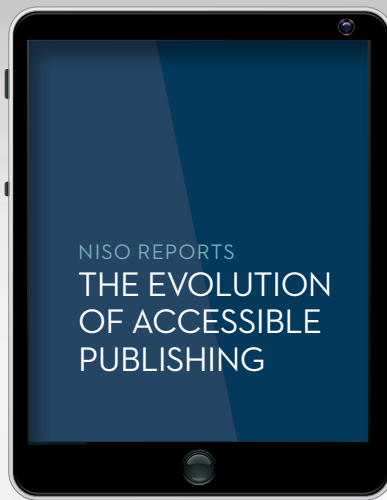
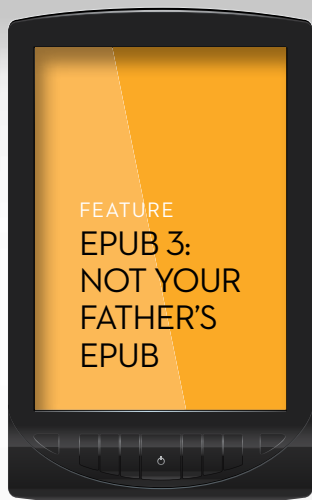


ARTICLE
EXCERPTED
FROM:

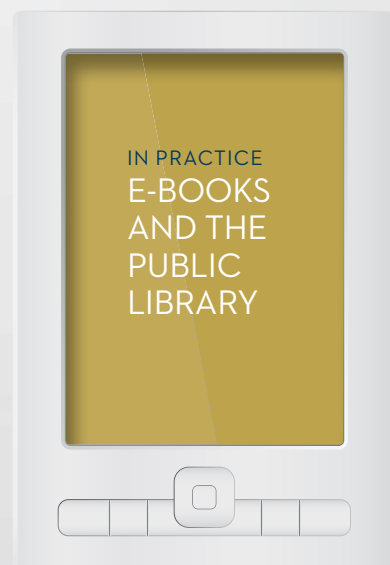
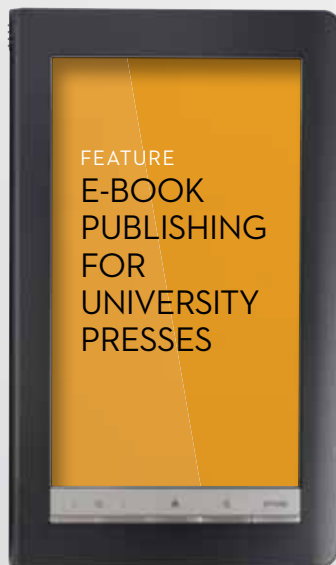


ISQ

INFORMATION STANDARDS QUARTERLY

SPRING 2011 | VOL 23 | ISSUE 2 | ISSN 1041-0031

SPECIAL EDITION: VIEWS OF THE E-BOOK RENAISSANCE



NISO
How the information world
CONNECTS

EPUB 3

(Not Your Father's EPUB)

A glowing red Pandora's box, a classic symbol of hidden dangers. The box is open, with a bright light emanating from the gap between the lid and the base. The lid is slightly ajar, and the light is most intense at the top edge. The box has a keyhole in the center of its front face. The name 'BILL KASDORF' is printed vertically on the front face, below the keyhole. The box is set against a dark background, and the light from the box illuminates the surrounding area.

OPENING PANDORA'S BOX IN THE WORLD OF E-BOOKS

BILL KASDORF



HTML 5

With all its new capabilities—handling rich media, complex layouts, scripting, global language support, MathML, synchronizing text and audio, and a host of other new features—EPUB 3, the new generation of the EPUB specification just issued by the IDPF (the International Digital Publishing Forum), may seem to be opening Pandora’s Box in the world of e-books.

I’d rather make the case that it’s trying to keep the lid on it—or at least trying to open the lid carefully, in the hope that all the creatures bursting out can be made to behave in a civilized way. That may seem to be a vain hope, but it’s a noble one. And I’m betting it will be successful.

Fundamentally, the revision of the EPUB specification was a response to the seemingly out-of-control pace of change in the world of e-publishing. Publishers are chafing at the limitations of today’s two fundamental e-book formats, PDF and EPUB 2.0.1, as advances in the wider world of the Web, the proliferation of new devices, and, most of all, the ubiquity of smartphones create both demand for more sophisticated functionality and impatience with solutions that fall short.

