



What Is EPUB3?

O'REILLY®

Matt Garrish

What Is EPUB 3?

Matt Garrish

What Is EPUB 3?

by Matt Garrish

Copyright © 2011 O'Reilly Media. All rights reserved.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editor: Kathleen Meyer
Production Editor: Teresa Elsey

Cover Designer: Karen Montgomery
Interior Designer: David Futato

Revision History for the First Edition:

2011-09-27 First release

See <http://oreilly.com/catalog/errata.csp?isbn=9781449314545> for release details.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *What Is EPUB 3?*, the image of a Thick-Knee, and related trade dress are trademarks of O'Reilly Media, Inc.

This book was published in collaboration with the [International Digital Publishing Forum \(IDPF\)](#), the global trade and standards organization for electronic publishing that is responsible for development of the EPUB format. EPUB is a registered trademark of the IDPF.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-1-449-31454-5
1317063966

Table of Contents

Preface	v
What Is EPUB 3?	1
EPUB 3 in a Nutshell	1
The EPUB 3 Revision	5
EPUB and Web Standards	7
The Goodies	9
Multimedia	9
Media Overlays	11
Scripting	11
Graphic Content	13
Globalization	13
Accessibility	14
Experimenting	15
The Arrival	15

Preface

Publishing has always been driven by content and the ephemeral “reading experience.” Until recently, the industry has struggled to adapt this experience to the demands of consumers used to getting what they want, when they want, and how they want in the now decade-old digital millennium. A new breed of reading devices has improved the physical experience, and now the new EPUB 3 format is poised to lead a content revolution that will transform the reading experience beyond two dimensions.

What Is EPUB 3? discusses this exciting new format, which is set to unleash a content revolution in the publishing world, both for people who are curious what the hype is about and for pros looking to learn more about the new format. Laden with features the printed page could never offer—such as embedded multimedia and scripted interactivity—EPUB 3 will forever change what a book can be. This article walks you through the format and its place in the digital landscape, explaining why EPUB 3 is set to become the new global standard for ebooks as it also becomes the new accessible standard for ebooks.

Attribution and Permissions

This article is here to help you get your job done. If you reference limited parts of it in your work or writings, we appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN: “*What Is EPUB 3?* by Matt Garrish (O’Reilly). Copyright 2011 O’Reilly Media, 978-1-449-31454-5.”

If you feel your use of examples or quotations from this book falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Books Online



Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at <http://my.safaribooksonline.com>.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://shop.oreilly.com/product/0636920022442.do>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

What Is EPUB 3?

EPUB 3 in a Nutshell

So what exactly is an EPUB? If you follow electronic publishing at all, you've undoubtedly run into the term, but you may have seen or heard it incorrectly being used as a synonym for "ebook"—as a shorthand for talking about *electronic books*. Although the two terms share a common relation in electronic book production, they aren't interchangeable. EPUB, as we're here to discover, is a *format* for representing documents in electronic form. "Ebook," on the other hand, is just an abstract term used to encompass any electronic representation of a book, and includes formats such as PDF, HTML, ASCII text, Word, and a host of others, in addition to EPUB.

Like "ebook," the EPUB name comes from a contraction of terms—in this case, **E**lectronic **P**ublication. The choice of the more general term "publication" was deliberate, and it was selected specifically so that expectations of the format wouldn't be pigeonholed into a single document type. EPUB may garner much of its notoriety from its prominence in the ebook world, but don't think that the format won't be a good fit for your needs just because you're not planning to publish novels.

EPUB was always intended to be a general-purpose document format, and it can be used to represent many kinds of publications other than just books: from magazines to newspapers to journals and on through office documents and policies and beyond. Just about any document type you want to distribute electronically can be represented as an EPUB.

On a practical level, EPUB defines both the format for your content and how reading systems go about discovering it and rendering it to readers (we'll avoid the word "display" for what a reading system does with content, as EPUBs aren't only for the sighted and don't only contain visual content).

But perhaps the best way to understand what goes into an EPUB is to quickly break down the creation process:

1. The first step in making an EPUB is to create your *content document(s)*. These must be either XHTML5 documents, SVG images, or a mixture of the two (we'll skip how to use nonstandard formats for now). If you've created web pages before,

marking up your EPUB content will be a breeze. And if not, there are many sophisticated layout programs that can make the process no more painful than formatting a Word document. (Some of these programs can even do the entire EPUB creation process.) Note, however, that your content document will typically reference many other types of resources required for its proper rendering—CSS style-sheets, images, multimedia clips, script files, and so on—all of which you will need to keep together to bundle up into the finished distribution file later.

