



American Council of Learned Societies

HISTORY E-BOOK PROJECT

ACLS History E-Book Project Report on Technology Development and Production Workflow for XML Encoded E-Books

ACLS History E-Book Project White Paper No. 1

by Nancy Lin
Electronic Publishing Specialist
ACLS History E-Book Project
<http://www.historyebook.org>

ACLS Contact: Nancy Lin
E-Mail: nlin@acsls.org
Date: October 3, 2003
Version: v. 1.0
White Paper URL: HTML: <http://www.historyebook.org/heb-whitepaper-1.html>
PDF: <http://www.historyebook.org/heb-whitepaper-1.pdf>

1. INTRODUCTION.....	4
The ACLS History E-Book Project.....	4
Purpose of Report.....	4
Sample Files	5
2. PROJECT GOALS AND XML FORMAT	5
Beyond Straight Digitization of the Print Book.....	5
A Cost-Efficient and Sustainable System.....	5
Cross-Collection Functionality	5
Why XML?	5
Other Formats Considered	6
3. BOOK ELEMENTS AND FUNCTIONS	7
Identifying Elements and Functions	7
Building Blocks for Creating HEB E-Books	7
Additional Features: Related Titles and Book Reviews.....	9
4. BOOK STRUCTURE AND ELEMENT IDS.....	10
Structural Divisions	10
“Chunking” Text.....	10
Paragraph Numbering	11
Placement and Numbering of Added Elements.....	11
5. TECHNOLOGY DEVELOPMENT, SCHOLARLY PUBLISHING OFFICE (SPO)....	12
Working with SPO	12
Developing the XML Publication System	12
Resources from DLPS	12
6. SPECIFICATIONS AND DTD DEVELOPMENT	13
Developing the DTD.....	13
Granularity in Tagging.....	14

Multiple DTDs	14
Metadata in MARC Format	15
Special Character Encoding	15
7. DESIGN AND STYLESHEETS.....	15
Designing and Styling E-Book Pages	15
HTML Limitations.....	16
Styling with XSLT.....	16
8. PRODUCTION WORKFLOW	17
“Print-First” to “Born-Digital” Books.....	17
Outline of Production Workflow	17
Quality-Control Procedures	19
9. XML TAGGING OPTIONS AND COSTS.....	20
Options for XML File Conversion and Tagging	20
Conversion Costs	21
10. CONCLUSION	22
Accomplishments	22
Upcoming Development.....	22
FIGURES	24
Figure 1. Frontlist E-Book Title Record Page with Table of Contents.....	24
Figure 2. Frontlist E-Book Page	25
Figure 3. Frontlist E-Book Page with Images.....	26
Figure 4. Frontlist E-Book Search Results Page	27
Figure 5. Frontlist E-Book Page with Search Terms Highlighted in Red	28

1. Introduction

The ACLS History E-Book Project

The ACLS History E-Book Project (HEB) is an electronic publishing initiative whose goals are to assist scholars in the electronic publication of high-quality works in history, to explore the intellectual possibilities of new technologies, and to help assure the continued viability of history writing in today's changing publishing environment. It is a cooperative venture among the American Council of Learned Societies (ACLS), eight scholarly societies, and ten university presses to publish, in electronic format, 85 new books by historians ("frontlist") and 500 previously published books of major importance to historical studies (retrospective conversion or "backlist").

In September 2002 the project was launched as a subscription product with a collection of over 500 backlist titles in the field of history. Each year, an additional 250 backlist titles will be added to the collection. As of September 2003, 6 new frontlist titles are published on the live site and another 20 titles are in various stages of production. Over the next few years, the 85 frontlist titles will range from "print-first" to completely new, "born-digital" titles.

The e-books in the collection are accessible to students and scholars through subscribing libraries and learned societies. The project now has over 180 [subscribing institutions](#), with new subscribers being added weekly.

This project is funded by a \$3 million grant from The Andrew W. Mellon Foundation, plus a smaller grant from the Gladys Krieble Delmas Foundation. The technology back-end for the project is provided by the Scholarly Publishing Office (SPO) at the University of Michigan Library.

Complete information about the project can be found at <http://www.historyebook.org>.

Purpose of Report

The purpose of this report is to document and disseminate information about the technical development and production processes for HEB frontlist titles. With these titles, we are experimenting with new technologies, functionality, and publication workflows. We hope sharing information about our efforts will be useful to the community of scholars, publishers, librarians, and others interested in the development of e-books in the humanities.

In this report, we cover five topics relating to the technical production of HEB frontlist titles. First, we discuss our project goals and reasons for using XML technologies. Second, we describe HEB e-book features and functionality, highlighting the technical solutions used to implement the features. Third, we discuss our work with the University of Michigan's Scholarly Publishing Office (SPO) in developing the back-end technology. Fourth, we cover our development of specifications for XML tagging and design and styling of HEB e-books. Finally, we outline our production workflow as well as options and costs for XML tagging. Throughout the report we talk about the rationale behind our decisions and discuss some of the challenges we encountered.

Sample Files

HEB's specifications, documentation, XML DTD, scripts, and sample files are available for download at <<http://www.historyebook.org/xml/doc/acls-hebook-doc.html>>. Publishers and their vendors use these materials to produce e-books for the HEB project. For examples of frontlist e-book features, see <<http://www.historyebook.org/frontlistfeatures.html>>.

2. Project Goals and XML Format

Beyond Straight Digitization of the Print Book

The project's goal with the frontlist titles is to push beyond simple digitization of the print book. HEB wants to promote the creation of high-quality works that take better advantage of the electronic medium, incorporating such features as hyperlinking, pop-up windows, images, sound and additional resources, in innovative ways. We have seen that when an author conceives a book knowing that there will be electronic publication, the resulting work will include creative use of electronic features. For example, these books might incorporate a large number of color images, links to external websites, comparison of text with primary source material, or they might be structured in a way difficult to realize in the print format.

A Cost-Efficient and Sustainable System

While HEB aims to produce innovative, feature filled e-books, the project also wants to make sure that development and maintenance is cost-efficient and sustainable. We are not developing stand-alone, interactive multimedia works requiring extensive programming, nor are we publishing large, discrete websites with ever-changing content. Instead, our books are structured texts, accessed through standard web browsers, and include elements such as dynamic pop-up boxes, linking, color images, interactive maps, juxtaposition of text with source or related materials, audio and video clips, etc. Updates will follow typical book publishing cycles for revised and new editions. Not only will this approach help to maintain a cost-efficient system, it also insures common standards for the structure, distribution, and review of these titles by historians who write, evaluate, and use them and the presses who publish them.

Cross-Collection Functionality

For the HEB project, one key to developing a robust, but sustainable, system is to develop features that can be used by multiple books across the collection. In other words, we avoid developing book-specific functionality. Thus, if a book requires a new function that is not yet available (e.g., a pop-up image slide show), we decide if it is a useful function to add to our core set, and, if so, we develop it as a new feature available for general use. This strategy of controlled development supports author innovation but at the same time promotes sustainability.

Why XML?

