em units. This ensures that the margins do not grow wide with large font sizes.

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

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. (**Example:** drop caps: Use font-size: 300%). 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-family: "Times New Roman";
    font-size: small;
    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>here is a sample
```

To verify that the drop caps display as intended, test the book as described in section 9.1, Testing Kindle Books.

3.1.8 Text Guideline #8: Page Numbers

Kindle books do not always map directly to page numbers in physical editions of the book. For this reason, there should not be any reference to page numbers in the book. Page numbers should not be included in cross-references or the index. Amazon may make page numbers available for books as additional book metadata. Amazon generates these page numbers based on its own internal technology.

3.2 Cover Image Guidelines

3.2.1 Cover Image Guideline #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 cover is a JPEG image of 2500 pixels on the longest side (with a minimum of 1000 pixels on the longest side). Covers with less than 500 pixels on the smaller side are uploaded, but are not displayed on the website. No error message is given at time of upload if the image size is too small. If the cover is smaller than the required size, do not stretch it, because this does not add any quality.

The content of the cover image should not:

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

3.2.2 Cover Image Guideline #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 cover, because Amazon quality assurance will fail the book if the cover is too small. Define covers in the OPF file using the following tags:

```
<metadata>
...
```

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

...

</metadata>
  ...
<manifest>

...

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

...

</manifest>
```

The use of `name="cover"` in the metadata element name is mandatory.

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

3.2.3 Cover Image Guideline #3: Internal Cover Must Not Appear Twice

Do not add cover images to the content in any way other than those described in section 3.2.2, Cover Image Guideline #2: Internal Content Cover Image Is Mandatory, or the cover might appear twice in the book.

There is one exception: if you want an HTML cover page for compatibility with software from other vendors, in addition to the proper logical cover, add all of the following tags in the OPF file (underlined elements are mandatory):

```
<spine> <itemref idref="my-html-cover" linear="no" /> </spine>
  ...
<manifest> <item id="my-html-cover" href="cover.xml" media-
type="application/xhtml+xml" /> </manifest>
  ...
<guide> <reference type="cover" title="Cover Image" href="cover.xml" /> </guide>
```

3.3 Table of Contents Guidelines

Amazon strongly recommends the use of an HTML TOC for all books that would benefit from this navigation feature. This applies to most books, with the exception of fixed-layout children's books (see section 4) and fixed-layout graphic novels/manga/comics (see section 5).

3.3.1 TOC Guideline #1: Logical TOC (NCX) Is Recommended

Amazon strongly recommends that all Kindle books include both logical and HTML TOCs. 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.

Logical TOCs are generated using a navigational control file for XML application (NCX). Creating an NCX exposes the hierarchical structure of a Kindle book and allows the user to navigate through it.

In NCX-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.

Logical TOCs are part of the IDPF 2.0 specification and are 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 use it in the `<spine>`:

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

3.3.2 TOC Guideline #2: HTML TOC Must Be Linked

Place an HTML page with a table of contents at the beginning of the book, so that users can easily jump to locations within it (typically to a chapter). 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.

3.3.3 TOC Guideline #3: HTML TOC Must Be Referenced as a Guide Item

To enable the customer to jump to the TOC from the Kindle menu, the OPF file must reference the TOC from a TOC guide item.

