Format

4

File formats are an essential part of digital media consumption. A software package requires instructions for how to present and interpret the data compiled in a media file. Despite their significance, formats often receive less attention than other aspects of media and platforms. Jonathan Sterne argues that format is too often conflated with medium, which only reveals itself during moments of competition: Betamax/VHS, Blu-ray / HD DVD, MiniDisc/MP3. Format encapsulates "a whole range of decisions that affect the look, feel, experience, and workings of a medium. It also names a set of rules according to which a technology can operate."[1] Within the confines of contemporary computing, the file extension is the most prominent signifier of format and demonstrates the fluidity between software and format. A word processor document can have multiple extensions (.doc, .docx, .odt), which in turn can be opened by various software packages (Microsoft Word, OpenOffice, Google Docs). Despite this fluidity, each of the formats has different material limits and structures.

Ebooks require an understanding of digital format theory *and* the substantial history and conventions of print bibliography, as a clear continuity exists between the two. In the study of the transmission of printed texts, the format refers to a work's wrapping, which signifies a book's role. For example, cloth-bound academic publications are a marker of prestige designed primarily for libraries, while the subsequent paperback reflects the economic or cultural success of the initial publication. Formats are ingrained in book history: from Aldus Manutius's development

of a Renaissance "paperback" to the distinctions between folio, quarto, octavo, and other bindings.[2] These formats referred to the binding and were an important indicator of a title's prestige: a cheap edition featured more sheets of paper stitched together in a single gathering, while the most important titles comprised many single folded sheets of paper stitched together into a large volume called a folio.

The digitization of the book has reshaped our understanding of format. Matthew Kirschenbaum argues that, rather than existing as a single finished object, "a book is an assemblage of digital assets, consisting, in practice, of multiple files and formats collected in a digital asset management system, or DAM. These files contain artwork, fonts, style sheets, metadata, and of course the text. A book is thus also a network, since these digital assets must be orchestrated to interact with one another in structured and predictable ways in order to generate desired outputs, such as an EPUB or an XHTML file."[3] Even so-called plain text, displayed without any formatting attributes through a command line or editor interface, is mediated through format by character maps and font families. Character maps translate hexadecimal numbers into the letter or symbol to display. The "map" in the name is literal: specifications include a grid with coordinates to locate specific characters. For example, the ASCII character map represents a capital *A* with the hexadecimal number 41 (column 4, row 1), and its lowercase equivalent is represented by 61. ASCII does not dictate how a character should be displayed, which is handled by font rendering systems such as TrueType and PostScript that take the number and convert it into a recognizable shape. Users can then choose a font family to display their text according to individual preferences. Text display therefore requires several processes before a single character is rendered as a series of pixels on a screen (table 4.1).[4]

Table 4.1 Layers of text display

Layer	Format	Output
Style sheet	h1 {color: gray;}	a
Markup	<h1>	**a**
Bitmap	a	a
Font renderer	Times New Roman	[instructions for drawing a lowercase *a* in Times New Roman]
Hexadecimal character map	0 x 61 (column 6, row 1)	a (lowercase)
Binary	110001	110001

Markup languages such as HTML enable further presentational differences. Software packages interpret extra text surrounding content to be displayed to present headings, bold text, and other visual differences. For example, in HTML, <h1> is the tag for displaying a heading with the highest priority (designated by the number, which notes a hierarchy down to <h6>, the smallest subheading). CSS determines how headings are displayed through additional markup: "h1 {color: gray;}." In this instance, the heading will be in gray, but further markup may include font size and alignment. Markup only offers recommendations, however, since readers may choose to increase the font size or change the color to suit their own preferences. While HTML and CSS are powerful tools for tinkering with presentation, the additional text can greatly increase the size of an uncompressed ebook.

Developing the Kindle Format

In *Memory Practices in the Sciences*, Geoffrey Bowker compares the Earth to a vast geological archive; similarly, the Kindle is a palimpsest that reveals earlier iterations if you just scratch below the surface.[5] This approach to formats ensures consistency but also makes it difficult to change things. As Daniel Pargman and Jacob Palme argue, "The most common solution is to patch things and hobble along. Any deviation from that solution requires potentially large alterations of the existing information infrastructure, and such alterations are *expensive*."[6] As a result, examining the origins of the Kindle file format is essential to understanding the platform's current limitations.

Amazon designed the Kindle during a period of relative stability within the ebook industry. In the early 2000s, an industry group initially called Open Ebook, spearheaded by Dick Brass of Microsoft, had agreed on a set of standards.[7] Over forty publishers, booksellers, and technology companies collaborated on the Open eBook Publication Structure (OEBPS), the precursor to EPUB, an interchange specification designed to allow publishers to easily convert files to various proprietary formats.[8] No reading system interpreted OEBPS directly, so technology companies could create proprietary formats derived from the specification. For example, Sony developed BroadBand eBook (BBEB) for its first-generation Librie in 2004, before committing to EPUB with the launch of the PRS-505 hardware in 2007. Likewise, Amazon's acquisition of Mobipocket, discussed in chapter 1, included the popular MOBI format, another derivative of OEBPS. The reliance on Mobipocket technology two decades later brings significant archaeological baggage to the platform. Amazon's commitment

to backward compatibility (ensuring new titles are still readable on older devices) while ensuring new devices have enough flagship improvements *and* ensuring broad cross-hardware support places a substantial strain on format specifications.