2. Once you've crafted your content, the next step is to create the *package document*, a special document used by reading systems to glean information about your publication—for ordering in your bookshelf, to render the content, and the like. The first step in creating this file is to list all of the resources you assembled in the content creation step in the *manifest section* of the package document (most of these resources will be in the EPUB file, but some, such as audio and video, can live on the Web). Reading systems need this list to determine whether a publication is complete or not and to discover which remote files will have to be retrieved. All your publication metadata (title, author, etc.) also goes in this file, consolidating it in a single, common location so that it can be easily extracted and used in distribution channels and by reading systems. You also have to include the default reading order in the *spine section*—a sequential list of your content files, from the first one to display to the last. And finally, the package document identifies which of the resources represents the *navigation document* (the table of contents).
3. The last step is to zip up your content documents, associated resources, and the package document into a single file with the *.epub* extension for distribution. This process isn't quite as simple as a standard zipping, however: a special *mimetype* file has to be added first to indicate that your zip file contains an EPUB and not something else, and a file called *container.xml* has to go in a directory named *META-INF* to tell reading systems where to find your package document.

But that's really all there is to an EPUB file under the hood. There are many details intentionally glossed over here for the sake of simplicity, but if you feel comfortable with the concept of an EPUB as a predictable, discoverable container of your content, you're well on your way to understanding the power of EPUB. And if you read the list above in reverse, you can understand how reading systems work: they examine your zip container, determine it's an EPUB, find the package document, and from there discover how to render the resources to readers.

What this quick overview of the content file should also show is that the EPUB format unabashedly draws many of its capabilities and its versatility from web technologies, but also that the Web alone doesn't tell the whole story of EPUB. Without the complementary technologies the EPUB format brings under its common umbrella, the ability to create distributable publications would be severely limited.

Some of the technologies used in EPUBs have been specially developed by the International Digital Publishing Forum (IDPF; <http://idpf.org>)—the organization that

maintains the EPUB suite of specifications (see “[The EPUB 3.0 Specifications](#)” on page 4), and whose membership includes [publishers](#), [distributors](#), and [reading system developers of all sizes](#). But most of the standards that have been leveraged are internationally recognized and should at least sound familiar to most readers. The key technologies you’ll find in EPUB 3 publications include:

XHTML5

For representing text and multimedia content, which now includes native support for MathML equations, ruby pronunciation markup, and embedded SVG images

SVG 1.1

For representing graphical works (for example, manga and comics)

CSS 2.1 and 3

To facilitate visual display and rendering of content

JavaScript

For interactivity and automation

TrueType and WOFF

To provide font support beyond the minimal base set that reading systems typically have available

SSML/PLS/CSS 3 Speech

For improved text-to-speech rendering

SMIL 3

For synchronizing text and audio playback

RDF vocabularies

For embedding semantic information about the publication and content

XML

A number of specialized grammars facilitate the discovery and processing aspects of EPUBs

ZIP

To wrap all the resources up into a single file

This list contains many new additions and upgrades from what you’ll find in the EPUB 2 publications on the market right now. EPUB 2 had support for XHTML 1.1, CSS 2, and the XML and ZIP standards, but lacked the ability to use the rest of the technologies listed (as we’ll come back to again). The last thing the publishing world needs is more proprietary standards, and EPUB has wisely avoided the proverbial re-inventing of the wheel by adopting the best existing technologies, a design approach it has maintained from version to version.

A great benefit of EPUB (especially for those new to electronic publishing) is that it’s not a whole new beast that you have to learn from the ground up before you can hope to do anything interesting. Almost all of the technologies it uses are common web and XML standards, not obscurities from the fringes of the spectrum. If you’re already familiar with these technologies, jumping into the core of EPUB is a simple task. And

even if you're only aware of these technologies at a higher level, you're still well on your way to understanding the format. (And if they seem foreign and scary, don't worry; we'll be elaborating on their uses as we work through all EPUB 3 has to offer.)

In this day and age, the Web skills needed to create basic EPUB content (XHTML and CSS) are not only ubiquitous but also a daily part of most people's digital lives. So if you're only interested in how to publish documents and not in all the details under the hood, you're no more excluded from digital publishing than you are from Web publishing. Add all the authoring tools at your disposal, and you don't have to be a programming geek to create top-flight EPUB content.