In order to fulfill HEB project goals, we need an efficient, flexible publication system that gives us power to easily program new features. With this need for extensibility, along with our need to distribute to a broad user base, it was not too difficult to conclude that our solutions should be based on XML technologies, even if they require higher costs of development. So we chose to encode all frontlist titles in XML and dynamically transform them into HTML for delivery via standard web browsers. Here are some key reasons why we chose this XML solution:

1. Flexible, Modular Structure:
HEB can easily construct books in a flexible, modular way, by piecing together different elements like text, images, sound, and invoking functions like hyperlinks and note pop-ups. XML allows us to create our own set of tags so we can mark up specific elements in the e-book and style and manipulate them as we choose.
2. Cross-Collection Development and Maintenance:
For more efficient development and maintenance, books encoded in XML can be processed as a collection. Designs can be managed through templates, and new functions can be programmed for books across the collection.
3. Styling with XSLT:
XSLT (Extensible Stylesheet Language Transformation) is a powerful tool for transforming XML into other documents such as styled HTML pages. It is an open, non-proprietary standard that can be used to format style based on XML elements and attributes. Although HEB does not yet use XSLT for styling on the live site, we use it heavily for designing and proofing e-books during production.
4. Robust Search:
With XML, HEB can develop more robust search mechanisms, such as searching within specific text elements (e.g., chapters, paragraphs), returning results with header details and highlighted search terms in context. (See [Figure 4](#) and [Figure 5](#))
5. Broad Web-Based Delivery:
By encoding in XML and transforming to HTML for web delivery, HEB can provide broad access without requiring users to have special software or hardware beyond a web browser and desktop computer.
6. Open Standards, Non-Proprietary:
Since XML is an open standard, HEB is not tied to a specific proprietary software format. We have open access to our text for editing using different software programs and platforms as well as for programming new features.
7. Long-Term Access:
HEB wants to ensure that these high-quality scholarly works are available in the long term. With XML encoded files we can more easily migrate to other formats as technology changes. We are also consulting with the Library of Congress on depositing archive copies of our XML and XSLT files.

Other Formats Considered

For the frontlist books, HEB also considered other types of e-book formats. These formats fall into two general categories: 1) e-books with pages based on the print-page layout, or 2) e-books tagged in HTML or XHTML, the standard mark-up languages for web pages. However, in our review, we encountered the following limitations:

1. PDF and Page-Image Formats:

E-Book formats that deliver pages based on the print-page layout include such formats as Adobe PDF files read in Adobe Acrobat or Adobe eBook Reader software, and web pages with images of scanned book pages. These formats have certain advantages. They can reproduce complex page layouts and can be made cheaply and accurately by converting directly from print-book pages. For our backlist books, using scanned page images is a good low-cost solution since our priority for the backlist collection is to put up a large number of cross-searchable books in a cost-effective and timely manner, delivered as exact reproductions of the print edition. However, for our frontlist titles, PDF files and scanned page images would be far too static and limiting. These formats employ a linear page-by-page structure, are slower and less easy to navigate, and cannot be easily programmed for new interactive functionality.

2. HTML and Open E-Book Formats:

Tagging our e-books in HTML or XHTML would provide more power to manipulate text and program interactivity than the page-image formats. HEB did consider the option of either tagging in basic HTML or using the Open E-Book standard, which is encoded in XHTML. However, for our purposes, the major disadvantages of HTML and XHTML are that the set of tags and attributes is finite and tagging is based more on presentation than content. With XML, we can build our own set of tags based on content (e.g., tagging foot/endnotes as notes rather than just a plain set of paragraphs) and set certain rules, such as “all chapters must have title heads.” This extensibility allows us to build such features as pop-up notes, and to style text across a collection. Hence, the lack of ability to tag text by content and assign attributes prevented us from using HTML or XHTML as our fundamental format.

3. Book Elements and Functions

Identifying Elements and Functions

HEB’s first task in setting up the system for the frontlist books was to identify a core set of elements and functions that the system needed to be able to handle. We reviewed about a dozen books proposed for the project and also analyzed some of our participating publishers’ in-house styles to establish a set of core book elements. Then we determined a set of basic functions, such as pop-up notes, text linking, image enlargement, linked indexes, etc.

Building Blocks for Creating HEB E-Books

HEB is now working with the following list of elements and functions that publishers can use as building blocks for constructing e-books for our project. We will continue to add elements and functionality as needed. (See [Figures 1-3](#) for sample e-book pages.)

Elements of Content

1. Basic Text Elements:

This includes text elements typically used in scholarly works: headers, paragraphs, extracts, epigraphs, line groups, lists, tables, salutation/signature elements, foot/endnotes, indexes, etc. HEB specifications currently include around 30 element tags.

2. Supplemental Text:

Supplemental text including original source material, transcriptions, additional commentary, translations, etc., can be added. For example, one author included the original-language transcriptions of the hand-written French manuscript, and another author included additional commentary on the comparison of two versions of a graphic print series. Supplemental text can be placed at the end of the text in appendices, inserted within the text, or placed in pop-up text boxes.

3. Images:

Images, in full color or black and white, can be inserted within a book in different ways, for example:

- a. In-line within text
- b. Separate section, as in plates
- c. Thumbnail index
- d. Comparison table

4. Sound / Video Clips / Interactive Images:

Various types of multimedia elements, such as sound and video clips and interactive or moving images, can be included. Since these elements often require users to have software beyond a basic web browser installed (e.g., Flash, RealPlayer), HEB reviews specifications for these elements on a case-by-case basis.

Functional Features

1. Pop-up Windows for Notes and Other Text:

Footnotes, endnotes, or other text can appear in pop-up windows. In the HEB system footnotes and endnotes appear by default in pop-up windows. When a user clicks on a linked note reference number within the main text, a small window pops up with the text of the note. The entire notes section can also be accessed from a link in the note pop-up window or directly through the table of contents. Authors can also include other types of text in pop-ups, including translations or other commentary.

2. Linking:

Several types of linking can be used, including:

a. Internal Linking to Text Elements:

Links can be made from any text location to specific elements within the e-book. For example, links can be made from anywhere in the text to a paragraph, table, list, or any section. Index entries can be linked to elements in the text. Publishers can also use links to build many useful scholarly tools, such as concordance charts comparing multiple versions of a text or direct links to translations or transcriptions.

b. Internal Linking to Figures:

Figure listings in a list of illustrations can be linked directly to figures within the text. References to a figure with the main text (e.g., "See Figure 5") can be linked to the exact location of the figure elsewhere in the text.

- c. External Linking to HEB Books:
Links can be made from the text to other books in the HEB collection.
 - d. External Linking to Other Websites:
Links can be made to websites, journal articles, reviews, scholarly resources, archives, databases, etc.
 - e. External Linking to Image Resources:
Links can be made to images residing at other website locations. For a few of our titles, publishers provide smaller versions of images for the HEB site and then link out to larger, higher resolution images that reside on other websites. Many institutions maintain image collections on their local servers for public access, so sometimes it makes more sense for us to link to these resources rather than reproduce them on our site. In some cases, this assists publishers in keeping permissions paperwork and costs in check. Also, for some books, linking to external image archives is preferable because some sites offer more robust image viewing tools (e.g., InSight) than what is currently offered on the HEB site.
3. Image Enlargement:
Publishers can set an image within the e-book to be enlarged in one of two ways:
- a. JPEG Pop-Up Window:
When a user clicks on an image in the text, a larger version of the image in JPEG format appears in a pop-up window. We recommend that publishers use this method for basic images.
 - b. Image Viewer Tool:
The image viewer tool is available for high-quality images, such as color art, line drawings, or detailed maps. When a user clicks on an image in the text that is tagged for the image viewer tool, a pop-up window appears with options to zoom in and pan on a large image. This tool works in a basic browser window and does not require additional software. Publishers submit high-resolution TIFF files to HEB, and these images are converted into SID files, an image compression format that facilitates the viewing of large image files via the web.