Amazon's cumulative approach to format created a series of geological layers visible within each ebook. Each successive upgrade to the file format moved further away from the original design but still accounted for its genesis as Palm Resource Code (PRC), a database language designed for Palm OS in 1996. Amazon's commitment to backward compatibility ties the company to this lineage to avoid obsolescence update cycles that have defined Apple's business model during the 2010s. Apple's yearly upgrade cycle primes customers for depreciated apps with major operating system updates, but the realities for text-oriented software are different. Without flagship hardware improvements or significant security vulnerabilities, it is more difficult to justify breaking access to content for users who continue to use older devices. Since static text appears to be simple to render, any changes will be met with greater resistance.

The first Palm hardware was released in 1996, the same year the XML 1.0 specification was published. Rather than adapting an emerging open standard, Palm built PRC and Palm Database (PDB) for text reading in the popular C programming language. These formats have little formal published documentation but provide the foundations for the contemporary Kindle ebook. The technical limitations of Palm OS determined how PRC files were processed on derivative devices.[9] The Palm Pilot 1000, one of the first Palm devices, released in 1996, handled an upper limit of twelve megabytes of random-access memory (RAM) and four megabytes of read-only memory (ROM). As a result, all documents were precompiled before transfer to Palm devices to reduce storage and memory costs. This asymmetrical structure was attractive to publishers hoping to protect intellectual property, and Kindle formats never deviated from this model. Until the development of KF8, all new Kindle file formats were renamed MOBI files, and devices would still process PRC ebooks. Amazon's file formats are based on archaic standards from 1996 rather than EPUB3, which derives from modern web specifications such as HTML5 and CSS3. Several of the Kindle's quirks emerge from this historical divergence and Amazon's later attempts to reconcile the expectations of users for contemporary ebook design. Newer file formats integrate features from EPUB3, but the geological layers of the original format remain. PDA users who wanted to read ebooks in the 1990s had different needs from ebook readers in 2007. PDAs featured small, low-resolution screens, so reflowable text, a central part of "ebookness," was more user-friendly than attempting to create print facsimiles.

MOBI in turn was built on Memoware, an amateur ebook production community launched in 1996 after the release of the first Palm Pilot. Once established, the community changed focus to create an early ebook store.[10] Memoware's name links to Palm's "Memo" software, as users wished to create ebooks for their Palm Pilots before reading systems were available for Palm OS. Early adopters focused on creating computer manuals for Memoware, which influenced the design of the MOBI format specification, which includes basic linking and interactive features required for the genre. The initial transition from MOBI to AZW was straightforward. For security reasons, Amazon's development team removed the ability to use JavaScript, extended the DRM protection, and changed the file format's name from MOBI to AZW. This early rebranding allowed publishers working outside of Amazon's service infrastructure to provide DRM-free Kindle-ready titles. If readers change an ebook's extension from MOBI to AZW, reading systems will interpret the file identically. While newer formats deviate from this model, newer devices still read legacy formats.

Amazon's choice of MOBI in 2005 was pragmatic. EPUB was not established as a consumer-facing format, and all proprietary formats derived from the best practices of OEBPS, HTML, and CSS. EPUB developed traction simultaneously to the Kindle, and without the momentum from the Kindle's high-profile launch, it is likely that EPUB would never have seen widespread adoption. Google Books focused on facsimiles; Apple would not launch iBooks until the reveal of the iPad in 2010. Other than Sony, the ebook market depended on a diffuse network of bookstores and had not gained critical mass since substantial investment in hardware earlier in the decade.

Since 2007, Amazon has developed several format specifications to meet the conflicting needs of dedicated hardware and third-party software, summarized in table 4.2. These sociomaterial constraints result in several contested aspects of the Kindle format's perceived pitfalls. Criticisms are often based on typographic issues that remain challenging for other ebook file formats.[11] If we move beyond these common ailments, the format ecosystem contains the geological layers that can be peeled back to reveal a broader history of displaying text, from precompiled databases to cloud-assembled files.