The EPUB format is also specifically designed to be free and open for anyone to use without having to sift through a litany of patent encumbrances and restrictions; it just doesn't make sense for an organization that promotes the global electronic publishing trade to tie its technology up and limit its usability, after all. EPUB's widespread adoption is also due in no small part to the fact that basic text editing tools can be used to create publications, and the EPUB 3 revision of the specification has not deviated from this core tenet.

And there you have EPUB in all its minimal glory.

But, of course, minimal EPUB 3 barely scratches the surface of the format.

The EPUB 3.0 Specifications

Although EPUB 3 aggregates a number of technologies, an EPUB is not just a loose collection of these technologies. The term "EPUB 3" actually encompasses four separate specification documents, each of which details an aspect of how the employed technologies interact. This allows anyone to author an EPUB without struggling through all the related specifications and allows the development of reading systems that can predictably process them. Another way to think of EPUB 3 is as the glue that binds these technologies into a usable reading experience.

The number and size of the specification documents is often intimidating to novices to electronic publishing, but once you understand which aspect of the content creation and rendering process each handles, they're not very difficult reads. They break down as follows:

EPUB Publications 3.0

The Publications specification defines the XML format used in the package document to store information about a publication. As noted earlier, the package document contains metadata about the publication (such as the title, author, and language), lists all the resources used, defines the default reading order, and indicates where to find the navigation document.

The Publications specification also defines general content requirements that all EPUBs must adhere to, such as required content types and when and how to provide fallbacks for content that isn't guaranteed to render on all devices.

EPUB Content Documents 3.0

The Content Documents specification defines profiles of XHTML5, SVG 1.1, and CSS 2.1 and 3 for use in authoring content. A *profile* can perhaps best be described as a snapshot of the specific functionality that you are allowed to use (that is, you may not get to use everything defined in those specifications just because it exists). If you skip or skim this specification, not only might you wind up using illegal elements, styles, and features, but you also might miss the additions that EPUB makes to improve the reading experience.

The Content Documents specification also defines the format of the special navigation document. This document contains the table of contents for a publication, but it may also include other navigational aids, from tables of figures and illustrations to specialized tours of content.

EPUB Media Overlays 3.0

For those already familiar with EPUB 2, the Media Overlays specification is the new kid on the specification block. The ability to include audio content in EPUB 3 does not limit you just to embedding audio clips in your documents. Media Overlays take advantage of the SMIL specification to enable the text content rendered in the reading system's display area to be synchronized with audio narration, so that, for example, words can be highlighted as they are narrated.

EPUB Open Container Format (OCF) 3.0

And, finally, the Container specification defines how you bundle all your resources together into a single file. As we noted above, creating an EPUB file is more complex than just a simple instruction to zip up content, and this specification defines the discovery aspects discussed previously.

The EPUB 3 Revision

So now that we know what an EPUB is, it's time to look at what drove the need for a new revision. If evolution is the cornerstone of life, that's certainly no less true in the electronic world. If you can't adapt—or fail to adapt in time—you're destined to join the ranks of the Netscape Navigators, OS/2 operating systems, and WordPerfect office suites of the world, as a warning to future technology developers that nothing lasts forever, and never in its original form. In this light, EPUB 3 is more than just bug fixes and tweaks from the last version; it represents a major change in what an ebook can be. It's a whole new beast, you might say.

The ebook market has been going through its own kind of hyper-evolution in the mere four years since EPUB 2 was released, and a flurry of new devices and document formats have come and gone in that time. E Ink technology was all the rage in 2007 when Adobe, Amazon, Sony, and others were entering the market, however, and EPUB 2 arrived to meet the new needs of these portable reading devices, with improved presentation capabilities, better navigation, support for [DAISY](#) accessibility features, and some advances in global language support. But EPUB 2, like its predecessor and

contemporaries, remained a static format, in that its core only allowed for the reading of basic text and image documents.

EPUB 2 *was* an advance, and for a time it served the needs of the market well. It might even have had a longer run had dedicated E Ink devices remained the predominant choice for reading. But just as readers were abandoning their paper books, tablet computers came storming onto the reading scene, not only adding visual and aural dimensions lacking from E Ink's shades of gray, but also including the appeal of merging many capabilities into a single device—reading, browsing, gaming, and music, to name just a few. Dedicated E Ink readers suddenly didn't seem so cool anymore, nor did bland content that looked just like a printed page.