Amazon didn’t name its e-reader “Kindle” by accident: Jeff Bezos wanted to light a fire, and he did. The stunning success of the iPhone was succeeded by the even more stunning success of the iPad. And in the meantime, we’ve all become Google-eyed. Suddenly, everybody wants to get everything (information, entertainment, instruction) in

CONTINUED »



It's often a revelation to find that an EPUB file is really a zip file: if you change the .epub extension to .zip, you can see all the goodies inside—metadata, XML files, images, and so forth. (Go ahead, try it.) But it's not *just* a zip file, it's a particular *kind* of zip file following very particular specifications—the EPUB specifications.

whatever form they prefer (text, video, audio) on whatever device they happen to be holding (laptop, e-reader, tablet, smartphone), whenever they want it, and wherever they happen to be. Overnight, our information landscape seems to have turned into another Pandora: James Cameron's bewildering *Avatar* landscape.

Most shocking of all, this is not a futuristic fever-dream of a bunch of techies who've had too much Mountain Dew. This is not a movie. This is real. We really *can* start reading a book on a laptop in the office and pick up where we left off on a phone on our way home. We really *can* click on the name of a baseball player in a magazine on a tablet and see his stats pop up or watch him hit that pennant-winning homer. We really *can* see and hear Dr. Martin Luther King's "I Have a Dream" speech in an encyclopedia. We really *can* do a self-test in a textbook and get an instant grade (and receive guided learning based on how well we did).

We even take these things for granted now. That's because the Web already enables all those things. The good news: the Web lets us do an amazing number of really cool things. The bad news: the Web lets us do them in lots of different ways, using lots of different technologies. Many of which are proprietary. Many of which are incompatible with each other.

Why isn't this more of a problem on the Web? There are two main reasons. First, we have a choice of browsers, and they're free and frequently updated, so if the browser we're using isn't able to handle something, we can often find another one that can. Second, "online" is a two-way street: the sender of the content can detect important information about the capabilities of the recipient, and can adjust what it's sending and how it's sending it. EPUBs can't always do that. Fundamentally, EPUB is a "package" masquerading as a file format, designed to be accessible and fully functional *offline*.

It's often a revelation to find that an EPUB file is really a zip file: if you change the .epub extension to .zip, you can see all the goodies inside—metadata, XML files, images, and so forth. (Go ahead, try it.) But it's not *just* a zip file, it's a particular *kind* of zip file following very particular specifications—the EPUB specifications. Specifications for how the text content is marked up in XML. Specifications for what metadata must be there, and how it must be expressed. Specifications for what other file types can be included. Specifications for the file types that conformant reading systems must be able to handle. Specifications for how all the pieces are organized—and how to *say* how the pieces are organized. (And lots more.) The Web doesn't have to do any of that.

Although these specifications may seem to be making things unduly complicated, they actually make things much easier. Easier for the publisher of the EPUB, who—faced with our Pandora's swarm of choices and options—has a clear path to consistency. Easier, too, for the maker of the reading systems that need to receive and render the EPUBs: instead of needing to be able to accommodate that multitude of options—or, worse, *not* to accommodate some of them—the system can know what it will get in an EPUB, how it can find the pieces it needs, and what it needs to do with them. All of which, ideally, makes it easier for the consumer as well.

First, a word about our sponsor

You may have noted that I've been careful to use the word "specification," not "standard," when referring to EPUB. The reason is important. It is, of course, intended to provide a standard way in which to interchange and deliver reflowable content to reading systems. ("Reading systems" is also a carefully chosen term.



EPUB 3 is designed to accommodate textbooks, scholarly and STM monographs, and technical manuals as well as non-book content like magazines, newspapers, journals, white papers, and corporate documents: anything anybody would find it useful to package as an EPUB and interchange or deliver through any of the rapidly proliferating choices of reading systems.

EPUBs aren't just for handheld e-reading devices; they're for laptops and desktop computers and text-to-speech reading systems for the print disabled—any e-reading environment, including ones not invented yet—as well.) For all practical purposes, it's a standard, just as the current EPUB 2.0.1 specification is. While it is not yet an International Standard from a de jure standards body like NISO, it is a standard similar to those from the W3C, the World Wide Web Consortium, the keeper of the XML family of standards and most standards fundamental to the Web.