Additional Features: Related Titles and Book Reviews

Along with this set of core elements and functions, there are other powerful features that enhance HEB frontlist books, including:

1. Related Titles, Backlist:
Authors are requested to provide 6 to 12 titles of related historiography, books that have framed the conceptual and factual background of their own works. HEB tries to acquire rights to put them on-line as part of the backlist collection. The related titles are listed in a pop-up window accessed from the frontlist book's title record page; backlist titles successfully acquired and added to the HEB collection are linked in this list.

2. Book Reviews:

For each book in the HEB collection, links to reviews in JSTOR, Project Muse, and the History Cooperative are provided if available. Since many institutions in our customer base have subscriptions to these resources, this is a very useful and popular feature.

4. Book Structure and Element IDs

Structural Divisions

After establishing the basic list of book elements and functions, HEB needed to work through some questions about handling book structure and navigation. We knew we would receive books of different structural types, ranging from traditional monographs with chapters/sections to works based on a series of color images or letters and commentary. Our system has to be capable of handling different types of books, but with one set of processes.

To accomplish this, we set up the HEB XML DTD (Document Type Definition), which defines rules for tagging e-books, to break down an e-book by hierarchical divisions that can be assigned to any type of structural unit. Each e-book is made up of divisions that can be assigned to different types of units, such as section, chapter, part, letter, image group, etc. A book made up of chapters and sections, for example, would be tagged with the first division level, <DIV1>, for chapters and the next division level, <DIV2>, nested inside for sections. A book made up of a series of letters might be tagged with <DIV1> for a group of letters and <DIV2> for each letter inside that group. This flexible division structure is based on the Text Encoding Initiative's TEI Lite DTD, from which our DTD is derived.

“Chunking” Text

The next challenge was deciding how a text should be broken down for viewing. What size “chunk” should be delivered to a user—a page, a chapter, or a section at a time?

Since most scholarly e-books are straight conversions of the print edition, they are often delivered either 1) a page at a time or 2) a chapter or section at a time. For our e-books, it did not make sense to chunk a book by print-page since neither text written for the on-line version nor titles “born-digital” would have print-page breaks. Furthermore, the browser page can be of variable length. We therefore decided to abandon print-page constraints in favor of a more flexible—but still citable—system.

It made more sense to break down HEB e-books by logical units, such as by chapter, section, or other unit. However, upon analyzing our publisher's needs, we saw that we could not simply apply one chunking rule for all frontlist books across the board. We could not set our system to *always* chunk by chapter or section, or, by division level. Our publishers need a flexible system that permits them to:

1. Chunk by different division levels within a book:

For example, for a book with chapters and sections as well as a separate group of letters, the publisher might want to deliver the main text a section at a time, but the letters as a group. Publishers can choose to chunk by <DIV2> level (sections) at one point in the book, and at another point chunk by <DIV1> level (group of letters).

2. Chunk by different division levels across collection:

For one book, the publisher might want to chunk by section, while for another book, it might want to chunk by chapter or other unit.

To create this flexible system, we set up our DTD so that publishers can select which division of text they want to show by setting an attribute in the division tag. The default breakdown for a book is by the top division level, <DIV1>. If a publisher wants to break down a text by section at the second division level, <DIV2>, it would simply add status="hidden" to the <DIV1> tag. The following XML tagging would instruct our system to deliver to the user text chunked by sections rather than by full chapters:

```
<DIV1 type="chapter" status="hidden">
  <DIV2 type="section">
```

Note that some HEB books, especially our earlier “print-first” titles, are made up of only chapters with no further breakdowns. We provide the option to deliver these books in chunks smaller than a full chapter to facilitate royalties accounting and viewing. HEB pays royalties through a formula based on “page hits.” A “page” in our backlist books is one book page, and a “page” in the frontlist books is a divisional “chunk” of text. So breaking down a frontlist book into smaller units creates more uniformly measurable page hits for royalty reporting across the collection. The other factor considered is that if text is broken down it is more difficult to make unauthorized downloads of an entire book. To address these concerns, we allow publishers to break down chapter text into smaller chunks, e.g., by every ten paragraphs.

Paragraph Numbering

Since some HEB titles have new text added or are born-digital, print-page breaks are not always available for reference and citation. Thus, a new numbering system for scholarly reference and citation was required. We decided to add to a growing consensus that included the on-line version of the *American Historical Review* and other journals on the History Cooperative site, and use paragraph numbers to serve as permanent text reference IDs. For all of our texts, each paragraph is assigned a permanent number that appears in gray to the left of the paragraph text. The range of paragraph numbers for each text chunk appears after each table of contents entry so that one can quickly locate the appropriate text from a paragraph number reference. For titles that have print counterparts, the print-book page number appears in brackets (e.g., [Page 24]) within the text and thus can be used by those who have a print-page reference. (See [Figures 1-3](#))

Placement and Numbering of Added Elements

Another structural issue HEB encountered for many books was how to best handle placement and numbering of elements that exist only in the e-book version. Thus far, for all of our frontlist titles, publishers are also producing print editions. Many of the on-line versions include elements that are not in the print edition. Often, more images are included on-line, and publishers must decide how to number the additional images relative to the print version and whether they should appear within the text or in a separate section. If additional text is being included, the publisher must decide where to put it—within the main text, in a pop-up note, or in an appendix. With the print-first editions, most publishers and authors opt for keeping the numbering schemes used in the print edition. To do this, some publishers used somewhat awkward numbering systems in the e-book version, such as adding Figures 5a, 5b, 5c for new figures between Figures 5 and 6.

This is not just an issue for print-first titles. Most of our born-digital books will include more elements than their derivative print counterparts. Publishers will need to resolve how to structure the various versions and decide whether they want to number consecutively in the full on-line version (and add an explanation of the gaps in numbering in the print version), or use some other scheme like adding the letter “E” to electronic elements.

5. Technology Development, Scholarly Publishing Office (SPO)

Working with SPO

Key to the success in developing the HEB publication system is our partnership with the Scholarly Publishing Office (SPO) at the University of Michigan Library <<http://spo.umdl.umich.edu>>. SPO is contracted by HEB to provide the technology back-end for the project, which includes backlist (scanned page images) and frontlist (XML) publication systems, search engine, server hosting, statistical reporting, user authentication, cataloging support, and backlist file conversion.

Developing the XML Publication System

To develop the frontlist XML publication system, HEB worked with SPO to enhance their XML system with new style templates and more interactive capabilities. HEB staff spent several months defining our functional specifications and creating design templates in HTML. SPO then refined and expanded their processing code and style templates to meet our specifications. This development required numerous rounds of designing, coding, testing, and negotiation.