Every Kindle device or application has a user interface element that allows the user to jump to the TOC guide item from anywhere in the book. 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>
```

3.3.4 TOC Guideline #4: No Tables in TOC

Do not create a TOC using HTML `<table>` tags. When the TOC includes HTML `<table>` tags, the links of the TOC become not clickable/ non-functional. Tables are for tabular data only, not for layout.

3.3.5 TOC Guideline #5: No Page Numbers in TOC

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.

3.3.6 TOC Guideline #6: Place the TOC at the Front of the Book

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. Inaccurate placement of the TOC affects the accuracy of the “Last Page Read” feature. Correct usage ensures that the TOC appears in the book’s sample.

3.3.7 TOC Guideline #7: Include a TOC for Bundled Editions

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

3.4 Guide Item Guidelines

3.4.1 Guide Item Guideline #1: Recommended Guide Items

The Kindle platform supports guide items for defining the cover, the table of contents (TOC), and the start reading location (”Go to Beginning”). Do not set the start reading location to a blank page.

Amazon does not recommend adding additional guide items to the OPF file, because they will be grayed out in the menu options and may cause customer confusion.

IMPORTANT: Guide items, especially the TOC guide item, do not replace the table of contents.

3.5 Image Guidelines

These guidelines apply to most books, but are not applicable to the image-intensive fixed-layout children's books (see section 4) and fixed-layout graphic novels/manga/comics (see section 5).

3.5.1 Image Guideline #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.

To future-proof the content, save images in 300 dpi or 300 ppi resolution.

3.5.2 Image Guideline #2: KindleGen Performs Automatic Image Conversions

The limitation on the image size depends upon the book’s format:

- Reflowable books (fiction and non-fiction): 127 KB maximum

- Fixed-layout books (other than comics): 256 KB maximum
- Comic books: 800 KB maximum

KindleGen performs the necessary conversions automatically from the supported input formats, so you should provide images with the maximum resolution available and let KindleGen do the rest.

If the automatic image conversion is unsatisfactory, try optimizing the images before feeding them to KindleGen. If the images are in the JPEG or GIF formats and are less than 127 KB in size, KindleGen does not alter them. For fixed-layout books and comic books, KindleGen reduces the quality factor of the image, but not the resolution. If the image cannot be scaled down to meet the size limits by reducing quality factor up to 40%, KindleGen fails the conversion process.

3.5.3 Image Guideline #3: Use Color Images

Use color images whenever possible and relevant. The Kindle e Ink devices currently have a black and white screen, but color is available on the Kindle Fire, Kindle for iPhone, and Kindle for PC.

3.5.4 Image Guideline #4: Photographs Should Use JPEG Format

Photographs should use the JPEG format with a quality factor of 40 or higher. Photographs should use the highest resolution available. KindleGen reprocesses the photographs as needed to adapt them to the requirements of the file format.

Photographs should not be too small. Make sure that input photos are at least 600 x 800 pixels in size, unless you optimize them according to section 3.5.2, Image Guideline #2: KindleGen Performs Automatic Image Conversions. Photographs of less than 300 x 400 pixels are much too small and can be rejected.

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.

3.5.5 Image Guideline #5: Use GIF for Line-Art and Text

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

Line-art should be in GIF format. The JPEG algorithm tries to blend parts of the image together, and blurs the sharp edges of the line-art.

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

Optimize line-art GIFs before submitting them to KindleGen. Resizing or JPEG compression can introduce blurriness or unwanted artifacts in line-art images, which is why the automatic conversions applied by KindleGen are best avoided.

To optimize the GIFs and make them fit in the 127 KB 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.
- Resize the image, if necessary, but pay close attention to the legibility of text (see section 3.5.6, Image Guideline #6: Image and Font Size Requirements for Line-Art and Text).

3.5.6 Image Guideline #6: Image and Font Size Requirements for Line-Art and Text

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

iPhone application allow zooming and panning. However, reading experience degrades rapidly for very large line-art images.

The following rules ensure a good rendering on all Kindle platforms for line-art images containing text:

- The MAXIMUM image size is 500 x 600 pixels. This ensures that the image is not shrunk on a Kindle device, which could make the text illegible.
- The MINIMUM size of text is 6 pixels for the height of a lowercase “a.”

These rules limit the size of tables rendered as pictures. Larger tables should be reformatted.

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).	
The same image with JPEG compression. Compression artifacts appear, making the text blurry although it remains legible. The size has increased to 17 KB.	

Description	Image
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	Quality	Country of origin
Apple	4	$ 250,000	25%		$ 250,000	1st choice	U.K.
Peach	2	$ 150,000	13%	Sold well	$ 150,000	1st choice	Holland
Banana	5	$ 670,000	31%		$ 670,000	2nd choice	Germany
Pear	3	$ 560,000	19%		$ 560,000	3rd choice	Mexico
Plum	2	$ 432,000	13%	Need more marketing	$ 432,000	1st choice	Argentina
Walnut	**1**	**$ 35,000**	**3%**		**$ 35,000**	**3rd choice**	**Uruguay**
Pineapple	15	$14,000	2 %		$14,000	2nd choice	Ethiopia
Grapefruit	5	$1,345,000	23 %	Prospective sales	$1,345,000	3rd choice	Iran
Hazelnut	3	$ 25,000	45 %		$ 25,000	2nd choice	Japan
Apple	4	$ 250,000	25%		$ 250,000	1st choice	U.K.
Peach	**2**	**$ 150,000**	**13%**	Different batch	**$ [illegible],000**	**1st choice**	**Holland**
Banana	5	$ 670,000	31%		$ 67[illegible],000	2nd choice	Germany
Pear	3	$ 560,000	19%	[illegible]otiate	$ 56[illegible],000	3rd choice	Mexico
Plum	2	$ 432,000	13%	[illegible] g[illegible]	[illegible],000	1st choice	Argentina
Walnut	1	$ [illegible],000	3%	[illegible]	$ 35,000	3rd choice	Uruguay
Pineapple	15	$ [illegible],000	[illegible]%		$14,000	2nd choice	Ethiopia
Grapefruit	5	$1,345,[illegible]00	[illegible]%	Prospective sales	$1,345,000	3rd choice	Iran
Hazelnut	3	$ 25,00[illegible]	45 %		$ 25,000	2nd choice	Japan
Apple	4	$ 250,000	25%		$ 250,000	1st choice	U.K.
Peach	2	$ 150,000	13%	Sold well	$ 150,000	1st choice	Holland
Banana	**5**	**$ 670,000**	**31%**		**$ 670,000**	**2nd choice**	**Germany**
Pear	3	$ 560,000	19%	Find alternative suppliers	$ 560,000	3rd choice	Mexico
Plum	2	$ 432,000	13%		$ 432,000	1st choice	Argentina
Walnut	1	$ 35,000	3%		$ 35,000	3rd choice	Uruguay
Pineapple	15	$14,000	2 %		$14,000	2nd choice	Ethiopia
Grapefruit	5	$1,345,000	23 %	Prospective sales	$1,345,000	3rd choice	Iran
Hazelnut	3	$ 25,000	45 %		$ 25,000	2nd choice	Japan
Plum	2	$ 432,000	13%		$ 432,000	1st choice	Argentina
Total	**16**	**$3,679,000**	**235 %**		**$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.

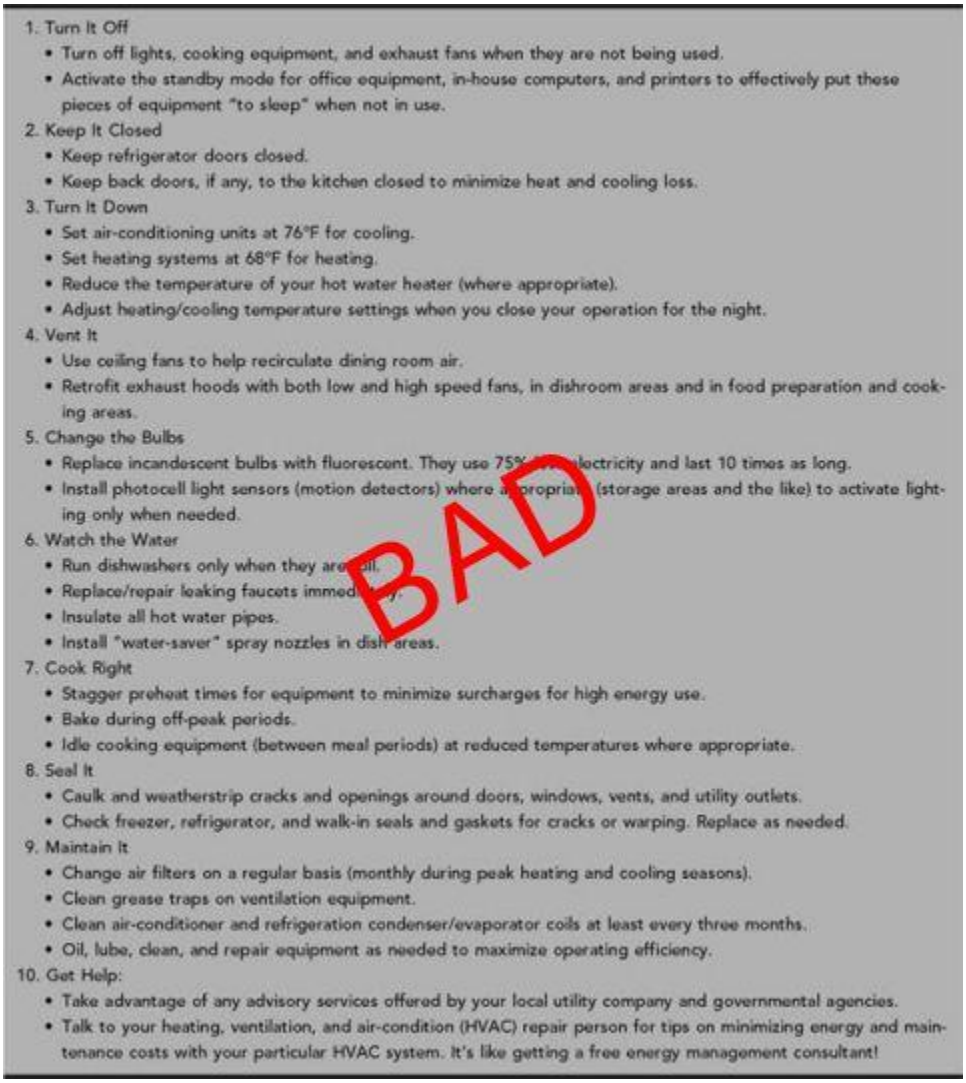
Fruit	#/week	Sales	Remarks	Sales	Quality	Country of origin
Apple	4	$ 250,000	Sold well	$ 250,000	1st choice	U.K.
Peach	2	$ 150,000		$ 150,000	1st choice	Holland
Banana	5	$ 670,000		$ 670,000	2nd choice	Germany
Pear	3	$ 560,000	Need more marketing	$ 560,000	3rd choice	Mexico
Plum	2	$ 432,000		$ 432,000	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**		

3.5.7 Image Guideline #7: 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.



3.5.8 Image Guideline #8: Image Caption Placement

Amazon recommends placing a caption below the related image, so that the reader views the image before the caption.

3.5.9 Image Guideline #9: 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.

3.5.10 Image Guideline #10: 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>
```

3.6 Table Guidelines

3.6.1 Table Guideline #1: Use Tables for Tabular Data Only

IMPORTANT: Use tables for tabular data only. Although it is common practice to use tables for layout in websites, this is not allowed in Kindle books. Do not use tables for dialogue, transcripts, chronologies, tables of contents, lists, sidebars, and so on.

3.6.2 Table Guideline #2: Avoid Large Tables

A table rendered as an image cannot be paginated, and the whole image is drawn 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 is a poor user experience.

For the best user experience, please respect the following rule: 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 enables legible text for the customer to view.

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)

3.6.3 Table Guideline #3: Create Simple HTML Tables

Use the `<table>` tags to create simple tables that have standard rows and columns. These tables display as tables on currently available Kindle devices and Kindle for iPhone. On Kindle 1, these tables are flattened (all of the content appears in one column). 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.