The IDPF, the International Digital Publishing Forum—the body that maintains EPUB—is a not-for-profit trade association of over 200 members from over twenty countries that represents a broad cross-section of the publishing ecosystem: publishers, technology companies, device manufacturers, and others, commercial and non-commercial, with a shared interest in fostering a robust and well functioning e-publishing environment. Its EPUB 3 working group is an extraordinarily large one—170-some members, with additional invited experts—representing that full spectrum.

It was a veritable “Peaceable Kingdom” of competitors collaborating: publishers of all types, from some of the largest commercial and nonprofit trade, scientific, scholarly, and educational publishers to the smallest; makers of e-book devices (both Apple and Sony were very active members); e-book distributors and retailers; technologists from companies like Adobe and Google to small consultancies and individual programmers; people from standards organizations like DAISY, NISO, EDItEUR, BISG, and IDEAlliance (though none *formally* representing those organizations, I should make clear); and many others, such as librarians, service providers, and other interested parties. It was also genuinely international, with particularly active and valuable participation from Japan and elsewhere in East Asia.

These folks obviously didn't always agree on everything, but they did agree on the group's fundamental mission, and they worked conscientiously (and *hard*) to come up with the best possible result. All of this work was done in a totally transparent fashion, on an open listserv and a publicly accessible wiki, with a strict avoidance of proprietary issues and intellectual property claims. There were clear mandates for what was to be accomplished and a formal process for accomplishing it. The result is an open standard, based as much as possible on open standards, that addresses real-world needs in realistic ways.

It's called “EPUB” not “EBOOK”

One of the most important goals for EPUB 3 was to accommodate a much broader range of content than EPUB 2.0.1 did. The IDPF originated mainly in trade publishing, and the earlier generations of EPUB reflected that. In contrast, EPUB 3 is designed to accommodate textbooks, scholarly and STM monographs, and technical manuals as well as non-book content like magazines, newspapers, journals, white papers, and corporate documents: anything anybody would find it useful to package as an EPUB and interchange or deliver through any of the rapidly proliferating choices of reading systems.

EPUB 3 was also designed to accommodate a much broader range of *types* of content. No longer primarily for simple text-and-image content, it now provides a practical solution for incorporating multimedia content like audio and video, as well as animations and other scripted functionality. In keeping with its international mission, it provides global language support. It provides much more sophisticated typographic and layout capabilities (this is especially important to magazine and textbook publishers). It accommodates much more extensive metadata, at all levels, from the package to the paragraph. And in keeping with one

CONTINUED »

of its most important mandates, it is designed to enable and facilitate conformance with standards for accessibility. (These aspects of EPUB 3 will all be discussed in more detail below.)

Creating a specification out of squishy standards

A dilemma facing the EPUB 3 Working Group from the outset was that some of the standards that were obvious candidates for inclusion in the EPUB 3 spec are not yet fully and formally final. (The technology folks call them “squishy” and “not fully baked,” in case you want to know the true technical terminology.) The most important of these were HTML5 and CSS3.

I mentioned that EPUB 3 was created in response to our rapidly changing e-publishing environment. I should state that more strongly: EPUB 3 was created in a mad dash to get ahead of the rapidly changing technological developments before it was too late to make a difference. To put it bluntly: *people are already doing all this stuff*. And although one of the problems EPUB 3 was created to address is the chaotic assortment of technologies and techniques being used today, the reality is that there are already some clear best practices. It would be foolhardy for EPUB 3 to try to force people in directions in which they clearly don't want to go, or which are recognized as bad ways to go, or for which it is already too late.

So, for example, while it prompted a bit of discussion (these working groups are, after all, the descendants of college debating societies and dorm room arguments), the decision on font formats was really pretty clear cut: OpenType and WOFF. OpenType is clearly the right font format from the print publishing world, from which lots of EPUB content comes; WOFF is clearly the dominant font format in the Web publishing world; and you can't easily make one conform to the other. Ergo, EPUBs have to use either OpenType or WOFF fonts, and reading systems have to support both. End of discussion.

The core standards at issue in this regard were HTML5 and CSS3. These are extremely extensive and fundamental standards. HTML is the language of the Web and CSS (Cascading Style Sheets) is the way the Web is mainly rendered (or should be). Their latest incarnations, HTML5 and CSS3, can be thought of as the next generation of the Web, taking shape in front of our very eyes. People already use them to do cool things with typography and layout, interactivity, animations, rich media. Browsers already implement them. Yet they are still both works in process. While they are both in a sense “modular,” important modules are nowhere close to being finalized. HTML5 is not expected to be a formal Recommendation of the W3C until 2014. There will be millions of EPUBs created between now

and 2014. EPUB 3 can't wait. So the dilemma boils down to this: EPUB 3 really *must* be based on HTML5 and CSS3, and yet they're not really finished standards.

