



50 E. Huron • Chicago, IL 60611-2795

REQUEST FOR PROPOSAL

Response requested by July 1, 2005

RFP Overview

The American Library Association (www.ala.org) is seeking a vendor to build and implement a content management system for its Web site. The completed product will be fully owned, hosted, and maintained by ALA staff.

ALA seeks a Web-based content management system that, at a minimum, encompasses the criteria described below. These criteria describe our bottom-line needs; products that cannot prove prior support for systems with these requirements will not be considered. References of clients with systems and structure of similar complexity to ALA's will be required.

Stability

The system we choose must remain online, functional, and accessible to all users—including international users—at all times of the day and night. Frequent “crashes,” “lost connections,” and unplanned or unexpected periods of downtime are unacceptable.

Reliability/Scalability

The ability for users of this system (“content developers” — see Appendix A) to predict its behavior is paramount. Functions should be clearly described in documentation, and their performance should not be degraded as traffic, number of users, or number of content records increase. At the outset of this project, the vendor chosen should recommend a software and hardware solution that will not require regular upgrades in order to insure its performance as resource demands increase. Rather, ALA would prefer to overestimate what is needed in order to keep hardware upgrades, code optimizations, and other similar necessary enhancements to a minimum. See “Use of Web Site/Content Management System” below for specific load/growth requirements.

Flexibility

ALA's complex, dynamic structure demands a system that can accommodate the following:

- An unlimited number of graphical templates that can be easily edited for various ALA and division/publication initiatives

- The ability to restructure and add content without undue strain on the system, for example, no need to rebuild vast sections of the site when one portion is added or moved, as well as the ability to permanently delete and make totally inaccessible any piece of HTML content, graphic, document file, etc. Please see the Scope of Project section of this RFP for more information on deletion and archiving needs.
- The ability to be integrated with other association-related Web applications and business processes hosted on other servers. This will include reading data from MS SQL servers and being able to seamlessly manage login sessions across servers. These systems will include:
 - ALA’s Online Communities system
 - ALA’s Online Store
 - ALA’s iMIS member database
 - ALA’s iMIS Business Objects (iBO) based Event Manager, User Manager, and Dues Manager applications
 - Various home-grown Cold Fusion applications
 - Future Web-based projects
- The CMS and Web site it generates must be cross-browser and cross-platform compatible.

Speed

Multiple simultaneous requests for static and dynamic content on the site, as well as multiple edits, additions, and deletions of content should be accommodated with no visible slowdowns in performance. See Technology Overview for more information on specific requirements.

Ease of Use

ALA’s content developers range from highly skilled programmers and scripters to those with only basic understanding of the Web. The system should be easy enough to use for someone looking only to make quick, textual changes to existing content, but flexible enough to accommodate someone managing large portions of the site that may include javascript, dynamic pages, Flash presentations, etc. Both types of users—and the varying levels between them—should find that their work integrates seamlessly with the rest of the site, both graphically and functionally.

The more advanced of ALA’s content developers are comfortable with straight HTML coding and are familiar with the use of such technologies as FTP. Accordingly, they should not be forced to work through pages of wizards or other forms to simply make a quick textual change. Multiple entry points to content-editing tools should be offered (i.e., an “Edit This Page” feature for content managers who are logged in to the site and viewing pages that fall under their control). The CMS must support the use of all known document types and graphics and animation formats.

The CMS must accommodate an unlimited number of distinct templates. Content developers must be allowed to apply distinct templates to individual pieces of content. Content developers must be able to create and modify their own templates using any external Web authoring tool, and apply them to their content within the CMS. Dynamic pages must be able to designate a template in which all linked content will appear, no matter what template that content may have associated with it elsewhere in the CMS.

An illustration of some of our user profiles is appended to this RFP for your reference, as Appendix B.

Conversion from Current System

The content of ALA's Web site is housed in a MS SQL 2000 database. Pieces of the code that generates member-only pages reference ALA's association management system database, which is in MS SQL 7.0. Converting our current site into a new CMS should be an automated process and involve little to no work on the part of our many content developers. Absolutely no manual conversion of content should be done.