In order to create an efficient system, SPO developed one set of processing code that could handle all the XML files in our collection. HEB worked with SPO to develop a DTD that streamlined our production workflow. HEB now sends a monthly batch of XML and associated files to SPO to process and upload. The XML files are indexed for searching and processed for web display. Note that transformations from the XML to styled HTML pages are done “on-the-fly” when a user views a page. Only the portion of text that a user chooses to view is transformed from XML into HTML format.

To transform and style XML files, SPO currently uses CGI scripts written in PERL. The scripts map XML elements to appropriate HTML tags and stylesheet classes, break text into chunks, and establish links and other interactive features. Both HEB and SPO are very interested in exploring new technologies, such as XSLT and Java servlets, to style and process XML. However, for now, SPO’s publication system serves us well; it is stable, flexible, and well integrated, with a robust suite of functions including searching, bibliographic management, statistical tracking, etc.

Resources from DLPS

To build the HEB XML publication system, SPO uses a suite of resources provided by the Digital Library Production Service (DLPS) group at the University of Michigan Library <<http://www.umdl.umich.edu>>. DLPS also licenses this publication system to a number of non-profit projects through a program called the Digital Library eXtension Service (DLXS). This

technology includes a base set of processes that can be edited and enhanced according to the needs of a project. The DLPS system provides us with the following resources:

1. XML Processing:
The system includes processing code to transform and style XML encoded texts for web delivery in HTML format.
2. Image Viewer with Zooming and Panning:
A browser-based image viewing tool with zooming and panning is available for books that have high-quality image files including color art, line art, and detailed maps. The image viewer tool also manages metadata associated with each image in a database so that users can easily search and organize collections of book images.
3. Powerful Search Engine:
A robust search engine allows HEB users to search within an individual book, or across all books in the collection, using simple, Boolean, or proximity searching options. In XML books, matched search terms are highlighted in red in the search-results list and within the text of the book. (See [Figure 4](#) and [Figure 5](#)) The system uses XPAT, a fast, XML-aware search engine that can handle advanced searching. XML-aware search engines index words in association with XML structure, so the user can search within particular elements like chapter, paragraph, or header.
4. Bibliographic Management:
Metadata, such as title, author, date, and subjects, are pulled from MARC catalog records and used across the HEB site for searching, heads, and title lists. SPO creates library catalog records in MARC format for each book.

6. Specifications and DTD Development

Developing the DTD

HEB developed the project's XML DTD with SPO by revising the Text Encoding Initiative's TEI Lite DTD to suit our needs. We began to work with a DTD that was very close to the TEI Lite version. However, as we moved through tagging and processing some samples, we saw that this DTD had many options available that we could not foresee using. It offered a high level of flexibility in tag and attribute usage, which made it difficult for us to clarify exactly how the publishers should tag certain elements. Furthermore, this flexibility made it difficult to program processing code for the various tagging possibilities. Therefore, we ended up stripping down the DTD to include only a core set of elements and attributes.

Having limited tagging options ensures that the XML files submitted to HEB will include only those elements that we have capability of processing. However, we make it clear to publishers that if their book requires a new element or functionality, we will work on adding it if appropriate. Before production begins on a book, we review each book at a production launch meeting to identify new requirements. As stressed earlier, our strategy is to use an additive approach, starting off with basic elements and functions and building new capabilities as needed.

Granularity in Tagging

A major challenge in developing HEB's DTD was deciding on the level of detail to tag. For many elements, the TEI standard suggests tagging text at a much higher level of granularity. However, some of TEI's detailed tagging specs were still unsupported in SPO's XML processing code. We had to assess whether the benefits gained outweighed the cost of tagging additional items and developing associated processing code. We concluded that some items were necessary to tag in detail, while others would take time and effort to tag and process, but would not necessarily be used.

Some examples of TEI tags that HEB did not ultimately include in the project's specifications include tagging of titles and elements within a bibliography item. TEI suggests tagging any title that appears within the text and assigning a type (monograph, journal, thesis, etc.) to the title tag. TEI also suggests breaking down each bibliography item by author, title, publisher, date, etc. After testing this on a few titles, we found that, since history monographs refer to many different types of source material, it was not always clear-cut how we should tag titles and sub-elements within the bibliography. This generated many questions from the publishers' vendors and often required editorial input, and still such elements sometimes could not be definitively assigned to a particular type. Some bibliography items also had complex combinations like "written by A, edited and translated by B," making them cumbersome to tag. We realized that this detailed tagging would not be serving any immediate use, so we ultimately did not require it in our specifications. One possible use for detailed tagging of titles includes facilitating direct linking to on-line editions; however, the majority of titles tagged are unavailable on-line, and will most likely remain so in the near future.

HEB did decide that detailed tagging would be useful for certain elements. For example, breaking down chapter/section heads by number, title, and subtitle, and figure captions by number, caption, and source, gave us more flexibility in styling and also allowed us to extract information for tables of contents and figure lists. An item that SPO's base system does not currently process in detail is the table element. At the moment, table row and column heads and table source notes are not specifically tagged. Attributes, such as border and column and row span, are also not allowed. We most likely will add elements based on the HTML 4.01 table model since this allows for a simple conversion from XML to HTML, our final display format.

Multiple DTDs

While working with SPO on back-end processing, HEB developed two versions of the DTD, one to be used by HEB and publishers and another to be used by SPO for processing. As is typical in XML production workflows, different DTDs are used at different production stages. SPO had strict constraints on what their DTD could include. At HEB, we wanted to be able to control our own DTD and include elements used during our production process or preserve tagging for elements that might be used in the future. Also, SPO's system requires element names to be in all-caps and special characters to use number references (e.g., è), while at HEB we use lower-case elements and character entity references (e.g., `). To resolve our different requirements, HEB and publishers use one production DTD and SPO uses another DTD with more constraints. When HEB is ready to send an XML file to SPO for uploading, we run a PERL script that converts our XML file into one that conforms to SPO's DTD.

Metadata in MARC Format

Note that such book metadata as author, title, date, publisher, and subjects are not heavily encoded within the XML files because the metadata on the HEB site is pulled from our library catalog records. A professional librarian at SPO catalogs each title according to standard MARC cataloging rules. Rather than introduce another source of metadata within the XML file, we are using the MARC data in headers, title lists, bibliographic records, and search indexes. While a few MARC rules do not conform to common practice (e.g., lower casing of some title words) or are not immediately clear in meaning to users (e.g., [1999], 1999, or c1999, depending on where copyright year appears), it is still the best way for us to manage the metadata on our site. This provides consistency and standards to the metadata and conforms to what our subscribers, who are currently mostly librarians, use as their primary bibliographic standard.

Special Character Encoding

In order to ensure that special characters are properly displayed and indexed in the search engine, SPO limits the use of special characters. At the moment, SPO can process all special characters in ISO Latin 1. They require that any special character above the 128 basic ASCII characters be tagged using an entity reference (e.g., ´).

To make sure that special characters are encoded using entity references, HEB requires publishers to set the encoding in their XML files to US-ASCII. So if there are characters above the 128 basic ASCII not tagged as entity references, when the XML is parsed, an alert will appear. For those characters or symbols beyond ISO Latin 1, we work with SPO on adding the character to the list. For some special characters or symbols that are hard to display using common fonts, SPO can render the character or symbol using an image inserted in-line with the rest of the text.