No public documentation for Kindle formats is available, but Amazon provides "best practice" advice for publishers preparing HTML or EPUB content for conversion to a Kindle ebook.[12] Instead, to understand the Kindle's formats, I reverse engineered available source code and DRM-free publications. Publishers can use a command line tool, KindleGen, or a graphical user interface wrapper, Kindle Previewer. These tools offer an

Table 4.2 Summary of Kindle file formats

File extension	Year released	Features
.prc	1996	Palm Resource Code. Database files containing on-screen reading documents.
.mobi	2000	An ebook format based on PRC and developed to display HTML-like attributes.
.azw	2007	The original Kindle format, based on MOBI. Speculated acronym: Amazon Word.
.tpz	2008	Topaz. The primary format for Kindle files downloaded for iOS devices.
.azw3/.kf8	2011	A transformation of AZW to bring it more in line with EPUB standards. Launched alongside the Fire tablet range.
.azw4	2014	A wrapper for PDFs.
.kfx	2015	Completely rewrote the format logic and rules. Marketing through "enhanced typography."

approximation of what the final output will look like on a range of devices to avoid any errors that can occur during the complex conversion process. Self-published authors can use Kindle Create, which converts Microsoft Word documents into Kindle files, alleviating the requirement to understand markup languages to produce a basic ebook.[13] Amazon's documentation shows disregard for publisher autonomy through comments such as "publishers do not need to define a start reading location because Amazon does this during the upload process."[14] Amazon asserts control over the Kindle ecosystem by keeping the internal workings of its file formats secret and stripping unwanted HTML tags from the conversion process. This approach ensures Amazon's dominance by creating an environment resembling Jonathan Zittrain's metaphorical use of "walled gardens" to describe "sterile appliances tethered to a network of control."[15]

On the other hand, due to the World Wide Web Consortium's dedication to open standards, EPUB is thoroughly documented. The industry standard forms the basis for many Kindle files at the production stage, but its cloud-based conversion process reshapes these data. Kirschenbaum warns that "programmatic computational environments [apply] some particular logic—a certain formal materiality—to the string of bits in question."[16] The differences in the "formal materiality" of Kindle files and EPUB stem from divergent ideologies in the production and consumption of digital content. Both formats have the same building blocks

derived from open web standards including HTML, CSS, and XML. The standards depart at the point of processing the raw materials. The most important differences are compression and data storage. While EPUBs use ZIP compression, which maintains the original file structure, Kindle files are delivered in binary format, which obfuscates the structure of the original file. This shift ensures Kindle formats are more difficult to preserve. Amazon changed its data-storage practices with the introduction of KFX, which changed the underlying format from XML, the standard for EPUB, to JSON. For our purposes here, the two markup languages are nearly identical beyond syntax differences. Let's take a name as an example:

XML: <person>
<firstName>John</firstName>
<lastName>Smith</lastName>
<person>

JSON: {"person" : {
"firstName" : "John",
"lastName" : "Smith"}}

Beyond minor technical and syntactical differences, the choice between the two standards comes down to developer preference and what systems the content needs to interact with. Nonetheless, Amazon's choice to move away from XML, the industry standard, introduces an additional conversion process that can increase corruptions while also reducing publisher autonomy.

The shift away from XML-based standards was part of a larger move toward a broader typology of ebooks. Amazon introduced multimedia ebooks with the launch of the Fire tablet range, leading to the development of KF8, an extension of AZW that incorporated elements of EPUB3 rather than resurrecting older aspects of the MOBI specification. While many of these new web technologies feature in the Mobipocket specification, Amazon intentionally restricted them when launching the initial Kindle to avoid the possibility of malicious JavaScript or SQL injections and to suit the restrictions of an electronic paper screen. The new format allowed for the development of new genres, including digital comics, that were not covered in previous iterations of AZW. KF8 introduced "app-amzn-magnify," a JavaScript library that allows publishers to locate areas to magnify on images rather than relying on the pinch-and-drag metaphors common on smartphones and tablets. These incremental improvements focused on balancing the needs of new genres with ensuring the security of the format.

KF8 enabled the publication of PDF-like content without simulating the flexibility of print. For example, in Doug Dorst and J. J. Abrams's *S.*,

napkins and bookmarks appear nestled in predetermined pages in the print version. In the Kindle edition, the fixed layout determines where annotations and ephemera appear, but the actual ephemera have to be represented as photocopies with a note that the page is a "[found object]."[17] The Kindle file format explicitly forbids more extensive interaction, although the iBooks version allows users to move the found objects across the page they are initially displayed on.[18] This deferral to a facsimile-level fidelity was only possible on higher-resolution screens, and both smartphones and dedicated ebook readers lack the screen capacity to display the content accurately. Nonetheless, this was an intentional decision that negatively shapes the reader's experience. Dedicated e-readers that launched before 2011 are unable to open KF8 titles, marking the first substantial break in backward compatibility. The format is largely relegated to displaying fixed-layout and interactive titles, and other ebooks that use the format strip any unnecessary elements to ensure a version of the text can be displayed on all dedicated hardware.

The launch of KFX in 2015 marked a break from the conventions of both EPUB and PRC and allowed the Lab126 Application Framework team to explore alternate layout engines, leading to the design of XYML.[19] As the quaint name suggests, the markup language is based on XML and describes the x- and y-coordinates of text and images. The Application Framework team designed XYML as a lightweight alternative to CSS.[20] Using this proprietary format, Amazon can strictly control how content appears on the platform. Electronic paper presents unique challenges for rendering content, since a single screen must load simultaneously when the user refreshes the page, whereas a web page using HTML and CSS can afford to load from top to bottom. The solution? Encourage a nonlinear approach to layout rendering. XYML breaks from the Western tradition of left-to-right, top-down presentation in favor of a nonlinear approach according to the needs of specific publications.[21] As a result, the Kindle can now render more complex comic book layouts and render more languages, including Arabic and Japanese. Rather than using CSS, this proprietary solution obfuscates page layouts, ensuring Amazon's control over the sort of content available on its Kindle Store.