Although the primary effect of this new progression in the way content is read was to expose the multimedia shortcomings of current formats, ebook content had been under assault for a variety of other reasons, too. The ebook community had been clamoring for the ability to make interactive content, for improved global language support, and for better accessibility features, as well as a whole host of other changes to the status quo.

The IDPF would have been foolish to sit tight on the EPUB 2 specification in the face of its own constituents' needs, so a revision was inevitable. Unlike Amazon's Mobi format for its Kindle devices, which is able to rest on its progressively aging technology because both the content and rendering are tightly controlled by a single company, EPUB requires the IDPF to take a much broader perspective with its development because of its diverse community. But this requirement has also kept the format at the leading edge of ebook technologies throughout its history. While EPUB 2 was, to use a common euphemism, good for its time, that time, dominated by the initial thrust of reproducing the static print page in electronic form, has passed.

EPUB 2 didn't suffer only from the lack of new features that HTML5 now offers; not every problem a format faces can be solved by new technology alone. Accessibility is one obvious example in EPUB 2. In retrospect, the way that features of the DAISY standard got bolted onto the specification led to aspects never being fully or properly implemented by publishers or developers (the DTBook grammar for content) and others being misunderstood or conflicting with general-purpose needs (the NCX navigation file being used for reduced tables of contents, undermining its use by the target audience). The EPUB 3 revision also presented a chance to revisit issues like these that had appeared or been left open since the previous revision, to see if new and better solutions were now possible.

The EPUB 3 working group began the revision of the specification in the summer of 2010 and had a one-year timeline to overhaul the format, in order to address [all these issues and more](#). The result is that the revision has seen major improvements in virtually all the key functional areas: integrated audio and video support (as we've mentioned), accessibility features are much more tightly entwined in the specification now, global language support mechanisms are more numerous and also more integrated,

publication-level metadata allows much richer expressions, and so on down through the original charter (but we'll get back to these in more detail in [“The Goodies” on page 9](#)).

This isn't to suggest that the EPUB 3 revision got everything perfect. The metadata world is in flux, and many had hoped that a more standards-oriented solution would be forthcoming. Video content support is divided between the H.264 and WebM codecs, leaving the specification without a single video type that all reading system developers could agree to support. The comic and manga communities still are looking for more improvements in supported formats and rendering. In other words, the evolution of EPUB doesn't end with the current revision, and thought is already going into improvements.

If you want an open, community-driven, standards-compliant specification that sits at the forefront of what an ebook can offer, however, there is no other solution but EPUB 3.

EPUB and Web Standards

The importance of web standards to EPUB has been mentioned a few times now, so before jumping into all the new features of EPUB 3, it's worth a brief diversion to look at why this integration with the browser stack (XHTML, CSS, and JavaScript) is so important to the format.

Saying that EPUB is tied to the browser stack is not to suggest that, back in 1999 when the first version of the specification was released, the Internet Explorer and Netscape Navigator browsers of the day were the intended reading targets. It simply means that the same open standards that underpin the World Wide Web were seen as both a natural fit for digital reading and integral to the long-term survival of the format. They were a natural fit because HTML had already swept the landscape as the king of *re-flowable content*—that is, as a format that scales easily to the viewing area available on any device, unlike fixed formats like PDF that rely on exact positioning—and they were integral to the long-term survival of the format because, even in 1999, there was little doubt that the Web was here to stay.

Being tied to the browser stack has also provided a number of technical advantages that have helped EPUB thrive while other formats have ingloriously come and gone from the landscape. For one, it has simplified the development of reading systems because rendering engines like Gecko and WebKit have been readily available to expedite the process (a rendering engine is the same core that browsers are built on). You don't have to be a massive multinational corporation to create a device capable of rendering EPUBs.

With these engines also comes CSS support, without which no format could hope to reach a global audience or provide a rich viewing experience. A document without CSS is little more than headings and text, as the Kindle's Mobi format largely exemplifies.

Mobi gives you the same reflowing capabilities of EPUB, but Mobi suffers from its reliance on the now-antiquated HTML 3.2 element set. You can emulate some styling using the old HTML features and the built-in additions, but, as anyone who's tried to create a Mobi file knows, your documents are a pale comparison to what they could look like with proper CSS.

And wouldn't you rather create your ebook for advanced viewing and have it scale down to the capabilities of the device than have to reformat your book to work on different devices? EPUB's tighter binding to standards gives you that capability, and it is undoubtedly a major influence on why Amazon accepts EPUBs from publishers and is able to automatically reformat them for redistribution as Mobi files.