While we recognize that XML can now take advantage of Unicode encoding, we are not encoding our XML in Unicode because it includes character sets that many users do not have installed. Furthermore, SPO cannot index all the characters in the search engine. So we accept SPO's current strategy of using this limited list. Their search engine currently maps accented characters to unaccented characters so they can be indexed together for searching. For example, if a user types in "Francois" without the cedilla, the search results will include hits on both "Francois" and "François." If, in the future, our search engine and more browser fonts can handle Unicode characters, we can switch to Unicode encoding.

7. Design and Stylesheets

Designing and Styling E-Book Pages

SPO's base system offers generic design and styling options that can be enhanced with new design specifications. HEB began by creating sample pages in HTML for SPO to use as design templates. Since we added many new elements and features to our frontlist books, we made a number of page-design changes. These included decreasing the width of the text block, adjusting line spacing, removing line breaks separating text by print page, changing fonts, adding paragraph numbers, and centering figures within a text block. However, since SPO's pages are built with data coming from different parts of their system, not all of our requests could be implemented within their framework.

For the sake of efficiency and consistency, all XML files are processed using the same scripts and stylesheets. The process for transforming an XML text into styled HTML pages includes:

1. mapping elements to corresponding HTML tags (e.g., <q1> quotation tag to HTML <blockquote> tag)
2. adding style classes to certain elements (e.g., class="pnum" to style paragraph numbers)
3. setting a default style to a specific HTML tag (e.g., line spacing for paragraphs)

SPO's basic system provides generic style formatting, which is set in a cascading stylesheet (CSS) called "text-class.css." HEB's project-specific style enhancements are set in another stylesheet called "text-specific.css." At the beginning of our design process, we spent considerable time and effort going through each book element and set parameters for font, margin, spacing, color, alignment, and other style specs. This process included input from design, editorial, and technical staff members at HEB. Publishers also sometimes request certain design changes. With styles standardized, we now edit and design new styles as needed.

HTML Limitations

The advantage of distributing HEB texts in HTML is that they can be read using any standard web browser. However, HTML encoding offers limited styling options, especially when compared to options available in print page-layout software. For print books, publishers work hard to create easy-to-read pages by following certain industry design standards. For example, most book designers carefully adjust spacing and leading, restrict a "widowed" word at the end of a paragraph, restrict placement of ellipses at the end of a line, etc. This kind of control is not possible with HTML pages rendered in web browsers. Also, some standard styles used in print do not work as well on-screen and thus are modified. For example, on-screen, superscripting note numbers often throws off line spacing. Thus our note numbers are put in brackets rather than superscript.

Another unavoidable problem with HTML page design is that all web browsers render HTML in slightly different ways. We tested HEB pages using different browsers running in various versions of both Macintosh and Windows operating systems, and adjusted our coding to display best across the most popular browsers. However, the pages inevitably vary somewhat in styling across browsers.

Faced with these issues, HEB worked for several months to carefully convert standard print approaches to layout and design into HTML-compatible solutions. Since broad web-based access is our primary delivery goal, then HTML is still our best option for distribution, even with some loss in design accuracy.

Styling with XSLT

XSLT (Extensible Stylesheet Language Transformation) scripts have become indispensable production tools for the HEB project. In-house programmers at HEB developed XSLT scripts to transform XML files into styled HTML pages so that we can preview files before they are uploaded. The XSLT script transforms our XML files into web pages that look exactly like pages processed on SPO's live server. This allows us to preview the text in order to test whether the XML was tagged correctly and check if all the elements can be styled properly. For example, if a book uses a new combination of elements (e.g., images, paragraphs, and tables to create a slide

show), when we run the XML through the XSLT transform, we can preview the text to see if our stylesheets can handle the new combination properly.

HEB looks forward to the possibility of using XSLT instead of CGI PERL scripts to style XML pages on the live server. Using the same XSLT script during production and on the live server would be more efficient and give HEB more power to edit styles. As XSLT technology becomes more popular, we want to take advantage of new software programs and middle-ware tools capable of handling XML publishing using XSLT. HEB tested one of these server-based XSLT software tools, Cocoon, which is developed by the Apache group. However, it proved to be quite slow and unable to handle our complex chunking options. DLPS and SPO are researching the possibility of styling pages on the live server using XSLT in the future.

8. Production Workflow

“Print-First” to “Born-Digital” Books

In this first round of HEB development, it was helpful to begin by producing print-first frontlist titles with the project’s participating publishers. Putting a few print-first titles through production gave publishers the chance to set up vendor relationships and workflows, and focus on how core elements within typical scholarly texts function on-line.

With the XML publication system in place, publishers and authors can easily expand on this basic print-first model. As HEB moves forward, books are evolving beyond expansion of print-first editions to more robust second editions, and to “born-digital” works that incorporate electronic elements and structures at the point of conception.

However, it is important to note that works that make good use of electronic features are not necessarily overly complex to tag and process. For example, HEB has works that incorporate hundreds of color images, links to academic resources, side-by-side comparisons, or linked concordance charts—all of which are difficult to produce in print. However, to prepare the files for HEB, the publisher can simply tag the text using HEB’s basic building blocks of links, tables, image pop-ups, etc., to produce a rich on-line scholarly work.

Outline of Production Workflow

In the section below, we outline the production workflow HEB has used for preparing XML and associated book files. After a frontlist title has been selected for inclusion in the project, the HEB acquisitions and production staff review the title at a production launch meeting. Following the launch meeting, production staff at HEB begin working with the production editor at the participating publisher. The publisher is responsible for submitting the specified text encoded in XML along with images and other multimedia files prepared according to HEB specifications. Note that for these e-books, publishers apply the same quality checks and standards that they impose on their print books. Here is an outline of a typical production workflow with approximate length of time per stage in brackets:

1. Publisher Submits Schedule A [varies]

- Publisher submits outline of e-book contents, associated bibliographic information, and estimated submission dates for a new title. This information is submitted on a Schedule A form (schedule from book contract between publisher and HEB).

2. HEB Staff Reviews Manuscript or PDF [1 week]

- HEB reviews manuscript or PDF if the title is a print-first conversion.
- HEB acquisitions and production staff review materials (manuscript/PDF/book) and Schedule A at production launch meeting. At the meeting, contents and functional features for the e-book are confirmed and new development needs and issues identified.

3. Publisher Prepares Files [5 weeks]

- The publisher works with HEB on structure, size of chunks, numbering schemes, insertion of new elements, linking opportunities, image handling, and identifying problematic elements.
- The publisher then works with a vendor or in-house staff to prepare XML and associated files using the latest HEB specifications.
- HEB advises on tagging and usage, striving for consistency of usage across books. HEB can review files at this stage if necessary.
- If the publisher has new elements or structures not included in our specifications, HEB consults with SPO on the best way to handle new elements. DTD and processing are updated if necessary.
- The publisher submits XML and associated files to HEB.