3.6.4 Table Guideline #4: 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.

3.6.5 Table Guideline #5: 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 3.6.4, Table Guideline #4: 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

3.7 Adobe Digital Editions Compatibility Guidelines

3.7.1 Adobe Digital Editions Compatibility Guideline #1: Use Unique Item IDs

When using Adobe Digital Editions, make sure that the item IDs in the manifest are unique. Adobe Digital Editions does not enforce uniqueness of IDs, which is incorrect according to the IDPF standard.

```
<manifest>
   <item id="css1" href="core.css" media-type="text/css"/>
   <item id="css2" href="template.css" media-type="text/css"/>
...
</manifest>
```

3.8 Styling Guidelines

3.8.1 Styling Guideline #1: Use a Nested HTML TOC

To create useful, navigable, deep 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>
```

3.8.2 Styling Guideline #2: Format Sidebars Correctly

When inserting sidebar content into the main flow of a book formatted for Kindle Format 8, use float via CSS. However, if the book is formatted for Mobi 7, use the `<hr/>` HTML tags before and after the sidebar to differentiate it from the main body of text. Avoid using negative em values when specifying dimensions for a float element.

3.9 HTML Guidelines

3.9.1 HTML Guideline #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

3.9.2 HTML Guideline #2: Anchors Must Be Added Before Formatting Tags

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

3.9.3 HTML Guideline #3: EPUB Guide Items Are Optional

Guide items are an optional feature in the EPUB format. Kindle provides support for the cover, TOC, and text guide items.

3.9.4 HTML Guideline #4: Using Single Column Layout and Avoiding Absolute Positions

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

3.9.5 HTML Guideline #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.

3.9.6 HTML Guideline #6: Avoid Using Negative Em Values

Avoid using negative em values for positioning text and margins.

3.9.7 HTML Guideline #7: Avoid Using Scripting with SVG Images

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

3.9.8 HTML Guideline #8: Avoid Using Negative Values for Line Height

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

3.10 Embedded Font Guidelines

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. Publishers do not need to include the Charis font with their Kindle books because it is an Open Font Licensed typeface.

4 Creating Fixed-Layout Children's Books

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 and graphic novels have full-page images with text set precisely in relation to the background art. To accommodate these media types, KF8 introduces new metadata fields and corresponding guidelines.

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.

4.1 Metadata Fields Supporting Fixed-Layout Books

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
`<meta name="fixed-layout" content="true"/>`	Required. Identifies the book as having a fixed layout. Valid values are `true` or `false`. The default value is `false`.
`<meta name="original-resolution" content="1024x600"/>`	Required. Identifies the original display size the content was designed for. The pixel dimensions can have any positive integer value. These values must equal the overall aspect ratio of the original content.
`<meta name="orientation-lock" content="landscape"/>`	Required for children's content; optional for comics. Valid values 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 4.2.2.
`<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`.
`<itemref id="page-id" properties="page-spread-left"/>`	Optional. 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="children"/>`	Optional. Removes reader functionality (e.g., share) which may not be relevant for certain books such as children's. Valid values are `children` or `comic`.

4.2 Content Requirements

4.2.1 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:

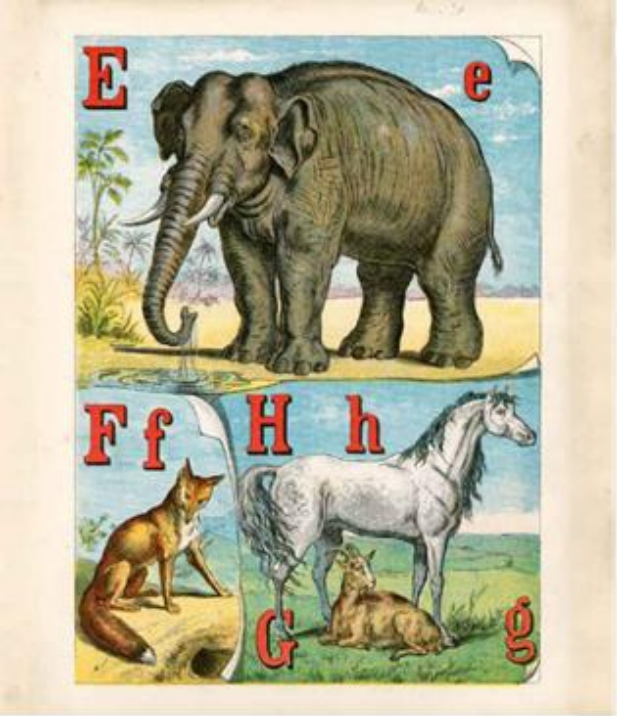
No orientation lock:

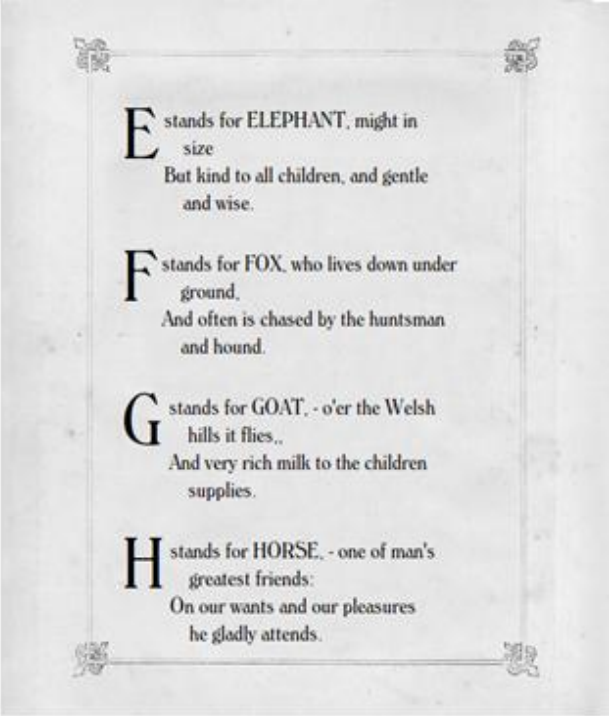
Use 1 print page = 1 HTML file.

The content should be designed for both portrait and landscape mode. In portrait mode, the pages are displayed one after another. In landscape mode, two pages are stitched together with a clear visual separator between them.

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

Example:





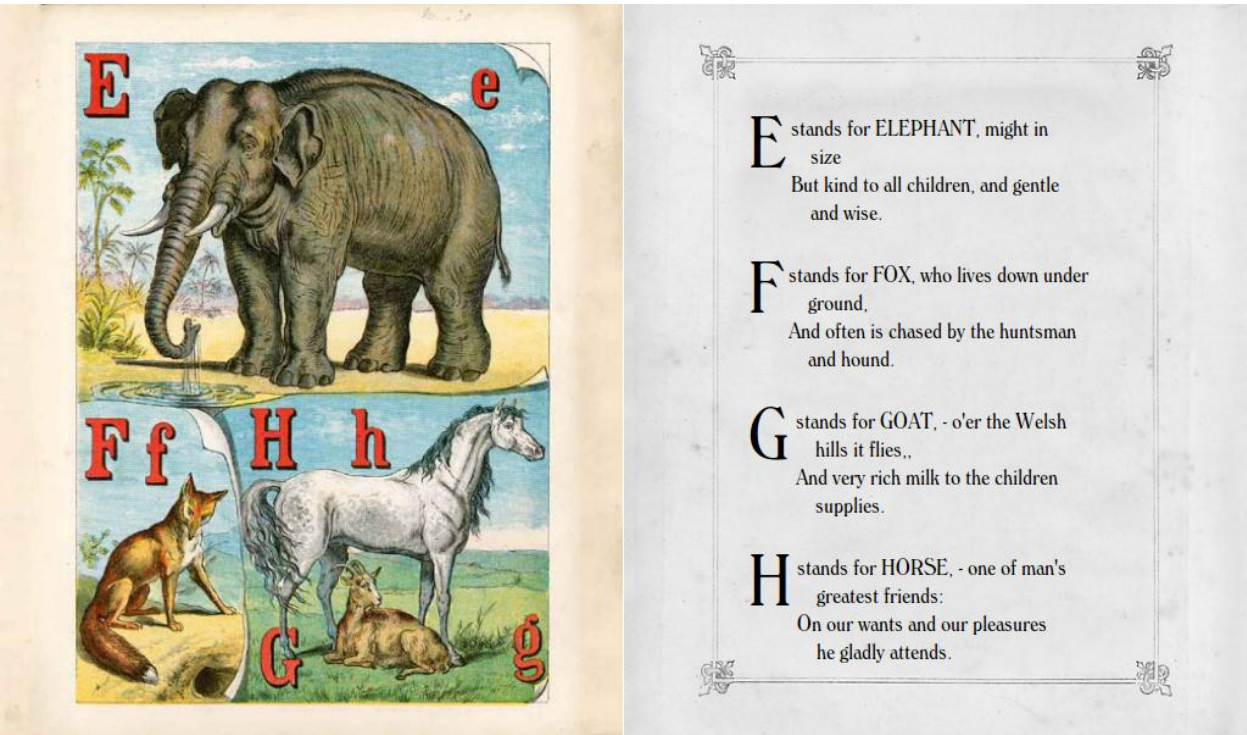
For double-page spreads, the publisher should specify properties `page-spread-left` and `page-spread-right`. In this case, the renderer will not add the spine while stitching two pages together.

Example:

```
<spine>
      <item idref="page1" properties="page-spread-left"/>
      <item idref="page2" properties="page-spread-right"/>
</spine>
```

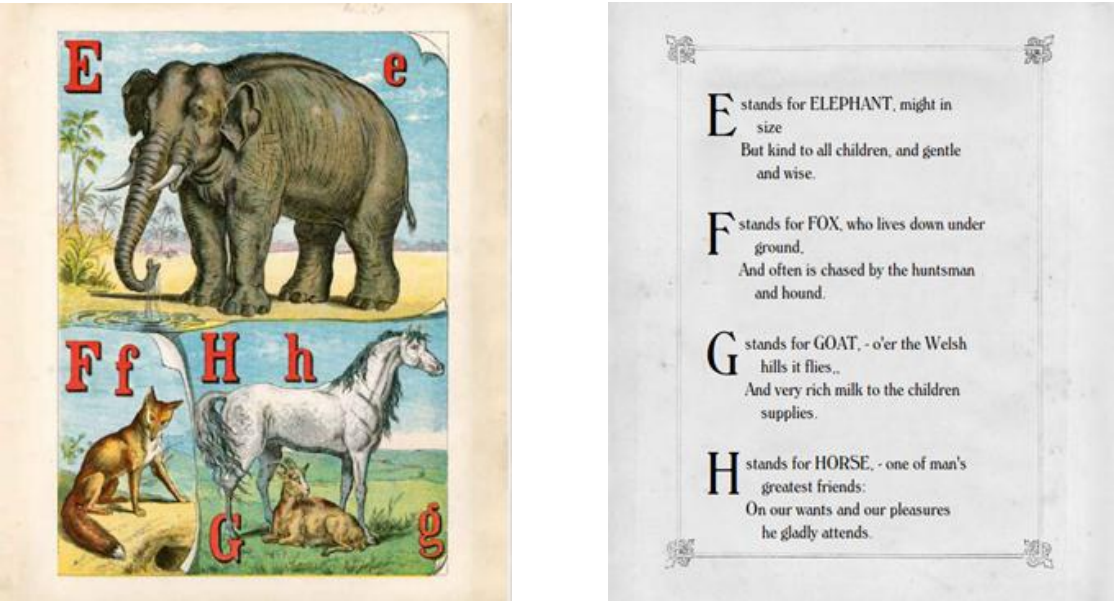
The following example shows a double-page spread without visual separator in landscape mode.

Example:



The following example shows a double-page spread in portrait mode, with the two pages rendered separately.

Example:



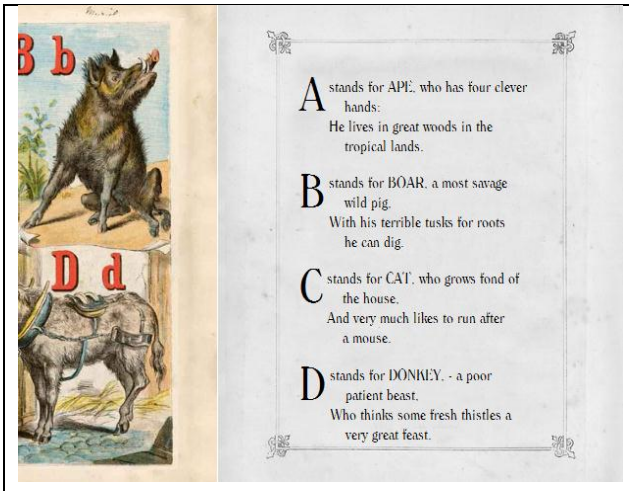
4.2.2 Requirement #2: Using Region Magnification (Pop-ups)

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.

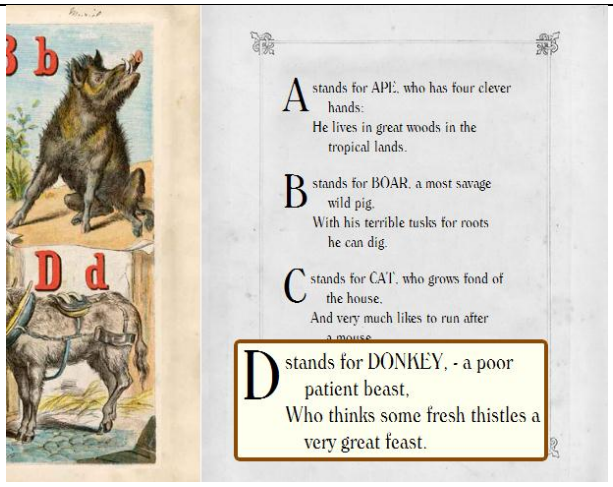
The user activates Region Magnification by double tapping an “active area” on touch screen devices. (On non-touch screen devices, clicking the up arrow on the 5-way controller selects the region and clicking the center button activates Kindle Text Pop-Up or Kindle Panel View.) During Region Magnification, 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 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 text will not disappear from the page view.
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 more readable.



Fixed-layout sample page



Same sample page with Region Magnification activated

Example:

```
<a class="app-amzn-magnify"

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

```
  <div id="fs1-4-txt">

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

4.2.3 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.

4.2.4 Requirement #4: Including HTML Front Cover

Include an HTML front cover page with the cover image set as a background-image. This determines where the book opens in the Kindle. This HTML page must be the first page listed in the `<spine>` in the OPF file, with the `linear` attribute assigned to `yes`. See sections 3.2.2 and 3.2.3 for other guidelines surrounding cover images.

4.2.5 Requirement #5: Pairing Pages When Orientation-Lock Equals None

When `orientation-lock` for content is not specified or equals `none`, 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`. 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.

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>

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

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

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

      <item 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>

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

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

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

      <item 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>

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

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

      <item idref="page2" properties="page-spread-left"/> <!—double page
spread's left viewport -->

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

</spine>
```

4.3 Content Recommendations

4.3.1 Recommendation #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.

4.3.2 Recommendation #2: Including One CSS File Per HTML Page

To increase page-turn performance on fixed-layout books, include one CSS file per HTML page.

4.3.3 Recommendation #3: Optimizing Content for Full Screen

Kindle books are read across a wide variety of devices (e.g., Kindle e Ink, Kindle Fire, and other manufacturers' smartphones and tablets) and a wide variety of screen dimensions. The Kindle Fire has a resolution of 1024 x 600 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. Amazon recommends submitting images scaled to support at least 2X magnification with high quality. For example, if planning for Kindle Fire, the image pixel dimensions should be at least 2048 x 1200 (this matches the aspect ratio and would support 2X zoom). Always use Kindle Previewer to validate the quality of the content.