And although JavaScript was always a part of the browser stack, EPUB 3 finally brings this black sheep back into the ebook fold to enable interactivity in documents (despite the dangers it brings with it, as we'll discuss later). A widely implemented and understood programming language waiting in the wings...again, it's easy to see why the decision was made to hitch EPUB to the web standards wagon.

But being hitched to web standards isn't a free ride. Much work has gone into making the format predictable and processable by reading systems. Much work has also gone into patching the holes that HTML doesn't address for document publishing. HTML on its own, for example, has weak mechanisms for richly annotating your data (that is, it has a simple set of tags but no standardized way to say anything about them, such as whether a `section` element represents a chapter or part or something else).

Whether your content will render in your browser is also much less of a concern in the wider web world, where you're expected to download plug-ins and applications on an as-needed basis. Publishers and reading system developers, however, have to worry about irate customers buying books that won't render on devices that may only rarely connect to the Internet or that may not provide the ability to install missing software. Even seemingly simple things like navigation and reading order have gone unstandardized for years in the wild West that is the WWW, meaning that this is work that has subsequently had to go into EPUB.

And building EPUB on top of other technologies always leaves the door open to uncertainty, as you can't control the future directions of those technologies. In the case of the HTML5 and CSS3 specifications, which may still change in unpredictable ways, that risk is considerably more immediate than the typical uncertainty of what you might have to do in a future revision if a technology moves off in an incompatible direction. That's not to suggest that EPUB 3 is built on a house of cards—support for both HTML5 and CSS3 is fast developing and the specifications themselves are slowly but surely moving to stable ground—but you always do need to be aware as you author an EPUB that not everything other technologies make available can be used right now.

It's also not the case the future of EPUB and HTML5 are on the same path. The HTML world is increasingly moving away from document publishing and into mobile application development and cloud-based web applications. While many of the changes this

altered focus entails may positively impact the EPUB sphere—cloud-based reading systems are already a reality, for example—the change also means an increasing role for the IDPF as both a maintainer and promoter of ebook technologies. Following only what the Web does certainly seems an increasingly less and less viable means of publishing.

But what the future of the Web holds is anyone's guess, so we'll end the speculation here.

The Goodies

So far I've been alluding to the fact that EPUB 3 has much more to offer than its predecessor and the competing formats on the market, without offering specifics. Well, the time has come to really get into how EPUB is set to change the traditional print-based publishing landscape.

Multimedia

That EPUB 3 is based on HTML5 means that you have the **audio** and **video** elements available to you for embedding multimedia content directly into the flow of your document. Always wanted to create a science textbook with audio directions and video demos so that students don't have to muddle their way through graphics and lists of instructions? Well, there's nothing stopping you now. Want to add an audio clip of Martin Luther King Jr.'s "I Have a Dream" speech to your textbook on the Civil Rights movement? Now you can with no special player needed. EPUB 3 requires support for the MP3 format, so insert your **audio** element and reference your clip and you're done. You can safely sit back and relax as your book goes to market, knowing that all your readers will be able to listen to it.

Want to go a little fancier and add the actual video clip of the speech so that your students get a more compelling presentation? No problem, same story...almost. Embedding your clip is still as easy as adding the HTML5 **video** element, but, as we touched on briefly earlier, in a rare exception to the specification, there isn't a single video format that all players are guaranteed to support, so you have to be a little more careful.



The IDPF is currently promoting two formats for video—H.264 and WebM—but is not requiring the use of either. The reason: the working group whittled the possible options down to these two formats but a consensus couldn't be achieved on support for one, the other, or both. Some members prefer H.264 because it has widespread adoption, but others objected because the format is patent-encumbered. Another segment preferred WebM because it is patent-free, but others objected that it's not a standard yet—and it's not clear if it will become the de facto standard for video on the Web just because it is patent-free. The EPUB working group fully intends to return to this issue again when consensus can be achieved, but for now you need to be aware that you may need to target video to the expected device during distribution or run the risk that your videos will not play. More suggestions will undoubtedly be forthcoming as authoring guidelines are developed, so for now the only advice is to keep an eye on the work of the IDPF.

This kind of seamless integration has been long overdue. Adding audio and video content will make for some really fantastic new multimedia experiences, never dreamed of in the print world. (CD supplements were always better Frisbees than resources!)