4. HEB Quality Review and Final Preparation of Files [2 weeks]

- HEB reviews XML files to check for conformance to specifications. HEB will make corrections to XML, but if edits are extensive, then XML is returned to publisher.
- HEB checks quality of images and other files.
- HEB uses XSLT scripts to create a preview version of the e-book to check tagging and design-styling.
- HEB runs PERL scripts for quality checking.
- HEB converts XML tagged using HEB's DTD to XML tagged according to SPO's DTD.
- HEB updates its in-house databases for related titles, related reviews, copyright, and image captions.
- HEB submits files to SPO. Files are submitted to SPO in monthly batches.

5. SPO Upload and Processing [2 weeks]

- SPO indexes text for search engine, processes and loads XML file on their test server.
- HEB reviews book design and styling and requests changes to style or programming of new functions if necessary.
- SPO librarian creates new MARC catalog record.
- SPO uploads database information for related titles, related reviews, and copyright.
- If high-resolution files are included, SPO uploads figure information from database and compresses files in SID format for distribution using the image viewer tool.

6. Publisher Proofs [4 weeks]

- Publisher proofs e-book on production server using HEB proofing guidelines.
- Publisher submits corrections to HEB.

7. Final Edits [2 weeks]

- HEB inputs publisher's corrections.
- HEB works with SPO to add new style or function if necessary.
- HEB sends final XML file to SPO.
- SPO uploads file.
- Publisher approves final version.

8. Book Live

- SPO moves final files, database information, MARC records, and search indexes to live server.

Quality-Control Procedures

To ensure that e-books are tagged and styled properly, HEB uses the following quality checks during the production cycle.

1. DTD Validation:

XML files are parsed against the HEB DTD to check proper use of elements, attributes, and special characters. Publishers and their vendors must validate files before submission. At HEB we validate files using XML Spy, an XML editing software program.

2. XSLT for Proofing:

HEB spot-checks a preview version of the e-book before sending files to SPO for upload. Using an XSLT script, the XML file is transformed into HTML pages that mimic the on-line book with all text and images properly displayed, but some interactive features (e.g., note pop-ups) not available. This tool helps us identify incorrect tagging and bad formatting. The preview HTML file refers to the same CSS stylesheet used on the live site, so we can also easily check that each element tagged is properly styled. Although we developed the XSLT script as a tool for in-house use, we do share it with publishers and vendors so that they can preview their files before submission.

3. Quality Checking with PERL Scripts:

Some tagging and usage cannot be checked by parsing against the DTD, so we use scripts written in PERL to check these items. The scripts make sure that all paragraph, note, figures, etc. are numbered consecutively and checks that rules, such as "no paragraph numbers in quotations," are followed. The PERL scripts also make reports showing tallies of all elements used and mark the location of each element. For example, a report might show that the book has 10 <div1>, 25 <div2>, 150 <figure>, 300 <note1>, etc. This is useful because we can make sure that all elements used can be processed and styled. If new combinations of elements and attributes are used, such as lists in quotations or line groups in epigraphs, we can easily identify and locate them. Though another standard for describing rules for XML documents, called XML schemas, offers some validation and checking options, we found that PERL is faster and more flexible for our purposes.

4. URL Link Checks:

All external URLs are extracted from the XML file and tested for dead links. HEB is currently using a shareware link-checking program called Xenu. URL checks are run every quarter, and a list of dead links is submitted to the publisher for review. Dead links can be either deleted or updated accordingly.

5. Publisher's Final Proofreading:

Since HEB wants to make sure that frontlist books are clean and of the highest quality, the project also requires publishers to proofread and approve the final on-line version. HEB provides publishers with proofing guidelines that include a list of common issues <<http://www.historyebook.org/xml/doc/proofing.html>>. Typical problems found include incorrect numbering, italic fonts missing, special characters not rendering correctly, wrong upper or lower casing of heads and small-cap text, and table formatting issues.

9. XML Tagging Options and Costs

Options for XML File Conversion and Tagging

To prepare XML files for HEB, publishers can use in-house staff, outside composition or conversion vendors, or freelancers. Publishers can choose to work with any vendor from their established list or one suggested by HEB. The participating publishers who already work with XML use their XML compositors to transform their files to HEB specifications. We want publishers to have the option of working with their own vendors so they can more easily build up XML output options into their workflow. Most publishers found vendors on their list with XML experience; and in general, these vendors were eager to take on an XML job. Publishers and vendors produce files according to HEB specifications, which are available online at <<http://www.historyebook.org/xml/doc/acls-hebook-doc.html>>.

The following list describes the various methods HEB publishers are using to produce XML encoded books.

1. XML Conversion Vendor:

XML conversion vendors are set up to convert from many formats into XML and are proficient at working with new DTDs and specifications. This method has been the most efficient with the fastest turn-around time.

2. Compositor, Page Layout to XML:

Quark is the most common print layout software used by our publishers and their compositors. Some compositors have established conversion routines since they convert many titles from Quark to XML, but many compositors use a mix of scripts and hand-tagging. Efficiency of conversion relies on the rigid use of templates and application of certain rules in the Quark file. However, pages typeset for print often do not follow these rules. Common conversion issues result from inconsistent use of templates and styles, use of italic, bold, ligature or rare fonts, forced hyphenation and line breaks, inclusion of non-linked text boxes, and complex layouts and tables. We have not yet had a critical mass of conversion using InDesign-based workflows. In any case, detailed proofing is mandatory.

3. Compositor, XML to XML:
Some HEB publishers encode books in XML using their own DTD. Converting to our DTD and specifications is efficient and accurate. However, our specifications do call for items that often are not included in the publisher's XML. These elements must be added and often require manual input (e.g., page breaks, division breaks, linked index, linking URLs, chunking structure, etc.).
4. Manual Tagging by Freelancer or In-House Staff:
Though tagging by hand can be time consuming for longer titles, it works well for publishers who are working on titles that are in development and use many new elements and experimental structures.
5. Programmer, Global Scripts:
This method was used by one publisher that had previously used freelance programmers for converting Quark files into XML. For the HEB project, their programmer used global scripts and some manual tagging to produce the XML file, but the file then needed further proofreading to ensure accuracy. This increased the cost and turn-around time.
6. Full-Service Composition:
For upcoming "born-digital" books, written and produced with multiple formats in mind, some publishers are using full-service compositors to take care of copyediting content, trafficking of files, and production of print pages and electronic files. Publishers with in-house DTDs often use their DTD during production, but keep HEB specs in mind so that XML transformation to our DTD is efficient.

Conversion Costs

At this stage, it is still too early to analyze and compare costs for the various conversion methods just outlined. Some compositors accepted conversion jobs to be able to begin working on XML projects with a publisher. Other compositors wanted to test their XML production conversion workflows. Most compositors were willing to absorb ramp-up costs associated with learning the HEB DTD. Vendors undoubtedly want to remain competitive and hope that they will get more projects in the future. Thus, prices charged to the publishers at this stage are not necessarily reflective of the real cost of production.

However, the costs for preparing files using XML conversion vendors are stable and can be used as a point of reference. XML conversion vendors recommended by HEB have been charging publishers around \$.50 per 1,000 characters, including spaces and tags. Thus, for a 300-page book, total costs for preparing the XML file might be \$500 to \$600. Image-processing costs vary, though most publishers have been able to convert image files easily from the images used in the print edition. Not all XML conversion vendors are willing to take on books one at a time since they are accustomed to working in volume. Higher volume could reduce costs to below \$.50 per 1,000 characters.