4.3.4 Recommendation #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.

4.3.5 Recommendation #5: 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).

4.3.6 Recommendation #6: 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(../fonts/arial.otf); /* includes the file for the correct font */
}
```

4.3.7 Recommendation #7: Including HTML Front Cover

Include an HTML front cover page with the cover image set as a background image. This will act as the beginning of the book. This HTML page must be the first page listed in the `<spine>` in the OPF file, with the `linear` attribute assigned to `yes`. See sections 3.2.2 and 3.2.3 for other guidelines surrounding cover images.

4.3.8 Recommendation #8: Including Back Cover

While Kindle ebooks in previous formats and reflowable text 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.

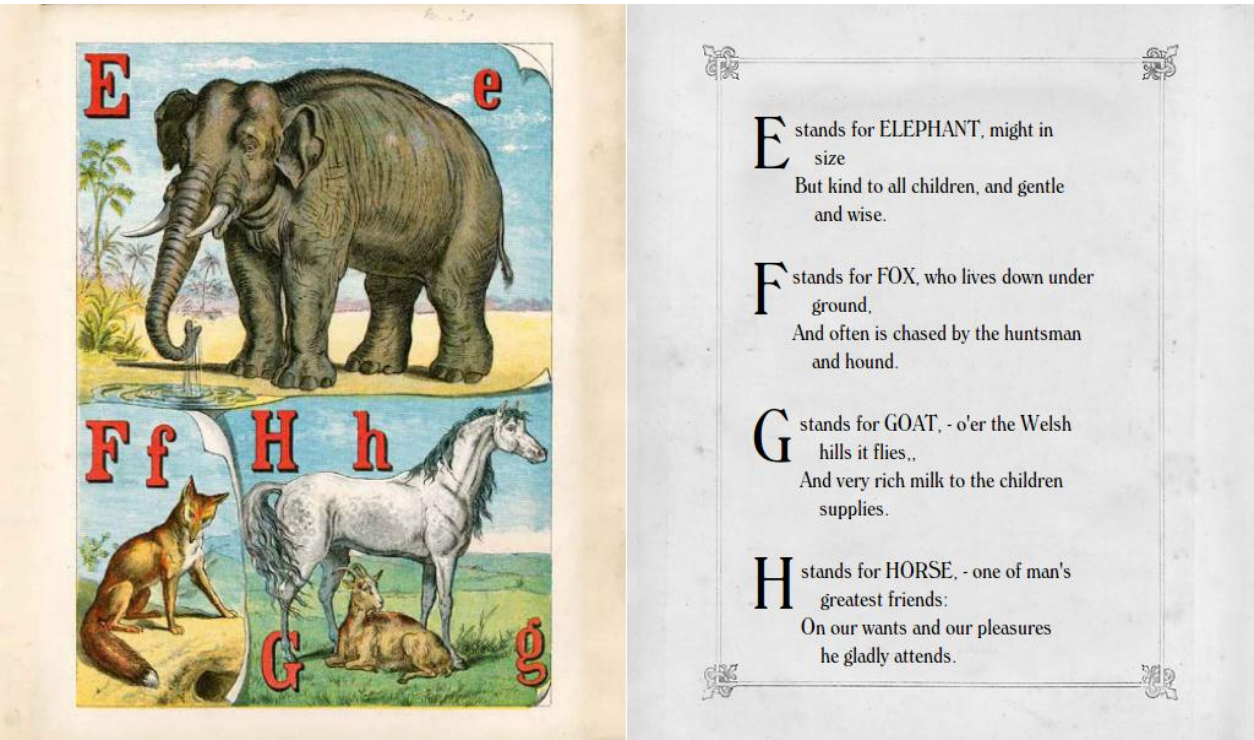
4.4 Creating Children's Books with Multipage Background Images and Text

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.

4.4.1 Using Side-by-Side Images to Form a Double-Page Spread When Orientation-Lock Equals 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. 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.



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 */

}
```

4.4.2 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.

The example in section 4.2.2, 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.

4.4.3 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.

Limit the use of non-breaking space (` `) characters. Instead, emphasize alignment by accurate use of CSS over manual spacing. Use top, right, bottom, and left to position `<div>` elements that contain absolutely positioned text. CSS `text-indent` and `line-height` are useful in aligning text within HTML block elements.

5 Creating Fixed-Layout Graphic Novels/Manga/Comics

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 1024 x 600 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.

When designing for graphic novels, the following generic fixed-layout requirements also apply:

5.1 Metadata Fields Supporting Fixed-Layout Books

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
`<meta name="fixed-layout" content="true"/>`	Required. Identifies the book as having a fixed layout. Valid values are `true` or `false`. The default value is `false`.
`<meta name="original-resolution" content="1024x600"/>`	Required. Identifies the original display size the content was designed for. The pixel dimensions can have any positive integer value. These values must equal the overall aspect ratio of the original content.
`<meta name="orientation-lock" content="landscape"/>`	Required for children's content; optional for comics. Valid values are `portrait`, `landscape`

Metadata	Description
	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 4.2.2.
`<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`.
`<itemref id="page-id" properties="page-spread-left"/>`	Optional. 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="children"/>`	Optional. Removes reader functionality (e.g., share) which may not be relevant for certain books such as children's. Valid values are `children` or `comic`.

5.2 Asset Requirements

When optimized for the Kindle Fire, graphic novels should maintain a 1024 x 600 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 and must be smaller than 800 KB in size. Image files larger than 800 KB greatly increase download time for the book and require more space on the device.

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.	1024 x 600 pixels
125%	Only use this zoom factor when it is absolutely necessary to enlarge a very	1280 x 750

Zoom Factor	When to Use	Required Image Resolution
	large panel. This allows the user to see a large action scene, but with the downside of limited enlargement.	pixels
150%	This is the default and preferred zoom factor. Use this zoom factor whenever possible.	1536 x 900 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.	1560 x 1500 pixels

5.3 Image Quality

Image quality for graphic novels requires that images follow the resolution standards listed in section 5.2, 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.

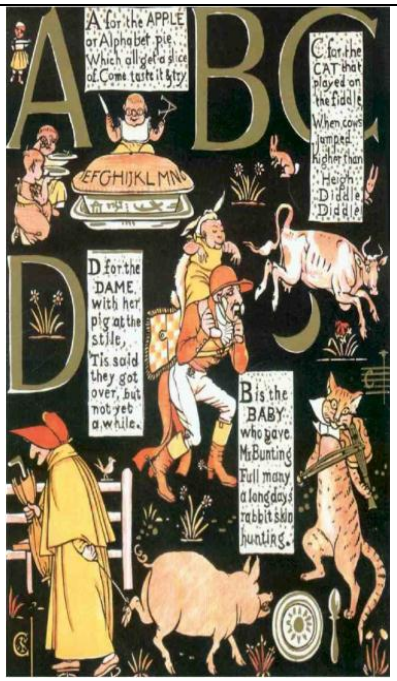
5.4 Panel View (Region Magnification)

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 (`<div>`) 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 id="page086-2">
  <a class="app-amzn-magnify"
    data-app-amzn-magnify=
      '{"targetId":"page086-2-magTargetParent", "ordinal":2}'>
  </a></div>
```

5.5 Optimizing Content for the Graphic Novel Experience

5.5.1 Optimizing Tap Targets

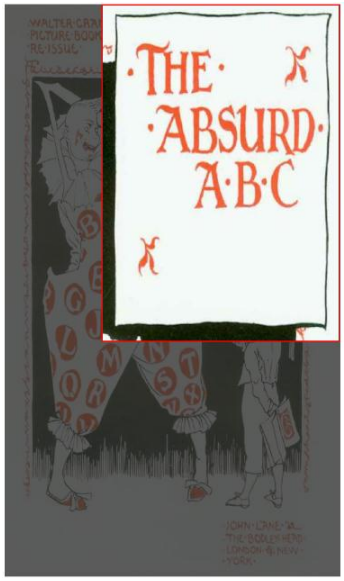
Tap targets should effectively cover 100% of the screen. This ensures that the user gets a magnified experience whenever the user double taps the graphic novel.

5.5.2 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.

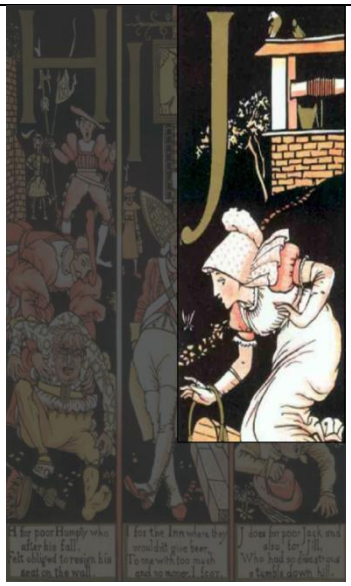


5.5.3 Optimizing for Wide or Tall Action Scenes

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.



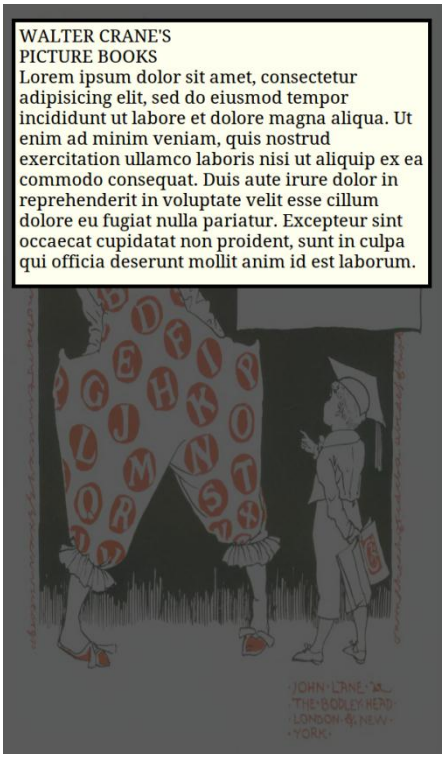
First view panel of an action scene divided into two view panels

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

5.5.4 Optimizing for Large Text Blocks

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;
```