But there's always the size caveat that comes with audio and video formats. Sprinkle a little clip here and there in your content and you may not think it's such a big deal, but start adding copious amounts of audio and video content and you're looking at distributing an EPUB file that could easily be hundreds of megabytes in size, if not more.

The EPUB working group recognized this issue in bringing in multimedia content and made an exception to allow audio and video resources to live outside the container. But that leaves a bit of a conundrum for content creators. If you bundle all your multimedia content in the container, your readers are going to have a long download to their devices (and depending on the system, a download to a computer and then an upload to each portable device), but they get a seamless reading experience (the audio and video content is immediately there for them when they want to play it). If you host your audio and video resource out on the wider Web somewhere, however, your readers will get their books faster and be able to start reading faster, but what happens when they reach each multimedia experience is a bit hazy. If their reading devices are wired and was able to fetch the content while they were reading, no problem. If their devices aren't wired, were taken offline before grabbing all the extra resources, or don't fetch resources until they're played (it's impossible to say which scenarios devices will support until they start appearing), your readers are going to be a bit unhappy.

It's also a real possibility that distribution channels may impose size restrictions, which would at least directly answer the question of how much you can fit into your ebook (or, at a minimum, force you to consider potential quality loss for better compression). With multimedia such a new experience, only time will tell how these interacting forces and requirements will play out.

Media Overlays

When talking about multimedia content, it's hard not to talk about the new Media Overlays specification at the same time. *Overlays* are a new concept for EPUB, but they're an old concept in the accessibility and talking book worlds. An overlay is what its name suggests: a method for layering audio content on top of the traditional print content of a book.

Overlays are achieved via a profile of the SMIL specification. But it's a very simple profile, so don't worry if you know how complex SMIL can get. (You're not about to lose a month of your life learning the full specification.) If you're at all familiar with DAISY Digital Talking Books (DTBs), you're already well on your way to being able to create an overlay (and even if not, they really aren't complicated).

EPUB overlays differ from DTBs in that the information for the overlay is cleanly separated from the content in a separate overlay document. Without getting too deep into the details, the overlay document functions like a map to the text and audio equivalents. The granularity of this synchronization is up to the content creator or developer. You could choose to synchronize at the paragraph level for a novel; for a child's learn-to-read book, you might mark up your file to the word level, so that the reading system highlights each word as it is narrated.

Although this might sound difficult (and a bit magical) to do, it doesn't require anything more than good markup and some time to synchronize element IDs with audio resources. The overlay maps the element IDs in your file to the corresponding audio resources (which could be the start of a new audio file for each paragraph but would probably be a time offset in a single file for a whole chapter, for example). Reading systems use these maps to move the highlighting on the text to the element with the specified ID as your audio passes the specified offset or the next resource gets loaded, allowing the reader to follow.

But you don't have to think in terms of synchronization. You can also think of overlays as creating two-in-one audio and text books. A reader could choose to listen to your audio and ignore the text; some devices may not even have a screen for reading. Or a reader could disable audio playback and just read the text. You could even develop the two-in-one route in house, offering an audio book version and automatically stripping the overlay document and related audio resources to publish the standard ebook edition to distribution channels.

Scripting

After multimedia, scripted interactivity is probably the most longed-for feature that EPUB 3 brings to ebooks. Prior to this version, EPUB prohibited the use of JavaScript in ebooks, and as you might imagine, there were very strong opinions both for and against making the change to include it.

There are the obvious issues of malicious code and interoperability problems arising from the different rights and privileges that any reading system might grant, but scripting also has the potential to undermine the IDPF's goal of making EPUB 3 an accessible format. But although these are all real issues, they are not issues that anyone developing for the Web doesn't face, and the march to interactive publications can't be stopped because of what people ultimately *might* do.

In opening the door to scripting, however, the EPUB specification sets out ground rules for its use, to try and prevent the most egregious abuses. You won't find EPUBs that are nothing but masses of scripted code and empty elements, or at least not ones anyone can honestly call valid to the specification. The ocean of overly wrought and needlessly scripted tag soup that accessible user agents struggle to process is not welcome in the EPUB publishing world.

To try to mitigate the accessibility issues that scripting brings into play, the notion of *progressive enhancement* was formalized in the specification to define how scripting can be used. What this means is that, while you are free to experiment with interactivity, you must ensure that any such scripting does not interfere with the ability of a reader to access the content of your EPUB. Or, put simply, if a reading system has no scripting abilities whatsoever, someone reading your EPUB must still have to access the same content as if it did.