The resolution, I think, was eminently reasonable. The EPUB 3 Working Group elected to selectively specify those modules of HTML5 and CSS3 that either (1) are in fact considered finished, for all practical purposes, or (2) are essential to an EPUB 3 requirement and are close enough to resolution that they are reasonably safe to use. Moreover, the EPUB 3 spec attaches a “warning label” to the latter: if at some point the HTML5 or CSS3 spec changes from what EPUB 3 is specifying, EPUB 3 makes the commitment to change along with it, so that EPUB 3 will stay in synch with HTML5 and CSS3. This approach is realistic, practical, and not as risky as it appears to be. For example, EPUB 3's use of CSS is really still almost entirely based on the existing CSS 2.1 specification; it just brings in certain modules from CSS 3 that are needed to accomplish certain things that CSS 2.1 does not address.

Markup and metadata

Although the interest in EPUB 3 is understandably focused on all the new capabilities it offers, it is important to understand that it is fully backwards compatible with EPUB 2.0.1. That means that all EPUB 3 conformant reading systems *must* render EPUB 2.0.1 publications properly. It also means that most of the new aspects of EPUB 3 are optional.

One very important change concerns the text markup vocabulary. EPUB 2.0.1 provided two vocabularies: XHTML (which was used by the vast majority of EPUBs) and DTBook, a vocabulary published by the DAISY Consortium for accessibility purposes. DTBook has been eliminated from the EPUB spec because the DAISY Consortium decided to work with the IDPF to enable EPUB 3 to become the distribution format for accessible content, rather than requiring a separate model as they had before. Thus DTBook is being phased out as a delivery format by both DAISY and EPUB, leaving the basic markup vocabulary for textual content in EPUB as XHTML—the vocabulary used by the overwhelming majority of EPUBs created so far. XHTML5 (the XML expression of HTML5) as used in EPUB 3 provides *additional* vocabulary features but does not change the basic XHTML vocabulary used by the previous spec.

There has been one change that's particularly important to STM publishers: MathML, the standard for representing mathematics in XML, is now a “first class citizen” in EPUB 3. MathML provides both semantic and presentational markup, the former concerned with what math expressions mean and the latter with what they look like. It is only the *presentational* aspects of MathML that EPUB 3 reading systems are required to support.

The metadata capabilities in EPUB 3 are also dramatically expanded from EPUB 2.0.1. While there are still only three required elements—`dc:identifier`, `dc:language`, and `dc:title`—EPUB 3 enables publishers to include many more identifiers, versions of titles (for example a title for sorting purposes or a short title), and many more elements of metadata, along with specifying the scheme that defines them (e.g., MARC 21, ONIX 3.0, PRISM).

There is a very basic set of standard metadata terms “built in” to EPUB 3 that can be used without a prefix; all other metadata requires a “profile” to be declared and a prefix to be used on those metadata elements. This enables content creators from many different interest groups with specific metadata practices to incorporate their own metadata vocabularies within an EPUB. (It’s a quasi-namespace approach, but it does not require resolution to a metadata authority. However, it does require *identification* of the appropriate metadata authority.)

In addition, EPUB 3 enables content creators to associate metadata with EPUBs at all levels. Previously, it was only possible to associate metadata at the package level. Now, much expanded metadata can be associated with the package as a whole, with a component of the package, or even embedded right in the content markup itself, down to the paragraph level. (Phrase-level metadata can also be done.) Two of the special metadata attributes specified by EPUB 3 are `epub:type`, for adding semantic information to markup, and `epub:trigger`, to launch a multimedia or scripted function.

Finally, it is also possible to provide what is referred to as “external” metadata. This can either be a file of metadata included within the EPUB package (for example, a MARC record or an ONIX file) or pointed to via a link (which of course will only work in an online environment).

All of the metadata in EPUB 3 is expressed in very standard, widely used ways. Most is based on Dublin Core (with a preference for DCTERMS); other metadata is added using very simple features taken from RDFa 1.1. The goal was to provide a metadata mechanism that would be extremely easy to implement, even for nontechnical content providers, while accommodating the rich metadata that is becoming an ever more essential part of the information ecosystem.