Amazon also introduced KFX to correct some long-standing issues with formatting through what was termed "enhanced typography." The new rendering engine built on "hyphen," a Linux package that replicated the digital typography pioneer Donald Knuth's TeX engine's implementation of automated hyphenation, where a set of rules determines if and where line breaks should be introduced in justified formatting. Amazon marketed the revised layout engine as a revolution in formatting, but as Guido Henkel

has argued, it has often detracted from the primary content.[22] He criticizes KFX for glossing over some of the long-running issues in the Kindle formatting process in place of superficial aesthetic improvements. Henkel attacks the Kindle's inability to offer transparent images in either JPEG or PNG, which are both accepted Kindle image formats. Instead Henkel notes that "Amazon is now converting all images in an eBook into JPG XR format, including all GIF and PNG images, as well as SVGs."[23] Converting images to a single format ensures consistency in delivery even if doing so alters the images. JPG XR is not a lossless format and can introduce new visual artifacts into images.[24] Henkel's focus on transparent images is misguided compared to the semantic and character mapping issues embedded in the format discussed in the "Accessibility and Internationalization" section. In fact, transparent images would run counter to Amazon's deference to user choice, as it is possible in some Kindle configurations to alter the background color, rendering some transparent images illegible. Amazon attempt to navigate the fine balance between using more flexible standards and ensuring the platform runs efficiently across user configurations.

The use of Cairo, a graphics-rendering package, reveals how Kindle engineers have navigated issues around displaying images, as the source code reveals that the Lab126 engineers removed Scalable Vector Graphics (SVG) compatibility from the library despite dedicated devices having the capability to display SVG.[25] The company emphasizes conservative aesthetic principles to avoid unforeseen bug exploitation. For example, in the Kindle format ecosystem, it is impossible to layer objects, as demonstrated by shifting the ephemera in *S.* across pages. SVG allows designers to draw objects on top of other text or images, and the oversight required to manage this feature is undesirable for Amazon, which restricted SVG's use. Recent guidelines have relaxed the ban on SVG with the introduction of enhanced typography, but publishers can only use a limited set of tags.[26] The design conservatism puts a clear focus on the primary object—most commonly the text, but also images in comic books—and ensures that content can be rendered on any Kindle hardware.

Autonomy and Control

The introduction of KFX allowed Amazon to assert its authority over the rendering of ebooks, counter to the more open approach taken by other platforms. KFX stores ebooks in a company-designed version of JSON called "Ion" designed in 2009 and deployed across Amazon's services. The company emphasizes readability in the format specifications: "Because most data is read more often than it is written, Ion defines a *read-optimized*

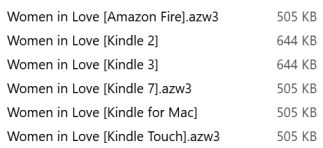
binary format."[27] Beyond the obfuscation potential of the specification, dictating a transition from EPUB to JSON allows Amazon to control features embedded in the content available via the Kindle. Jiminy Panoz, an ebook designer, notes that Amazon processes EPUBs through a command line simulation of a web browser, also known as a headless browser, to check the HTML before rendering the content in Ion. Preprocessing offers Amazon control over unwanted features, including "removing drop caps or floats when the user reaches a font-size ceiling," but at the expense of the autonomy of both the consumer and producer.[28] The inherent tensions between publisher and Amazon control within the Kindle platform ensure that the reader cannot expect the affordances of contemporary reading platforms without the full palette of open web standards.

Ebook formats are designed to meet the needs of publishers and technology companies rather than users, resulting in criticisms of the presentational aspects of the format. The ability to compress data to protect proprietary elements of both the main text and the algorithmically generated aspects of an ebook was vital for publishers' comfort in adopting open standards. The World Wide Web Consortium (W3C) designed open web standards such as HTML and CSS to work in the broadest possible context, while the book trade had specific intellectual property and metadata requirements. For example, ebooks need to contain identifiable information about authors or editors, and most titles are linked to ISBNs. When users visit web pages, they download a copy and can "view source" to see how they work; publishers wanted to protect their content through use of digital rights management (DRM). The Kindle DRM system is based on eBookBase, a collaboration between Mobipocket and Franklin Electronic Publishers in the early 2000s to create a cross-platform DRM database for ebooks.[29] Copyright reform advocates such as Cory Doctorow lobby against DRM as anticonsumer, but the technology offers conservative publishers reassurances about the unauthorized transmission of their intellectual property.[30] Ebook formats err on the side of publisher control over user autonomy. Formats thus extend beyond logical representations to consolidating power within relationships. While reassuring publishers hesitant about digital distribution, the Kindle format's black box positions Amazon as dominant in the hierarchy. As Kirschenbaum notes, this problem extends beyond the Kindle to much of contemporary publishing, where "no publisher has as much oversight of their workflow as they might like. Much happens through contractors and third parties."[31]