With HEB's born-digital books, there will be a shift away from post-production conversion to preparing files from the outset with electronic output in mind. For new books with simultaneous

print and electronic outputs, costs for preparing XML files will be bundled into the total cost of production rather than comprising a separate post-production conversion expense.

10. Conclusion

Accomplishments

When HEB started the technical and workflow development for the frontlist books, the major challenge was to build a system that was structured and cost-efficient, but flexible enough to handle innovative types of electronic books. After 18 months of development, we have successfully built an efficient XML publication system and streamlined our production workflow. Major accomplishments include:

1. Established HEB E-Book Specifications:
HEB has established standardized specifications, including all e-book elements, functions, and display styles. HEB invested considerable resources in time, money, and expertise to move beyond simple and straight conversion of the traditional print book, and developed effective solutions for handling e-book specific issues, such as citation (using paragraph numbers for reference IDs) and reporting and display (chunking a text in various sizes).
2. Developed Documentation, Procedures, and Tools for Production:
HEB staff developed detailed XML tagging documentation, proofing guidelines, sample files, and production tools. Publishers and their vendors use these materials to create XML encoded e-books for HEB.
3. Developed an Efficient XML Publication System:
HEB and SPO developed a system that can efficiently process and publish HEB's XML and associated e-book files.
4. Streamlined the Production Workflow:
HEB production staff streamlined the production workflow, which involved various parties—production editors at participating publishers, outside vendors, HEB staff, and SPO programmers. HEB built good working relationships with the publishers' production staff, learned about each publisher's procedures, and provided guidance on preparing files. To help make production more efficient, HEB also created tools, such as XSLT scripts, for previewing books and PERL scripts for quality-checking.

With HEB's XML publication system and production workflow in place, the project is well prepared to deal with more complex books and assist authors and sponsoring publishers with cost-effective and reliable e-publishing models.

Upcoming Development

During the next phase of frontlist development, HEB plans to focus on the following:

1. Develop More Complex E-Books:
The next batch of e-books submitted to HEB includes more complex titles. Some titles will incorporate hundreds of color images and links, as well as scholarly tools, such as

concordance charts, side-by-side image comparisons, and interactive maps. A few titles are not based on typical chapter and section book structure, but rather are based on translations and transcriptions of letters, or series of images. HEB will need to enhance the XML DTD, design new stylesheets, and work with SPO on back-end processing for the new features. For HEB and SPO, the major challenge will be to expand the DTD and template-based designs to fit the increasingly varied types of e-books being submitted to HEB.

2. Implement Production Workflow for “Born-Digital” Works:

HEB will begin working with publishers on “born-digital” works. These books are written from the start for both electronic and print publication, and are submitted with elements specifically designed for on-line publication (e.g., hyperlinks, additional color images, pop-up boxes, etc.). Publishers will be producing the print and electronic edition simultaneously using one integrated production workflow.

3. Increase Efficiency of Production Process:

During the first round of development, many new procedures and relationships needed to be established. In the next round of development, HEB hopes to standardize turn-around time for each stage of production. However, as the e-books become more complex, production tasks will become more involved, so timing of each development stage is currently difficult to predict.

4. Revise HEB HTML and CSS Code:

Since SPO implemented HEB’s page design templates by adjusting DLPS’s basic code, some pages include HTML and CSS (Cascading Style Sheet) code that is not in conformance to accessibility standards. Though the pages all render without a problem, clean up of code will be necessary to allow for accessibility compliance.

5. Research New Standards for URL Linking and Permanent Identifiers:

As more electronic resources in the humanities become available, new opportunities for inter-linking among resources become possible. HEB will research new standards for linking and permanent identifiers, including Open URL and DOI (Digital Object Identifiers). HEB will continue to engage in discussions about these topics with other groups who are publishing scholarly history resources on-line, such as the American Historical Association, the History Cooperative, Gutenberg-E Project, and JSTOR.

Figures

Figure 1. Frontlist E-Book Title Record Page with Table of Contents

The screenshot shows a Microsoft Internet Explorer browser window displaying the ACLS History E-Book Project website. The page title is "ACL5 History E-Book Project: Title Record - Microsoft Internet Explorer". The browser's address bar shows the URL: <http://ferment.umdl.umich.edu/cgi/t/text/text-idx?sid=dae15bfe6e3244e3dff10d04e407a603;cc=acsl;c=:>

The website header features the logo of the American Council of Learned Societies and the text "HISTORY E-BOOK PROJECT". Navigation links include "search", "browse", "help", "home", and "terms". A secondary navigation bar contains "title record", "contents", "reviews", "search this book", "related titles", and "encoded text".

The main content area displays the book title "A time for choosing : the rise of modern American conservatism" by Jonathan M. Schoenwald, published in 2002 by Oxford University Press. A small image of the book cover is shown on the left. Below the title, there are links for "view full catalog record" and "enter book".

A section titled "table of contents" lists the following items:

- [Copyright and Permissions](#)
- [\[Dedication\]](#)
- [Acknowledgments \[para. 1-5\]](#)
- [Introduction \[para. 6-35\]](#)
- Chapter 1: The Birth of Postwar Conservatism
 - [\[Intro\] \[para. 36-43\]](#)
 - [Russell Kirk and Postwar Conservatism \[para. 44-52\]](#)
 - [Conservative Challenge to Containment \[para. 53-65\]](#)
 - [Two Popularizers: Albert Wedemeyer and Raymond Moley \[para. 66-78\]](#)
 - [Toward a Definition of Conservatism \[para. 79-82\]](#)
- Chapter 2: The Cold War Hits Home
 - [\[Intro\] \[para. 83-84\]](#)
 - ["Red Monday" at the Supreme Court \[para. 85-99\]](#)