Taking advantage of our new real estate

It’s a little ironic how much influence the emergence of tablets has had on the e-publishing landscape. Although there are many things to love about tablets—they are unquestionably already an indispensable component of our ecosystem—what seems at first to be their salient feature, the amount of real estate they offer in which to render content, has been there all along in desktop and laptop computers. (Of course it’s how that feature plays with all their other features—including their portability, their gesture-based interface, and their ability to be both “Web” and “not-Web”—that makes all the difference.)

It’s important to realize that what is most important about tablets from the EPUB point of view is that they are one mode among many in which to render EPUB content. They exist as a component of an ecosystem that does still include laptop and desktop computers, along with handheld reading devices and smartphones. Unlike PDF, which is a fixed-page format that locks in everything about the page (in fact, it is the stability of that format across media, from print to online, that is PDF’s greatest virtue), EPUB is all about *reflowable content*.

EPUB 3 provides the capability to design rich layouts like those common to magazines and textbooks—such as multiple columns (with hyphenation) whose



“Fixed Page” Layout

Through a function called “media queries” (the EPUB basically asks “where am I?”), different style sheets can be used to produce, for example, a two-page spread on a tablet held in landscape mode, a one-page two-column layout when that tablet is turned to portrait mode, and a single column format on a mobile phone, all from the same XHTML5 file. This enables a type of “fixed page” layout—control of the content page-by-page—while still enabling reflow.

CONTINUED »

CORE MEDIA TYPES



EPUB CONTENT DOCUMENTS

XHTML
SVG

FONTS

OpenType
WOFF

GRAPHICS

.jpg .gif
.png .svg

AUDIO

MP3
MP4 AAC LC

EPUB 3 specifies certain file types as Core Media Types. Reading systems must be able to render these properly, if they offer the functionality provided by a given Core Media Type. If a content provider uses a file that is not a Core Media Type, it must provide a fallback to a file that is a Core Media Type. EPUB 3 Core Media Types include those shown above.

CONTINUED »

text flows around images and sidebars—while enabling the design to adapt to the real estate available to it. Through a function called “media queries” (the EPUB basically asks “where am I?”), different style sheets can be used to produce, for example, a two-page spread on a tablet held in landscape mode, a one-page two-column layout when that tablet is turned to portrait mode, and a single column format on a mobile phone, all from the same XHTML5 file. This enables a type of “fixed page” layout—control of the content page-by-page—while still enabling reflow. This will be the most important feature of EPUB 3 for many publishers.

This is only one example of the dramatically improved capability for control of graphic design offered by EPUB 3. It not only permits embedded fonts, it encourages them (and provides for what is called “font obfuscation” to prevent font piracy). This is important not only for publishers who want to maintain branding or a certain “look and feel” for their publications, but also for publishers of specialized content like technical or linguistic content, which requires special “glyphs” that are unavailable on standard fonts.

Layout issues are of particular concern to Asian publishers or others who use non-Latin alphabets. Few people realize that EPUB 2.0.1 permitted right-to-left text reading (required for languages like Hebrew and Arabic) because reading systems didn’t implement this capability. EPUB 3 goes much farther, allowing vertical writing as well. In addition, SVG (Scalable Vector Graphics) is a Core Media Type (see figure above) in EPUB 3. Think of SVG as PDF expressed as XML: images are captured as vector graphics that adapt to the size and resolution of the rendering environment. This means that even publications like manga and graphic novels can be delivered as EPUB. And the

EPUB 3 spec enables the publisher to specify reading order, as well, so that a book can be read from right to left and the first page of a spread can be understood to be on the right.

Rich media and scripting

The topic of rich media and scripting provides me an opportunity to address the “app” issue. The success of the iPad and the proliferation of competing tablets, along with the parallel proliferation of the iPhone and other smartphones—especially those created for the Android operating system—have made it possible for publishers to provide content that includes audio and video content as well as scripted behaviors ranging from simple animations to elaborate interactive functionality. Before EPUB 3, these were done through apps, the small single purpose applications that these technologies have made so popular. However, apps are an impractical way for most publishers to publish most of their content. They are specific to an operating system (an app for iOS won’t run on an Android device), they require programming, and they usually prove to be too expensive and time consuming to create for all but the most popular or high-priced products.