The mixture of Amazon's DRM, the Kindle file ecosystem, and the demands of different operating systems has a clear effect on generating unique ebooks. A user's hardware determines the format type of an ebook

(see table 4.2). This can be as minor as a different file extension for identical files, as in the case of Touch and Kindle for Mac ebooks, which have different extension names (figure 4.1). Greater differences occur at the forensic level, where each file header includes a DRM payload to determine if an ebook is loaded to the authorized hardware device. Every Kindle download is identified by a unique number generated with reference to the ebook's ASIN and the reading system software package as part of its social DRM protection (that is, embedding the ability to identify where a pirated file originated). If a user downloads the same book twice, each copy will contain a unique identifying number. Amazon uses this number to authenticate an ebook with an associated hardware serial number to stop users from sharing content.[32] The resulting encrypted text differs between copies, but the files remain the same size.

Beyond DRM and strict metadata requirements, ebook specifications are flexible and open for interpretation by the reading system, which in turn depends on the hardware and software available to the end user. This affords the user limited autonomy over aspects of the presentation within the publisher's and technology company's constraints. Formats offer recommendations rather than dictate rules to enable user customization through fonts and layout. Reading systems are inconsistent in their support for features, so the format cannot mandate elements unavailable for certain software without breaking backward compatibility. For example, due to the interpretive differences between platforms—an Android mobile phone does not have the same screen real estate as Kindle for Mac displayed on a forty-inch screen—the same content will not be displayed consistently across the Kindle platform. Dedicated hardware displays have variable requirements: the original Kindle had a six-inch screen, the DX featured a nine-inch screen, and the recent Oasis range settled on seven inches. The format must be rigorous to provide a standardized "Kindle aesthetic" but also allow for ample variation between devices and reading preferences. Despite the introduction of comics and fixed-layout and media-rich ebooks, the "text-heavy" reflowable ebook remains the most common format. The genre of Kindle content is optimized for the



4.1 Logical similarities between copies of D. H. Lawrence's *Women in Love*. Screenshot by the author.

limitations of dedicated e-readers rather than designed to showcase the more complex rendering capabilities of desktops and smartphones.

The technical details and choices made about the format specification have an impact on the structure and typography of specific texts. In Walter Isaacson's *Steve Jobs*, the Kindle's ability to mark the entry point of the text as separate from the beginning of the main body means that the ebook initially opens on the "Characters" page rather than the table of contents.[33] Brian Abel Ragen acknowledged this structure with regard to Junot Díaz's *The Brief Wondrous History of Oscar Wao*.[34] Both titles feature preliminary materials that shape interpretation, so this decision alters the reading experience. The Kindle Publishing guidelines reveal that the process is algorithmic rather than editorial, which leads to inconsistencies. The algorithm does not open the book at page 1 by default. While Isaacson's biography of Jobs opens before page 1, *Oscar Wao* starts with the main body, ignoring the front matter. A 2016 patent filing by Sravan Babu Bodapati and Venkatraman Kalyanapasupathy, two members of the Machine Learning team, captures how the algorithm worked that year. First, titles were scanned for keywords in titles deemed either relevant ("chapter 1," "prologue," "introduction") or irrelevant ("copyright," "acknowledgments"). If this process fails to produce a clear cutoff point, the algorithm compares a "bag of words" from any introductory matter and compares its similarity to the main text with the assumption that any "relevant" text will bear similarity to the main body. When the algorithm fails to determine an appropriate Start Reading Location (SRL), a "manual review operator" from the publisher makes the decision.[35]

Accessibility and Internationalization

Pargman and Palme introduced the concept of "ASCII imperialism" to discuss internet and platform governance's Anglocentrism, highlighted by the dominance of ASCII across computational systems.[36] The Kindle's file format contributes to this phenomenon despite its origins. Thierry Brethes and Nathalie Ting designed Mobipocket to be used internationally. The MOBI specification offered publishers a choice of two character maps: UTF-8, which features over one million unique characters from global languages; or Windows-1252, designed by Microsoft, emphasizing the Latin alphabet with a limit of 256 characters.[37] To extend the map metaphor, UTF-8 offers a detailed map of the globe, while Windows-1252 documents a province. Publishers could choose between the limited palette of Windows-1252 to decrease the file size or use the richer character set of UTF-8 to publish non-English language content. When revis-

ing the MOBI specification for the Kindle, Amazon engineers restricted access to Windows-1252, reducing support for non-English languages and nontraditional typography.

Publishers worked around this early constraint by inserting images of characters not available in Windows-1252. Figure 4.2 shows a JPEG used to display ō (014d in the Latin Extended-A Unicode character map) in Walter Isaacson's *Steve Jobs*. The solution is incompatible with the core tenet of reflowability, as the image does not scale with the font sizes, leading to unreadable text. The workaround also affects the algorithmic reading paratext, including text-to-speech, definitions, and Word Runner, which cannot parse the images as part of the word, rendering the text as "S [obj] t [obj]." The white background draws the reader's attention to the character's absence. The restriction ironically mirrors the Unicode Consortium's insistence that text without a visual representation in a text-rendering engine should instead appear as an empty box (□) to signify an absence. Newer hardware offers support for Unicode, which could fix this error, but older devices without UTF-8 support cannot display the text correctly. Publishers would need to create multiple unique versions of the same text to be optimized for the different devices. Since this would only benefit Amazon, it is difficult to justify the expense.