Although this restriction might seem a bit imposing at first, ask yourself how often you use scripts to dynamically alter the text of your documents. Probably rarely, if ever, and with a little care and thought, creative alternatives to inaccessible practices are never hard to find. And don't confuse this requirement with a restriction on all document manipulation. If you wrote a book on .NET programming, for example, and wanted to script an option to let a reader hide either all the VB or all the C# examples, so long as your document didn't initially hide either (or worse, both), it would likely be perfectly compliant. If you tag and label your examples in a way that allows for easy identification, someone using an accessible reading device will be able to navigate around the ones he or she isn't interested in. Hide all the VB examples by default and you may take them away forever from some readers.



When adding scripting, be aware that not all devices will offer the same level of support and access and that readers may further be able to restrict what they allow their devices to do. To facilitate graceful degradation, EPUB 3 includes a new custom JavaScript object called `epubReadingSystem`, which can be used to query the reading system—to find out its name, its version, and whether it supports scrolled or paginated content—and the EPUB features it supports. This functionality allows you to either alert a user that your scripting can't run or to program alternatives if you know different devices use different capabilities to achieve the same result. For more information, refer to the Content Documents specification.

It's going to be fun to watch the development of scripted ebooks. The static print mentality has largely ruled ebook development, so it's a bit like the dawn of a new age. Scripting will no doubt first take hold in commonplace ways, such as for the automation of `canvas` tags to embed interactive demos or games or to include calculators and the like. Dynamic test booklets that can grab questions from a remote server or give you on-the-spot results of an included test might be getting a little more imaginative. But it's impossible to predict the innovative ways people will find to use this functionality; creative authors will no doubt find ways to improve on the traditional novel itself.

Graphic Content

The inclusion of SVG as a core media type means that SVG-only publications can now be created, opening the door to manga and comics and other image-driven publication types without the old headache of having to embed those images inside of XHTML documents for the sake of being compliant to the specification.

Although perhaps not so groundbreaking as it might seem (electronic versions of manga and comics aren't exactly new), this does take EPUB one step closer to the simplicity that CBR and CBZ offer, for example. But instead of relying on file naming to get the correct rendering and the possible inclusion of `ComicBookInfo` metadata, you gain both the industry-standard metadata of EPUB and the navigation document for more reliable rendering. Of course, not everyone is going to see these improvements as improvements, but from a publisher's perspective, an EPUB represents a more reliable distribution mechanism for this kind of content than the image-only formats provide.

Not everyone is a fan of SVG, however, and discussions have been ongoing in the EPUB working group about additional ways to improve the reading experience in this area. Some members are looking to expand the range of image formats that can be used without having to add fallbacks (but accessibility remains a concern in this regard), while others are looking at ways to improve the reading experience on devices unable to display the full panel. The IDPF will be hosting workshops in the near future to explore these and other issues, and while it's impossible to predict what additional benefits might be integrated moving forward, it's clear that EPUB will continue to evolve in this area.

So although it's fair to say that EPUB 3 has made strides, there is still much work to be done. It's also fair to say that these formats have many proponents in the EPUB working group: if you aren't satisfied now, just keep checking in, as the situation will only improve moving forward.

Globalization

CSS3 support is one key behind EPUB 3 being the format for a global audience and global production. Amazon's Mobi format is fine, for example, if you only want to publish romance languages, but what are you going to do when you have to render a

book's writing vertically? What are you going to do if your audience reads from right to left and bottom to top? They can just learn to read the way we do, right? (Let's hope the silliness of that statement is abundantly obvious!)

But new CSS features alone don't make a global standard. EPUB also supports ruby pronunciations in XHTML content documents. EPUB includes HTML5's bidirectional support mechanisms. EPUB also has hooks in at the package file level to assist reading systems in determining the correct page rendering direction. EPUB metadata allows information like titles and author names to be represented in different scripts so that they can be represented differently for different markets. EPUB allows the embedding of WOFF and TrueType fonts so that the limited selection of fonts on a typical reader doesn't lead to characters "going missing." EPUB 3 is truly unique as a reflowable format in the breadth and scope of language and writing styles it now supports.

This is one more way that EPUB is knocking down publishing barriers. You may not think about distributing your book in another market, but what happens when you do? EPUB will be there for you. Or on a larger scale, if your company publishes in more than one country, wouldn't it be great if all your production houses could standardize on a single universal format? EPUB will be there for you, too.