URL Stability

Due to circumstances beyond our control, a given piece of content may need to be accessible from many different URLs, all of which must be stable and accessible indefinitely. While some of the "redirected" URLs may be "dirty" or inappropriately long, the URLs automatically generated by the new Content Management System must be persistent, short, easy to alter/control by content developers and administrators, and act as permanent Digital Object Identifiers (www.doi.org) for each piece of content they represent, even if that content gets edited over time.

We will also need a mapping or redirect utility to assign "vanity" URLs to content. For example, there will often be a piece of content that is located, structurally, deep within the hierarchy of the site but that needs a short, easily published URL that appears to be directly off the root of the site. The CMS must allow redirects of a top-level file to flow down to all items beneath it, so that items in the hierarchy below an item that is given a directory-level redirect will all be automatically assigned redirects that begin with that shortened directory. This tool can stand alone or be part of the properties for every content item, but needs to be accessible by all those with the proper authority assignments.

RSS

The ability to syndicate site content for our users is becoming increasingly important. Either on a page-by-page basis or limited to specific types of content (press releases, journal articles, etc.), users of this CMS must be able to define, create, and manage content feeds for their content. The RSS system must also be flexible enough to allow per-feed customization such as adding or removing specific fields as they are needed.

Online Help/FAQ

The system should provide a Help or FAQ function for commonly encountered problems.

Link Checking

Either in an automated or user-initiated fashion, links on the site should be checked/be able to be checked regularly for viability, and a report should be generated and accessible to each content developer for the pages for which she or he is responsible. The link checker or resolver should be available through a link in the page properties section.

Traffic Statistics Solution

The ability to analyze statistics regarding the traffic on our site is crucial. An analysis tool need not be part of the CMS, but the logs must be in a standard format. However, Web statistics should be available from within the CMS, through a link in the page properties section that offers the option, "View statistics from this page."

Introduction to ALA

Organizational Overview

The American Library Association (ALA), founded in 1876, is the oldest and largest national library association in the world. Its concerns span all types of libraries: state, public, school, and academic libraries, and special libraries serving persons in government, commerce, and industry, the arts, the armed services, hospitals, prisons, and other institutions. ALA has about 65,000 members in the United States, Canada, and over 115 countries. The Association's mission is to provide leadership for the development, promotion, and improvement of library and information services and the profession of librarianship in order to enhance learning and ensure access to information for all.

With a membership of libraries, librarians, library trustees, and other interested persons from every state and many countries of the world, the Association is the chief advocate for the people of the United States in their search for the highest quality in library and information services. The Association maintains a close working relationship with more than 70 other library associations in the United States, Canada, and other countries, and it works closely with many other organizations concerned with education, research, cultural development, recreation, and public service organizations concerned with education, research, cultural development, recreation, and public services.

In order to meet the needs of its varying constituencies and their concerns, ALA has developed into a complex organization with an array of membership units, including the ALA Council (the overall governing body), the Executive Board (central management body), ALA committees, 11 Divisions, 17 Round Tables, and 57 autonomous chapters of the ALA, each with its own elective structure. In sum, approximately 5,000 members are actively involved in the work of the Association.

The ALA staff of over 270 persons consists of a small publishing enterprise in Middletown, Connecticut, a lobbying and public policy office in Washington, D.C., and the main headquarters in Chicago. The information needs of the staff are as broad as the Association's activities and also include association management issues such as meeting management, governing board operations, finance, and marketing. Most staff are "knowledge workers" with regular and frequent need to gather and synthesize data, information, and research in support of member activities or constituent information needs.