Formatting errors extend beyond the integration of characters incompatible with Windows-1252. As with the use of images in place of text, these issues that appear to be minor inconveniences can affect the broader accessibility of the ebook. Isaacson opens his book with a dramatis personae, where Little, Brown decided to format individuals' names with small capitalization (the first letter of each word is larger than the remaining characters, although all letters are in uppercase). Publishers achieve this effect through tagging, but the implementation is inconsistent across ebook platforms. For example, in a simplified version of ebook formatting, "H<small>ELLO</small>" would be rendered as "Hello." Small capital letters are parsed separately from full capitalization when KindleGen converts EPUB to AZW, which was only fixed with the launch of "enhanced typography" with the Voyage. As a result, "Kobun Chino" is presented in text as "KOBUN CHINO," a minor typographic difference, but the <small> tags are interpreted as spaces, so the software reads the

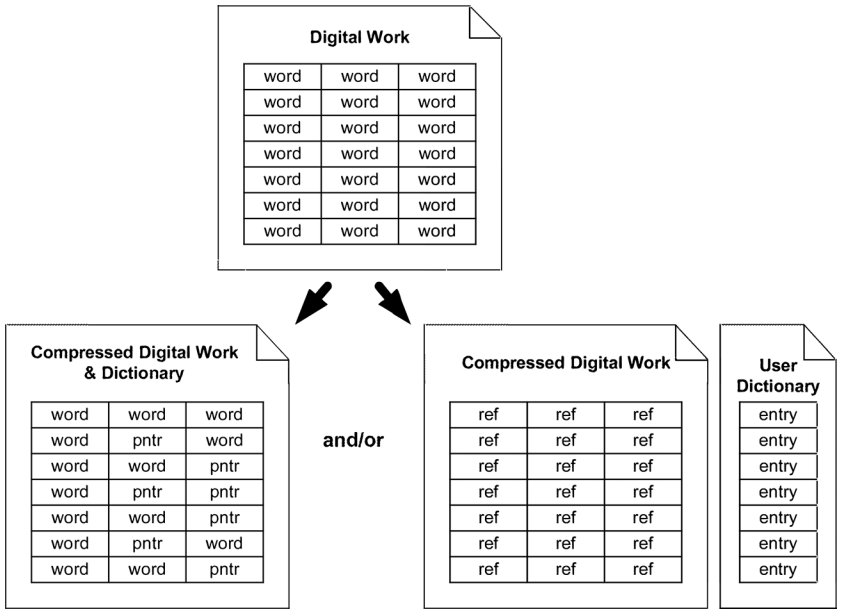
KOBUN CHINO. A Sōtō Zen master

4.2 Detail of JPEG for diacritics in Walter Isaacson's *Steve Jobs: The Exclusive Biography*. Source: Walter Isaacson, *Steve Jobs: The Exclusive Biography* [Kindle for iPad 4.10], AZW3, B005J3IEZQ (Little, Brown Book Group, 2011), loc. 200.

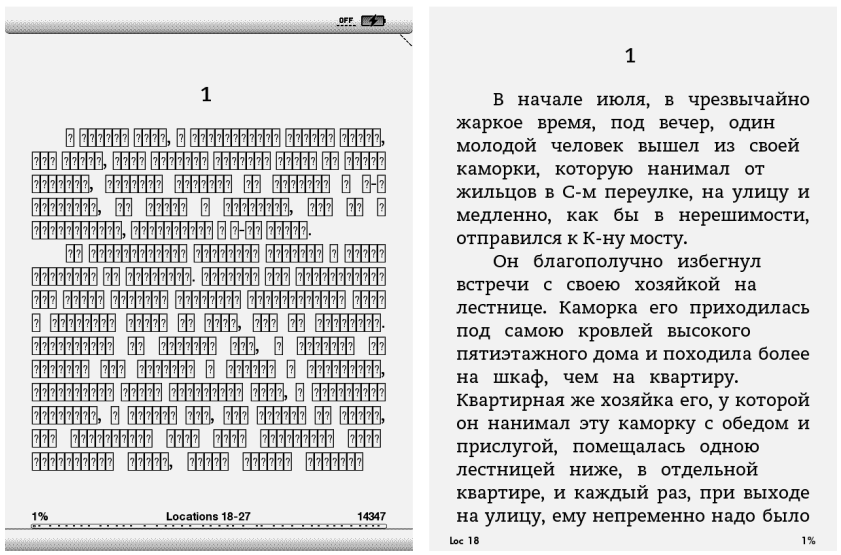
text as "K OBUN C HINO," which affects the quality of its text-to-speech and Word Runner conversion. Choosing Windows-1252 by default and stripping tags without context may create a consistent presentation, but the changes to accessibility and semantic access can adversely affect readability.