Figure 2. Frontlist E-Book Page

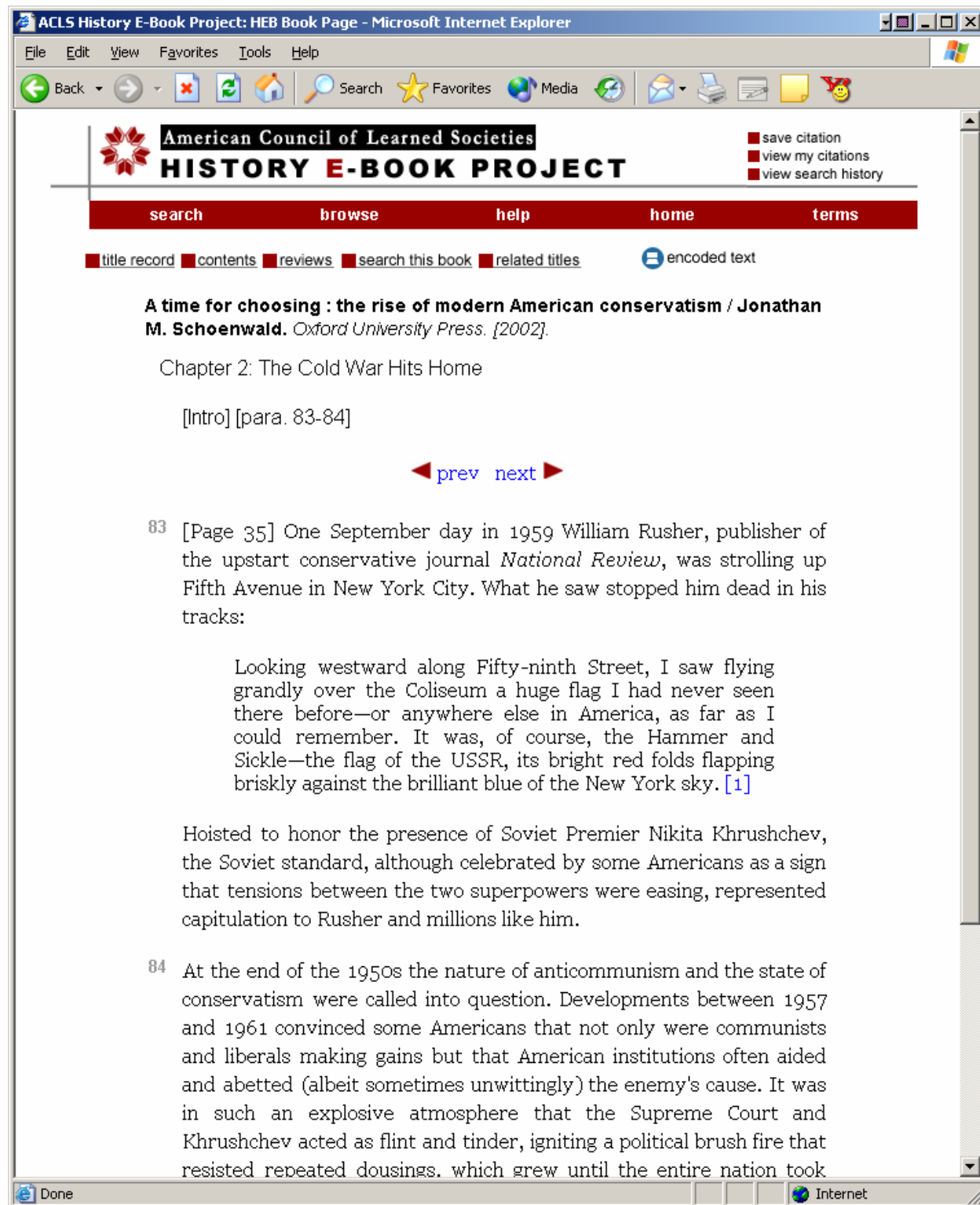


Figure 3. Frontlist E-Book Page with Images

134 These blind musicians, some of whom played in the orchestra led by Galliod at the Quinze-Vingts, would also perform at the Café des Aveugles (Blind People's Café). Located in one of the basements of the Palais Royal, the club offered up entertainment by blind musicians in a shady quarter of Paris frequented by prostitutes, gamblers, and others of dubious distinction. Alternating with acts such as clown-like jugglers, a man known for dressing like a savage and dancing to wild drumbeats, and other carnivalesque entertainments, the orchestra of blind men (including several teachers and an occasional female singer) hovered on the edge of professional and moral respectability, much to the dismay of many administrators at the institutions for the blind. [73] Some contemporary writers described the locale as a "hideous den of rogues and prostitutes," with the Catholic ultraroyalists who ran the Quinze-Vingts looking particularly unfavorably upon their student-musicians or singers who performed in such immoral places at all hours. [74] [Page 118]




Figure 5.
The Stage of the Café des Aveugles around 1800,
Anonymous, Musée Carnavalet, D 3105.





Figure 4. Frontlist E-Book Search Results Page

ACLs History E-Book Project: Search Results - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail News RSS

 **American Council of Learned Societies**
HISTORY E-BOOK PROJECT

[view my citations](#)
[view search history](#)

[search](#) [browse](#) [help](#) [home](#) [terms](#)


Results

Your basic search for "nafta", sorted by in **ACLS History E-Book Project** resulted in 71 matching terms in 2 full texts.

[Refine this search](#)

[return to full results list](#) << [previous item](#)

1 to 25 of 70 matches
[1](#) | [26](#) | [46](#) | [next](#)>

Author: Hart, John M. (John Mason), 1935-
Title: Empire and revolution : the Americans in Mexico since the Civil War / John Mason Hart. 

Publication date: [2002]
Search results: 70 matches in full text

[table of contents](#) | [save citation](#)

[Introduction: Imperial Ambition \[para. 3-15\]](#)

- ... vents and measure long-term effects. Manifest Destiny, World Wars I and II, and the ratification of **NAFTA** are all part of the story, but the depth of the American commitment to Mexico transcends momen ...

Part IV: The Reencounter, 1940-2000
Chapter 14: Return of the American Financiers

[American Capital and Mexican Debt \[para. 1241-1258\]](#)

- ... fforts to market products in the United States and may even have helped gain high-level support for **NAFTA** among New York bankers. / The largest fifty of the former pa ...
- ... The institution assumed a leading role in the corporate campaign to gain congressional approval of **NAFTA**. / Although Citibank was the only bank to operate op ...
- ... corporations. In late 1994, benefiting from the new tolerance of American financiers underscored by **NAFTA**, Morgan led an array of over seventy American and other foreign banking houses, sixteen broker ...

[Free Trade \[para. 1259-1271\]](#)

- ... at Alan Knight calls the "congenial and collaborating elites" of Mexico. **NAFTA** was at

Done Internet

Figure 5. Frontlist E-Book Page with Search Terms Highlighted in Red

ACLS History E-Book Project: HEB Book Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Refresh Print Mail Stop

American Council of Learned Societies
HISTORY E-BOOK PROJECT

■ save citation
■ view my citations
■ view search history

search browse help home terms

■ title record ■ contents ■ reviews ■ search this book ■ related titles encoded text

Empire and revolution : the Americans in Mexico since the Civil War / John Mason Hart. *University of California Press.* [2002].

Part IV: The Reencounter, 1940-2000
Chapter 14: Return of the American Financiers

Free Trade [para. 1259-1271]

< prev next >

1259 During the early 1990s the American economic and political leadership reached a final accord on **NAFTA**—the North American Free Trade Agreement—with what Alan Knight calls the "congenial and collaborating elites" of Mexico. [7] **NAFTA** was at the cutting edge of free-trade arrangements in the Americas [Page 438] at the end of the twentieth century. It reflected changes in the trade relationship between the United States and Mexico that had already taken place. The American propagation of **NAFTA** derived from long-standing business and geopolitical needs. The treaty satisfied a sense of urgency on the part of the American elite to stabilize a large and potentially chaotic neighbor caught in a state of rapid population growth and economic stagnation.

1260 The leaders of the United States, Mexico, and Canada created **NAFTA** as the legal framework for a restructuring of their economies. The pact was signed on 17 December 1992 by American president George Bush, Mexican president Carlos Salinas de Gortari, and Canadian prime minister Brian Mulroney. Its ratification in 1993 marked the opening phase of what President Bush called the "new world order," an era of worldwide free trade in which local "democracies" were willing to accept American economic leadership. The governments agreed that the United States would handle the high end of electronic, industrial, and farming technology and that Mexico would provide labor for a range of jobs, from menial to highly

Internet