Accessibility

But being a global standard means more than just accommodating the writing methods and languages of the world's diverse peoples. In a rather bold move, the IDPF has made accessibility support an integral part of the new specification and not just a tack-on, as has been HTML's long-time failing.

But that doesn't mean that just because you create an EPUB you've ipso facto done your part in making the world a more accessible place. Work is still required, but the work it takes to create an accessible document is rarely more than the work it takes to create a well-structured document good for data exchange and transformation.

Judicious use of the new EPUB `type` attribute for semantic inflection will go a long way toward improving the quality of your books (for example, adding to HTML5's `section` elements to indicate if they are parts, chapters, indexes, or something else; as we noted, HTML5 doesn't offer in a standardized way to do this now). Including a navigation document with the complete structural outline of your book is another step; the specification now provides a declarative means of hiding unwanted heading levels from sighted readers, a significant improvement over the deprecated NCX. There are a number of ways to improve text-to-speech quality, too, such as being able to include pronunciation lexicons and embed SSML in content documents.

Accessible content, as many publishers already know, is not something that you can just decide whether you feel like doing; it is a legal requirement in many jurisdictions for many types of documents. If you put the effort into creating accessible EPUBs up front, you can save the time and energy of having to go back and reformat your content

later. And with EPUB 3 poised to supersede DAISY's DTBook grammar, American publishers required to submit NIMAS-compliant versions of their ebooks to NIMAC, for example, should recognize a great potential here to only have to create a single EPUB format moving forward. Once again, EPUB 3 shines as a kind of chameleon format that can be all things to all people.

Experimenting

Last, but certainly not least, EPUB 3 is unique in that it allows you to experiment with your content while ensuring that your readers don't face the penalty for having the wrong device to render it. What that means is that you aren't limited to making a book that is only either XHTML or SVG (or a mixture), but rather you can author in any format you can find a reading system to support. The caveat, as we saw with audio and video, is that you have to provide a fallback in XHTML or SVG (kind of a comedown if you were getting ahead of yourself), but that's how the specification ensures content renderability. You don't want angry customers clamoring for a refund because your content doesn't show up on their devices!

But full spine-level content replacement is probably not going to garner as much attention as the ability to alternate your markup at the content level. Within XHTML documents you can make use of the EPUB `switch` element (kind of like a declarative version of the programming construct) to improve on the reader's experience. Write a chemistry book and you could include fancy ChemML models that degrade gracefully back to mundane XHTML text representations or images. Everyone gets a bit of something, but anyone whose device supports your preferred format gets a lot more bang for their buck. This may not make everyone who doesn't have support happy, but it's a lot better than the situation with many formats where you just don't get to see anything when it comes to support for inline grammars.

The Arrival

And that brings us to the big question: who is going to support EPUB 3 and when? Because it's community driven, and because it has a very large community, people have already begun experimenting, even though the specification isn't yet a formal recommendation. Apple iBooks, for example, already has early support for the overlays feature of EPUB 3, and read-along children's books are already available.

With EPUB 3 now in the final stages of the approval process, it will receive recommendation status in fall 2011. In that light, it's not a stretch to predict that EPUB 3 reading systems will be on shelves—and upgrades to your current systems available—by early 2012 (if not sooner). Many industry heavyweights—like Apple, Sony, Google and Adobe—have been involved in the specification development, and all are expected to begin working on implementations once the finished specification is available to develop against.

The specification has also been crafted with backward compatibility of content in mind. Duplicating content for EPUB 2 and 3 readers is not anything anyone wants to do, so to that end, EPUB 3 has maintained minimal compatibility with EPUB 2 reading systems to ensure that anyone using an older system to read a new book is still (reasonably) able to do so. That's not to suggest that all EPUB 3s will render seamlessly on older readers (in particular if you are taking heavy advantage of all the new features EPUB 3 has to offer), but if your EPUB 3 is a simple novel—with just chapters and paragraphs and using basic CSS styling—it should render equally well on an EPUB 2 reader as on a 3.

Depending on when you read this article, you may be able to rush right out now and start creating all the great new kinds of content discussed. Otherwise, you'll just have to wait a little longer—but not much! And if this article has done its job, in either case, your head is already swimming with the new possibilities EPUB opens up, whether in content creation or production.

About the Author

Matt Garrish has been working in both mainstream and accessible publishing for nearly 15 years, abandoning editorial dreams after falling in love with structured data. He was the chief editor of the EPUB 3 suite of specifications and also played a key role in the design and writing of the new revision of the ANSI/NISO Z39.86 standard for authoring accessible content.