Amazon engineers have developed various technological solutions to what remains a social problem. For example, Anubhav Kushwaha, a software development manager at Amazon, experimented with compressing ebooks by using an index dictionary. Each unique word received an identifying number in a dictionary that is then recalled in text rather than repeatedly spelling out words (figure 4.3).[38] The method has a historical precedent in early ebooks implementations developed for the US Navy to increase the efficiency and speed of reading reference manuals.[39] Pertinent strings can also be compressed into dictionary entries, reducing commonly used markup such as <h1> for headings or navigational elements to a numerical string far more compact than the original markup. The technique was initially designed to circumvent the expense of storage memory in the 1980s, but Amazon recast the technique for obfuscation of content in the constant arms race with users who were reverse engineering the Kindle DRM to transfer ebooks to other platforms.

Kushwaha's work mirrored earlier research into character-level dictionary indexing. Amazon headhunted Eric Menninga, a typography veteran of nineteen years for Adobe, to work on improving the layout of Kindle ebooks and "solve the problem of authoring complex documents for display on devices with varying capabilities."[40] Menninga worked with Lokesh Joshi, another typography expert, to expand the Kindle character map to be compatible with UTF-8 while restricting access to exploits within the system.[41] Building on Kushwaha's work, Menninga and Joshi focused on generating indexed dictionaries of characters in an ebook rather than words, allowing for a customized character map for each title. Each ebook features a compressed version of UTF-8 rather than offering access to the complete character map, with a customized map that includes only the characters present in a particular title.[42] A new character map with just an index of characters present in the text further obfuscates the content and adds an extra layer of security to the DRM. Just like Amazon's approach to converting EPUB to AZW through JSON, the character map conversion introduces an additional risk for creating unintended errors that can be difficult to troubleshoot given the obfuscation process. As figure 4.4 shows, this represents clear progress for better internationalization support, but the approach is constrained overall by Amazon's extensive use of proprietary solutions based on available open technologies.



4.3 The user dictionary approach to compressing ebooks



4.4 Comparison of how a Kindle 2 (left) and Kindle Touch (right) render the same MOBI file of Фёдор Достоевский (Fyodor Dostoyevsky), *Преступление и наказание* (*Crime and Punishment*). Source: Федор Михайлович Достоевский, *Преступление и наказание* (Russian Edition) [Kindle Touch 5.3.7.3], AZW3, B00KHRXSMI (Общественное достояние, 2014), loc. 31.

This approach ensures that a reader can access the same ebook title on a Kindle 1 or Voyage and it will be optimized for that hardware, not to mention the array of secondary platforms Amazon supports. The hardware and software will also create fluctuations in the naming conventions of a file. For example, the same title will be branded "AZW" on a Kindle 3 but appear as "PRC" on a non-Amazon-branded Android device. These minor deviations appear through the longitudinal development of platform-specific workflows that lead to different results even if the output is nearly identical. The formatting of several editions of Walter Isaacson's *Steve Jobs* biography shows a consistency between file types, but also format differences in variants. The ebook is not a single file, however, as several services are sideloaded through separate files (table 4.3).

Steve Jobs is a technically simple ebook with occasional images, but otherwise no complex formatting. Nonetheless, the biography is made up of many distinct files: outside of the main ebook, we find over thirty different files sideloaded with metadata (table 4.3). Many of these files are optional services available only on certain devices, but in several instances, the Kindle offers a unique interpretation of formatting choices owing to its archaic roots. The available sideloaded content can be grouped into four primary categories:

1. *Bibliographic metadata* about the length of the book, the author, and other aspects often found on the opening pages of a printed book
2. *User-generated metadata* pertaining to reading time and the collation of popular highlights
3. *Generative paratext* created through algorithms packaged with the ebook to interact with other elements of the Kindle platform, including page numbers, indexing, reading speed, and vocabulary building
4. *Metrics and logs*: local files compiling information about the user to be sent back to Amazon

The ad hoc assembly of files server-side has clear consequences for the platform. While users have the flexibility to view the same title across a spectrum of hardware and software configurations, with a range of additional features, the structure of the text as a material object is permeable and prone to error. Each download is unique and offers a narrative of both its production and consumption through metadata traces. The platform focuses on scale and variety, which can adversely affect the quality of products that have not been optimized for the Kindle.