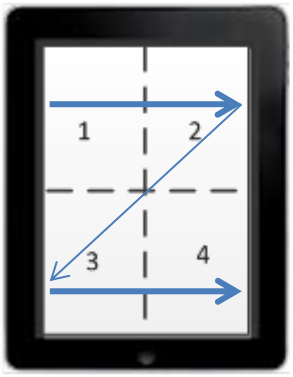
```
  font-size: 150%;
  font-family: "Georgia";
}
```

5.6 Virtual Panels in Comics and Manga

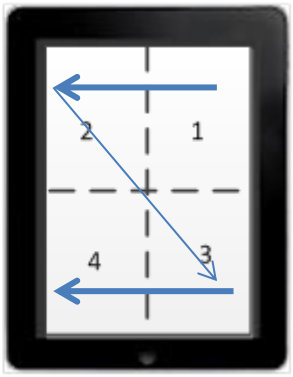
The Virtual Panels feature is activated for comics and manga books in the absence of publisher-provided panels. The `RegionMagnification` metadata is used to identify whether the publisher has provided panel information. If the publisher has designed the content with panels, the Kindle Virtual Panel 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.

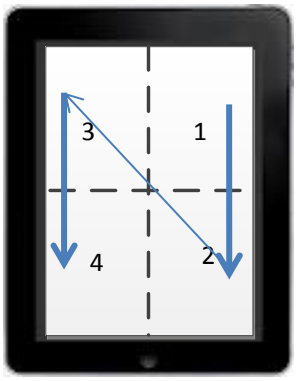
Example:



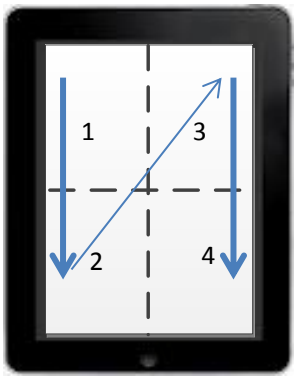
Horizontal-lr



Horizontal-rl



Vertical-rl



Vertical-lr

6 Audio and Video Guidelines

Currently, only Kindle for IOS supports audio and/or video content. Kindle e Ink devices and Kindle Fire do not support Kindle Editions with Audio/Video. To add audio and/or video content to your Kindle book, follow the guidelines and examples below.

The file delivered to Amazon should be an EPUB with self-contained audio and video or a .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).

6.1 Embedded Video

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.

6.2 Streaming Video

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

6.3 Embedded Audio

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.

6.4 Streaming Audio

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

6.5 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.

6.6 Audio Guidelines

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.

6.7 Video Guidelines

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
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

6.8 Audio and Video Metadata

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>
```

6.9 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 6.8 would say “How to create Kindle content (5:01)”.) This information should be embedded in the NavList portion of the NCX file.

6.10 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>
```

However, if you are using KindleGen version 1.2 or earlier, use the following HTML instead:

Example:

```
<a onclick=”document.getElementById(‘audio1’).play()”><img src=”play.jpg”/></a>
```

The Amazon PLAY icon is available upon request.

6.11 File Names Are Case-Sensitive

Kindle books are case-sensitive. When referencing audio and video files within the HTML, be careful about case sensitivity. (**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.)

6.12 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”.)

6.13 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.

6.14 Narration

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.

6.15 Table of Contents

All books 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>
```

6.16 Guidance on 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.).

6.17 Custom Sample File

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.

7 Dictionary Overview

A dictionary is a Kindle eBook (.mobi file) with extra tags for dictionary functionality. Dictionary eBooks:

- Contain a primary index: a list of words or sentences that are sorted in alphabetical order. Readers can quickly search in this list by typing the beginning of the word and selecting the entry they want.
- Are marked as being a dictionary. This tells the Kindle to use this book for the lookup feature on Kindle e Ink devices. The input and output languages of the dictionary must be properly defined. 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.

Note: If you want 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. Specifically, a 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)

7.1 Metadata: Creating the OPF File

The OPF file of a dictionary is similar to other Kindle books, but it also contains specialized metadata tags in the `<x-metadata>` section.

The OPF file of a dictionary sets the source language and the target language. If the dictionary has multiple indexes, the OPF file specifies the name of the primary lookup index.

- `<DefaultLookupIndex>` = Forces the default lookup index, i.e., the index that is opened when the eBook is used as a lookup dictionary from another eBook.
- `<DictionaryInLanguage>` = The language of the book this dictionary is designed to be used on. If you are creating a Spanish- French dictionary, the input language is Spanish.
- `<DictionaryOutLanguage>` = The language of the definitions. If you are creating a Spanish-French dictionary, the output language is French.

Example:

```
<x-metadata>
<DictionaryInLanguage>en-us</DictionaryInLanguage>
<DictionaryOutLanguage>en-us</DictionaryOutLanguage>
<DefaultLookupIndex>Index Name goes here</DefaultLookupIndex>
...
</x-metadata>
```

7.2 Basic Dictionary HTML

In order to make an alphabetical index, you have to use special tags that are not standard HTML to tell the Kindle to index certain words.

The `<idx:entry>` tag is used to mark the scope of an entry in the index. In a dictionary, all definitions should be between `<idx:entry>` and `</idx:entry>`. You can put any type of HTML within this tag. When there is more than one index in the eBook, use the **`name`** attribute to identify an index.

Example:

```
<idx:entry name="xxx">
```

The `<idx:orth>...</idx:orth>` tag is used to delimit the label that will appear in the index list. This is the text that users can enter in the search box to find an entry. For a dictionary, this is typically the headword. Use the `value` attribute to include a hidden label in the entry.

Example:

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

Here is an example of an extremely simple entry that could be part of an English dictionary. In this example, the word "chair" appears in the index list and can be searched 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>
```

7.3 Inflections for Dictionaries

When building dictionaries, you may have multiple inflected forms of a single root word that should access the same entry. However, adding all of these inflected forms under the orthography (pronunciation) of a single entry leads to the generation of a large index, which negatively affects performance and user experience. Kindle has a disinflection engine that uses a set of rules for disinflecting any given word to its headword. The index then has only the headword to look up.

To generate the set of disinflection rules for the dictionary, the input must include some information about the inflections. There are two ways to provide this information: simplified inflection syntax and advanced inflection syntax.

7.3.1 Advanced inflection syntax

Inflections are handled by the inflection index, which is built into the dictionary based on the inflected forms which are tagged in the content using the `<idx:infl>` tag. Inflections are attached to the

orthography of the entry. They must be specified inside of an `<idx:orth>` tag. If an entry has multiple orthographies, each must have its own inflections.

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 `inflgrp` and `name` attributes are optional. The `idx:infl`, `idx:iform`, and `value` attributes are mandatory.

7.3.2 Simplified inflection syntax

For English dictionaries, simplified inflection syntax is a very simple way of giving information about the inflections. Previous versions of the file format supported using the `infl` attribute in either the `<idx:orth>` or the `<idx:gramgrp>` tag and specifying a comma-separated list of inflected forms. This syntax is now deprecated, as it is not as accurate when disinflecting, particularly for non-English languages.