EPUB 3 enables all this to be done in a standard way that is device agnostic. Any reading system that is EPUB 3 conformant and which offers the ability to play video, audio, and scripts will properly render a conforming EPUB 3 publication. This is a huge benefit to content creators. No longer do they need to create different versions, with different specs and even different file formats, for different environments. Yet, in cases where it does still make sense to create an app—and there will be many such cases—doing so

is that much easier when it is done on the basis of an EPUB 3 in the first place. EPUB 3 helps answer the questions like: “Which format should my audio files be in?” and “Which scripting language should I use?” In an EPUB, an audio file will always be either MP3 or MP4 AAC LC (the latter because it’s required by the required MPEG 4 video format); if there is scripting, it will always be done using JavaScript. This makes things so much easier for content providers and reading systems alike!

Accessibility

EPUB 3 was created from the outset to address issues of accessibility, and experts from the DAISY Consortium have been instrumental in the development of the spec. An important benefit of this close collaboration is that in its update of DAISY (which is a NISO standard), the Consortium expects to specify EPUB 3 as the delivery format for DAISY. (See related article on page 35.) This means that in addition to using EPUB 3 to make files that will work on a host of reading systems, publishers can also use the same EPUB 3 files to deliver their content accessibly.

In EPUB 3, the navigational structure is specified not by the former proprietary format, but in an XHTML microformat. The XHTML5 markup of EPUB 3 content (with the addition of a new attribute, `epub:type`) can accommodate all the semantics necessary for accessibility. A new feature of EPUB, media overlays, enables the synchronization of text to audio, enabling print-disabled users to use the XHTML file to search and navigate its audio counterpart. And there are important text-to-speech features in the EPUB 3 spec, including PLS (the Pronunciation Lexicon Specification) and fine-grained pronunciation control via SSML (Speech Synthesis Markup Language).

Rapid EPUB 3 adoption is expected

Because EPUB 3 provides much needed clarity to our currently chaotic e-publishing environment, its adoption is expected to be swift. As soon as it is formally introduced, it is expected to be endorsed by major information industry organizations and adopted by major technology companies. There will be EPUB 3 reading systems available commercially before the end of 2011, and EPUB 3 is expected to be in wide use by 2012.

In the meantime, as soon as the specification is formally published, IDPF plans to publish extensive documentation, examples, and best practices to make it easy for publishers to incorporate EPUB 3 into their workflows, along with a validation mechanism to help ensure that EPUBs conform

properly to the specification.

While it’s clear that more work will need to be done to continue to advance the EPUB specification (which the IDPF plans to do in a modular fashion, rather than issuing future “monolithic” releases), the EPUB 3.0 spec is a major watershed. It will be the foundation on which our e-publishing ecosystem will be based for many years to come.

| FE | doi: 10.3789/isqv23n2.2011.02



BILL KASDORF -bkasdorf@apexcovantage.com is Vice President of Apex Content Solutions and General Editor of *The Columbia Guide to Digital Publishing*. He is a member of the IDPF’s EPUB 3 Working Group (and leads its Metadata Subgroup); BISG’s Content Structure Working Group (and chairs its Standards Survey Subgroup); and the IDEAlliance nextPub Working Group (chairing its EPUB-to-nextPub Mapping Committee). He is Past President of the Society for Scholarly Publishing and is a frequent speaker for publishing industry organizations.

CSS3

www.w3.org/TR/css3-text

Dublin Core Metadata Element Set

dublincore.org/documents/dces/

Dublin Core Metadata Terms [DCTERMS]

dublincore.org/documents/dcterm-terms/

EPUB 3

dpf.org/epub/30

HTML5

www.w3.org/TR/html5/

International Digital Publishing Forum

www.idpf.org/

MathML

www.w3.org/TR/MathML3/

MPEG-4

mpeg.chiariglione.org/standards/mpeg-4/mpeg-4.htm

OpenType

www.microsoft.com/typography/otspec/

www.adobe.com/type/opentype/

Pronunciation Lexicon Specification (PLS)

www.w3.org/TR/pronunciation-lexicon/

RDFa 1.1 Primer

www.w3.org/TR/rdfa-primer/

Speech Synthesis Markup Language (SSML)

www.w3.org/TR/speech-synthesis/

Web Open Font Format (WOFF)

www.w3.org/TR/WOFF



RELEVANT
LINKS