Table 4.3 Folder structure for various active Kindle versions of Walter Isaacson's *Steve Jobs*

File	Description	Touch	Kindle 8	Android	Fire	Mac
AZW	Base ebook file in most advanced format available for each device					✓
AZW3		✓				
KFX			✓	✓	✓	
SDR	Folder containing relevant metadata	✓	✓			
AuthorProfile.ASC	Popup boxes in JSON about the author, and to appear at the beginning and end of the ebook	✓				
StartActions.ASC				✓	✓	
EndActions.ASC		✓	✓	✓	✓	
AZW3F	Encrypted reading metrics data	✓				
TICR				✓	✓	
YJF			✓			
MBP	JSON reading metrics data					✓
AZW3R	Annotations / page number data	✓				
APNX	Page number algorithm					✓
PHL	Popular highlights XML	✓	✓	✓	✓	✓
MF	A log of server requests		✓			
LanguageLayer.kll	SQL database for Word Wise, Amazon's vocabulary builder		✓			
XRAY.db	SQL database containing the index information for the X-Ray system			✓	✓	
XRAY.asc			✓			
amazn1.drm-voucher.ast	DRM information			✓	✓	
kfx.luci	A series of 11 encrypted metadata files				✓	

Kindle ebooks are created on Amazon Web Services to optimize titles to work with new features while remaining backward compatible with older devices. Over the Kindle's first decade, the company introduced new formats while maintaining support for older devices. The resulting infrastructure supports an ecosystem of formats rather than a single file format that has been developed alongside hardware and software innovations in an incremental fashion. The Kindle file format is not a stable object until a user downloads the file for a specific hardware-software combination. The results are closer to the letterpress era, where print runs could be varied according to their production rather than a seamless and standardized process. Since 2012, Amazon has processed a stack of "source ebook data" including "text," "graphics," and "font data" through a server to produce "modified ebook data" personalized for individual users.[43] The shift from storing compiled ebooks on a server to generating personalized copies on the fly offers extra DRM protection through identifying specific copies circulating in the wild while also allowing publishers to provide a single "master copy" that is then converted to suit specific hardware configurations by stripping out or adding content where appropriate.

The complexities of the Kindle's formatting process are difficult to reconcile within the usual confines of format, as the Kindle exhibits a greater fluidity emblematic of Amazon's algorithmic control of the process. No archetypal Kindle file format exists, and even though newer formats derive from older specifications and remain stable based on the hardware-software configuration of a particular user, the notion of a prototypical Kindle format is difficult to formalize in practice. It is closer to the complex multiserver assemblage of Google Docs than the more straightforward single-file PDF. The combination of sideloaded paratext and vastly different file formats for the primary content, as well as the generation of data and content algorithmically through first and subsequent use, makes the Kindle file format difficult to pin down as a single conceptual unit. Jiminy Panoz describes the equivalent EPUB phenomenon as a spectrum, since reading systems are not required or able to interpret all aspects of a common format.[44] This courtesy also extends to developers to adapt the standards to best suit their platforms. The common principles that emerge across format iterations are more important than the differences, as this is what defines the contemporary ebook.

Amazon is unlikely to switch to the status quo of EPUB. The decision to implement JSON on top of keeping elements of PRC allows the company to maintain control over the features available to the title and protect publishers' intellectual property. An asymmetrical process allows the company to only include white-listed elements rather than experience

unwanted behavior. As a consequence, the most accurate representation of a Kindle file exists not in publishers' archives or on readers' devices but in the databases containing JSON to be converted to a title when requested by a user. From the perspective of creators, it is impossible to ensure that the carefully designed ebook will be preserved through the upload *and* download processes according to the latest, or historic, software demands. Users—and researchers—will not necessarily have an identical experience on any two devices, and finding consistency is difficult.

Through Amazon's continual expansion of the platform, and its long history embedded in older computing traditions, the format has remained one of the most complex parts of the Kindle infrastructure. Amazon's commitment to backward compatibility while simultaneously attempting to diversify into tablets and encourage a small number of readers to upgrade on a yearly obsolescence cycle has led to a complicated ecosystem of file formats that are distinct yet similar. As the Kindle platform has grown, quality control has become more difficult to maintain, leading to differences in the display of titles between a Kindle 1 and iPhone X. Simultaneously, each new iteration of the Kindle file format has introduced new proprietary and divergent rules and logics, ensuring that Amazon maintains control over what is allowed on the platform. While this initially came at the cost of accessibility and internationalization, newer devices have struck the balance between control and access.

Four Shades of Gray

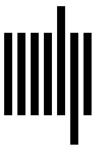
The Amazon Kindle Platform

By: Simon Peter Rowberry

Citation:

Four Shades of Gray: The Amazon Kindle Platform
By: Simon Peter Rowberry
DOI: 10.7551/mitpress/11985.001.0001
ISBN (electronic): 9780262369114
Publisher: The MIT Press
Published: 2022

The open access edition of this book was made possible by generous funding and support from MIT Press Direct to Open



The MIT Press

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in Filosofia OT by Jen Jackowitz. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Names: Rowberry, Simon Peter, author.
Title: Four shades of gray : the Amazon kindle platform / Simon Peter Rowberry.
Description: Cambridge, Massachusetts : The MIT Press, [2022] | Series: Platform studies | Includes bibliographical references and index.
Identifiers: LCCN 2021013279 | ISBN 9780262543507 (paperback)
Subjects: LCSH: Kindle (Electronic book reader) | Electronic book readers. | Electronic books.
Classification: LCC Z286.E43 R689 2022 | DDC 004.1675—dc23
LC record available at https://lccn.loc.gov/2021013279

10 9 8 7 6 5 4 3 2 1