7.4 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
```

7.5 Testing Kindle Dictionaries

7.5.1 Format Testing

This component of dictionary testing ensures that the dictionary is formatted well and provides a good visual experience.

Check the formatting of the definitions by paging through the dictionary and reading several definitions. This can be done on the Kindle Previewer or on any device. Amazon recommends:

- Horizontal rules between each definition.
- Check words for unsupported characters, broken or joined words, proper display of accented characters, symbols, pronunciation guide, etc.
- Start each new alphabet/letter section on a new page.
- The headword (word being defined) should come first, and be distinguished somehow (on its own line, in bold, etc.)
- Check that there are no typos.
- Check that links (if present) are working correctly.
- Check that images (if present) are clear and readable.
- Avoid using tables unless absolutely necessary.

Check the code in the .epub file. This can be done in any text editor.

- Check metadata within the file. It is especially important to have the language tags correct so the dictionary can be used as default to return definitions on books of the specified language. The tags to check are below (Note: `en-us` is American English).

```
<x-metadata>
<DictionaryInLanguage>en-us</DictionaryInLanguage>
```

```
<DictionaryOutLanguage>en-us</DictionaryOutLanguage>
...
</x-metadata>
```

- Make sure the special dictionary XML tags are used properly. The example below shows the correct format.

```
<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 `inflgrp` and `name` attributes are optional. The `idx:infl`, `idx:iform`, and `value` attributes are mandatory.

```
<idx:orth>record
<idx:infl>
<idx:iform value="records" />
<idx:iform value="recording" />
<idx:iform value="recorded" />
<idx:iform value="records" />
</idx:infl>
</idx:orth>
```

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

7.5.2 Lookup Testing

This component of dictionary testing ensures that definitions are returned correctly when using the dictionary as a default to look up words in other books. This component of testing can only be done with e Ink devices (not including Previewer).

- Start by sideloading the dictionary onto the 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 eject your Kindle from the computer.
- Look up a variety of words to see what definition is returned. This can only be done on an e Ink device. To do this, set the test dictionary as default, and look up words in other books. To set this dictionary as the default dictionary:
    - Kindle Keyboard: Go to **Home > Menu > Settings >Menu > Change Primary Dictionary.**
    - Kindle 4 or Kindle Touch: Go to **Home > Menu > Settings > Dictionaries**.

Open a different title, select a word, and note the definition returned. If words are not returning the correct definition (or any definition), there is probably a problem with the HTML tagging.

Suggestions of words to look up include:

  - Root words and conjugations of regular and irregular verbs. Example: walk, walks, walked, walking; go, goes, went, gone, going.
  - Nouns, adjectives, adverbs and their conjugations/declensions if they exist. Example: desk, desks; wolf, wolves; hot, hotter, hottest.
  - Any type of word or grammatical/punctuation convention commonly used in the language and not listed above. Example: contractions.

- Check the index view of the dictionary. This can only be done on an e Ink device. To do this, open the dictionary and start typing a word in the **Search** box. An alphabetized list of headwords should appear and should dynamically update based on which letters are typed.

8 Media Queries

The new Kindle Format 8 (KF8) includes greater support for Cascading Style Sheets (CSS). Currently, Kindle Fire is the only KF8 compliant device, but Amazon will extend KF8 to current generation e Ink devices and apps as soon as possible.

For devices without KF8 support, content creators may require more control over the Mobi 7 experience than provided by the standard conversions supported by KindleGen. To address this need, Amazon has implemented media queries as a way to apply the best CSS styles for each file format. This allows complex CSS formatting to be used for KF8 and more basic formatting to be used for the Mobi format. Media queries are part of the W3 standard. For more information, visit http://www.w3.org/TR/css3-mediaqueries/

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. If the media type is not `amzn-mobi`, `amzn-kf8`, `screen`, or `all`, Kindle ignores it.

8.1 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;
	}		

8.2 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:
	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:
		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;

CSS	CSS Styles Applied to Mobi	CSS Styles Applied to KF8

.2em;
}
 	
@media amzn-mobi
{
.firstletter {
float: 0;
font-size: 3em;
line-height: 0;
font-weight: bold;
padding-right: 0;
}
}
	bold;
	padding-right: 0;
	)
		padding-right: .2em;
		}

8.3 Submitting a Media Query

There are four options for submitting media queries:

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

8.3.1 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;
}
}
```

8.3.2 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">
```

8.3.3 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">
```

8.3.4 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;
```

8.4 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; }
}
```

8.4.1 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>
```

8.4.2 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>
```

8.4.3 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.

9 Kindle Best Practices

9.1 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, Kindle Fire, Kindle for PC, and Kindle for IOS. The Kindle Fire device view displays the content in Kindle Format 8. For installation instructions, see section 2.2.3, Kindle Previewer Software.
2. **Use Kindle devices and Kindle applications.** You can test Mobi 7 content on a Kindle e Ink device and on Kindle applications for PC/Mac/Android. You can text KF8 content on a Kindle Fire.
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.

Once you can read your book, use this checklist to confirm that your Kindle book does not contain blatant errors. (For a finer level of quality assurance, check against the complete formatting guidelines in section 3, General Formatting Guidelines):

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 jump to the correct location in the book. There should be no page numbers in the TOC.
3. Go to any location in the book.
    - **Font size:** Change the font size in the Kindle menu; the book font should change accordingly. Regular text should not be bold or italicized, and its alignment should not be forced.
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.
    - **Tables:** Tables should appear correctly. Make sure that all text in tables is legible.
    - **Page numbers:** There should not be any references to page numbers in the book, including in the cross-references, TOC, and index.
    - **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.

10 Kindle Quality Guidelines

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 exported to HTML.

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

- missing content;
- wrong content;
- typos;
- alignment errors;
- forced font throughout the entire book; and
- proper paragraph spacing.

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

11 Appendices

11.1 Appendix A: HTML Tags Supported in Kindle Format 8

HTML Tag	Description	Supported on KF8-Enabled Devices & Apps	Notes
<!--...-->	Specifies a comment	Yes	
<!DOCTYPE>	Specifies the document type	Yes	Not on e Ink
<?xml?>	This tag identifies a document as an XML document	Yes	
<a>	Specifies a hyperlink	Yes	Links can only refer to items within the same file
<address>	Specifies an address element	Yes	
<article>	Specifies an article	Yes	
<aside>	Specifies content aside from the page content	Yes	
<b>	Specifies bold text	Yes	
<big>	Makes the enclosed text one font size larger than the current or default font size	Yes	
<blockquote>	Specifies a long quotation	Yes	
<body>	Specifies the body element	Yes	
 	Inserts a single line break	Yes	
<caption>	Specifies a table caption	Yes	
	Centers text horizontally	Yes	
<cite>	Specifies a citation	Yes	
<code>	Specifies computer code text	Yes	
<col>	Specifies attributes for table columns	Yes	
<dd>	Specifies a definition description	Yes	
<del>	Specifies deleted text	Yes	
<dfn>	Defines a definition term	Yes	

HTML Tag	Description	Supported on KF8-Enabled Devices & Apps	Notes
	Specifies a section in a document	Yes	
<dl>	Specifies a definition list	Yes	
<dt>	Specifies a definition term	Yes	
	Specifies emphasized text	Yes	
<font>	Alters the font appearance of the text it encloses	Yes	
<figcaption>	Specifies caption for the figure element.	Yes	
<figure>	Specifies a group of media content, and its caption	Yes	
<footer>	Specifies a footer for a section or page	Yes	
<h1>	Specifies a heading level 1	Yes	
<h2>	Specifies a heading level 2	Yes	
<h3>	Specifies a heading level 3	Yes	
<h4>	Specifies a heading level 4	Yes	
<h5>	Specifies a heading level 5	Yes	
<h6>	Specifies a heading level 6	Yes	
<head>	Specifies information about the document	Yes	
<header>	Specifies a group of introductory or navigational aids, including hgroup elements	Yes	
<hgroup>	Specifies a header for a section or page	Yes	
<hr>	Specifies a horizontal rule	Yes	
<html>	Specifies an html document	Yes	
<i>	Specifies italic text	Yes	
<img>	Specifies an image	Yes	
<ins>	Specifies inserted text	Yes	
<kbd>	Specifies keyboard text	Yes	

HTML Tag	Description	Supported on KF8-Enabled Devices & Apps	Notes
	Specifies a list item	Yes	
<link>	Specifies a resource reference	Yes	Can only reference items within the same file
<mark>	Specifies marked text	Yes	
<menu>	Specifies a menu list	Yes	
	Specifies an ordered list	Yes	
<output>	Specifies some types of output	Yes	
<p>	Specifies a paragraph	Yes	
<pre>	Specifies preformatted text	Yes	
<q>	Specifies a short quotation	Yes	
<rp>	Used for the benefit of browsers that don't support ruby annotations	Yes	
<rt>	Specifies the ruby text component of a ruby annotation.	Yes	
<samp>	Specifies sample computer code	Yes	
<section>	Specifies a section	Yes	
<small>	Specifies small text	Yes	
<source>	Specifies media resources	Yes	
	Specifies a section in a document	Yes	
	Specifies strong text	Yes	
<style>	Specifies a style definition	Yes	Can only reference items within the same file
<strike>	Create a strikethrough text	Yes	
<sub>	Specifies subscripted text	Yes	
<sup>	Specifies superscripted text	Yes	
<table>	Specifies a table	Yes	
<tbody>	Specifies a table body	Yes	
<td>	Specifies a table cell	Yes	

HTML Tag	Description	Supported on KF8-Enabled Devices & Apps	Notes
<tfoot>	Specifies a table footer	Yes	
<th>	Specifies a table header	Yes	
<thead>	Specifies a table header	Yes	
<time>	Specifies a date/time	Yes	
<title>	Specifies the document title	Yes	
<tr>	Specifies a table row	Yes	
<u>	Underlines any text it encloses	Yes	
	Specifies an unordered list	Yes	
<var>	Specifies a variable	Yes	
<wbr>	Specifies a line break *opportunity* for very long words and strings of text with no spaces.	Yes	
<nav>	Specifies navigation links	Yes	
<summary>	Specifies a summary/caption for the <details> element	Yes	
<video>	Specifies a video	No	
<audio>	Specified an audio content	No	

*Partial support (see *http://www.mobipocket.com/dev/article.asp?BaseFolder=prcgen&File=TagRef_OEB.htm*)

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 for Kindle Fire)
- Object (Only SVG is supported for Kindle Fire)

- Param
- Noscript
- IFrame

11.2 Appendix B: CSS Selectors, Attributes, and Properties Supported in Kindle Format

CSS Attribute	Supported on KF8-Enabled Devices & Apps	Notes
`/*Comment*/`	Yes	CSS comment
`@import`	Yes	Import external style sheets
`@charset`	Yes	Declares character encoding
`@font-face`	Yes	Allows for linking to fonts
`*`	Yes	Selects all elements
`E`	Yes	Matches any E element (div, span, p)
`E.class`	Yes	Class selector
`E#id`	Yes	ID selector
`E:link`	Yes	Link selector (not yet visited)
`E:visited`	Yes	Link selector (visited)
`Margin`	Yes	Sets margin properties
`Padding`	Yes	Sets padding properties
`Width`	Yes	Sets width of an element
`Height`	Yes	Sets height of an element
`Float`	Yes	Specifies if a box should float on the side
`Clear`	Yes	Specifies which side of an element where other floating elements are not allowed
`Display`	Yes	Specifies the type of box an element should generate
`min-width`	Yes	Sets minimum width of an element
`max-width`	Yes	Sets maximum width of an element
`min-height`	Yes	Sets minimum height of an element
`max-height`	Yes	Sets maximum height of an element
`Clip`	Yes	Clips an absolutely positioned element
`Visibility`	Yes	Specifies whether or not an element is visible
`Border`	Yes	Sets all border properties
`border-color`	Yes	Sets the colors of all four borders

CSS Attribute	Supported on KF8-Enabled Devices & Apps	Notes
`border-style`	Yes	Sets the style of all four borders
`border-width`	Yes	Sets the width of all four borders
`border-top`	Yes	Sets the top border properties
`border-right`	Yes	Sets the right border properties
`border-bottom`	Yes	Sets the bottom border properties
`border-left`	Yes	Sets the left border properties
`border-radius`	Yes	Sets the radius for rounded corner in borders
`line-height`	Yes	Sets the vertical space between baselines
`vertical-align`	Yes	Sets vertical alignment
`Position`	Yes	Sets type of positioning (static, relative, absolute)
`Top`	Yes	Sets the top margin edge for a position box
`Right`	Yes	Sets the right margin edge for a position box
`Bottom`	Yes	Sets the bottom margin edge for a position box
`Left`	Yes	Sets the left margin edge for a position box
`z-index`	Yes	Sets the stack order of an element
`list-style`	Yes	Sets the properties of a list
`list-style-image`	Yes	Specifies an image as the list-item marker
`list-style-position`	Yes	Specifies where to place the list item market
`list-style-type`	Yes	Specifies the type of list item marker
`Color`	Yes	Sets the color of text
`Opacity`	Yes	Sets the transparency of an element
`Background`	Yes	Sets the background property

CSS Attribute	Supported on KF8-Enabled Devices & Apps	Notes
background-attachment	Yes	Sets whether a background image is fixed or scrolls with the rest of the page
background-color	Yes	Sets the background color of an element
background-image	Yes	Sets the background image of an element
background-position	Yes	Sets the starting position of a background image
background-repeat	Yes	Sets how a background image is repeated
background-clip	Yes	Specifies whether an element's background, either the color or image, extends underneath its border
background-origin	Yes	Determines the background positioning area
background-size	Yes	Specifies the size of the background images
Font	Yes	Sets all font properties
font-family	Yes	Allows for a prioritized list of font family names and/or generic family names to be specified for the selected element
font-size	Yes	Specifies the size of the font
font-style	Yes	Allows font style (italic or oblique) to be selected within a font-family
font-variant	Yes	Selects a normal, or small-caps face from a font family
font-weight	Yes	Specifies the weight or boldness of the font
text-align	Yes	Specifies the horizontal alignment of text
text-decoration	Yes	Specifies the decoration to be added to the text
text-indent	Yes	Specifies the indentation of the first line in a text block
text-	Yes	Controls the capitalization of the text

CSS Attribute	Supported on KF8-Enabled Devices & Apps	Notes
transform		
letter-spacing	Yes	Increases or decreases the space between characters in a text
word-spacing	Yes	Increases or decreases the space between words in a text
white-space	Yes	Specifies how white space inside an element is handled
text-shadow	Yes	Specifies the shadow effect added to text
text-overflow	Yes	Specifies whether an ellipsis displays when text content has overflowed its given layout area
word-wrap	Yes	Specifies whether or not the browser is allowed to break lines within words to prevent overflow when an otherwise unbreakable string is too long to fit
Direction	Yes	Sets the base text direction and block-level elements and the direction that cells flow within a table row
border-collapse	Yes	Selects the border model
border-spacing	Yes	Specifies the distance between the borders of adjacent cells
caption-side	Yes	Positions the content of table-caption at the specified side
empty-cells	Yes	Specifies how to render borders and backgrounds around cells that have no visible content.
Outline	No	Sets the outline properties
outline-color	No	Sets the color of an outline
outline-style	No	Sets the style of an outline
outline-width	No	Sets the width of an outline
outline-offset	Yes	Sets the space between an outline and the edge or border of an element
Width	Yes	Specifies the width of the content area of an element
Height	Yes	Specifies the height of the content area of an element
device-width	Yes	The width of the screen in CSS pixels at zoom factor 1.0

CSS Attribute	Supported on KF8-Enabled Devices & Apps	Notes
`device-height`	Yes	The height of the screen in CSS pixels at zoom factor 1.0
`device-aspect-ratio`	Yes	Describes the aspect ratio of the output device
`Color`	Yes	Sets the color of text
`color-index`	Yes	Describes the number of entries in the color lookup table of the output device
`Monochrome`	Yes	Describes the number of bits per pixel in a monochrome frame buffer

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
- E::before
- E::afterCounter-incrementCounter-reset