Technical Expertise of ALA Staff

Director, Information Technology & Telecommunication Services - Primary point of contact for all enterprise systems and manages all technical services for the association. Years of experience: 15+

Assistant Director, Information Technology & Telecommunication Services — Primary point of contact for database interfaces using Microsoft SQL 7.x and 2000. Years of experience: 6

Manager of Web Development — Primary point of contact for Web-based development. Experienced ColdFusion developer using Microsoft SQL 7.0, 2000, and Access 2000 databases. Years of experience: 3

Web Developer - Accountable for the overall development, configuration, deployment, and administration of all installed Web applications. Experienced ColdFusion developer using Microsoft SQL 7.0, 2000, and Access 2000 databases. Years of experience: 6

Web Administrator - Accountable for the overall system support and configuration for large diverse Web sites and two list processors (ListProc and Sympa). Experienced ColdFusion developer using Microsoft SQL 7.0, 2000, and Access 2000 databases. Years of experience: 3

Network Administrator — Accountable for firewall and TCPIP infrastructure. Experienced Novell and Microsoft network engineer. Years of experience: 8

Content Developer(s) — More than 120 staff and members who maintain content on ALA's Web site. See Appendix B for details on ALA's Content Developers.

Technology Overview

Desktop Hardware/Software

Approximately 355 workstations distributed as follows:

Headquarters Office — ca. 300;

Washington Office — ca. 30;

Connecticut Office — 25

Workstations: PC's — Pentium 4, 2.4 ghz and above

Operating System: MS Windows XP

Office Software: Microsoft Office 2000 and Microsoft Office XP

Other Software: Desktop Publishing applications (InDesign, Quark, Framemaker)
Internet Explorer, Mozilla Firefox, GroupWise Email 6.5,
Symantec Virus Protection, Crystal Reports

Network

Operating System: Novell 6.x and eDirectory 8.7x, MS Windows NT/2000/2003 application servers

Topology: 100BaseT Ethernet, 3 — T1 lines to Internet

Servers: 15 MS Windows NT/2000/2003 servers, 7 Novell servers, 2 Linux servers performing network, communications, email, and application functions running HP/Compaq Proliant hardware

Core Business Applications

Association Management:

iMIS 4.41 Association Management System 130 user license (MSSQL 7.0) — used for both billing and A/R

Accounting:

GEAC SQL Financials 6.0 (MSSQL 7.0);
ADP 4.x;
HR Perspective 4.x

Library:

OCLC Connexion
OCLC FirstSearch
Dynix Horizon Information Portal 3.06.A
Dynix Horizon 7.3.3

Web Applications:

Web Content Management System (MSSQL 2000, Cold Fusion 5.x with ActiveMatter, IIS 5.0, indexed with Google Search Appliance);
Estore (Tango 2000, MSSQL 7.0, IIS 5.0)
WebCT Courseware 4.10.20 Campus Edition (WebCT on Sun Microsystems 9.x)
List Processors (ListProc 8.1 on Linux 2.x and Sympa 3.x on Linux 2.x)
NetOn-Line Community Builder

In addition, ALA staff use home-based PC's in a variety of configurations, including Macintosh operating systems, laptops, and personal digital assistants (PDA's).

Use of Web Site/Content Management System

The ALA Web site generally gets more than 5,000,000 requests for pages a month. This traffic can more than double during peak periods, particularly around ALA's two conferences, which occur approximately six-months apart and include announcements of special events (e.g., announcement of the Newbery and Caldecott winners). Traffic to ALA's site, even during peak traffic periods, tends to be spread across a large number of different pages. ALA's site needs to be able to handle the stress of the high-volume periods, and any individual page or application that may become highly popular should not degrade server or site performance for other users.

More than 120 distinct users—with that number subject to continued growth—have content creation and/or publishing rights on ALA’s Web site. The majority are ALA staff, but a significant percentage are ALA members. Changes to the site occur daily and concurrently. Multiple users will have rights to the same content, but may have varying levels of authority to make types of changes to the content.

The site has more than 35,000 “pages” and is expected to grow unabated. Some content is repurposed for use in multiple locations.

Authenticating member status with ALA’s member database (iMIS) will be necessary to facilitate the development of member-only content. ALA’s membership structure is complex and unique, and prospective vendors should not underestimate the time necessary to understand all the variables that determine a given member’s viewing rights.

WAI Accessibility Guidelines

The content management system’s output must meet ALA’s compliance with W3C’s Web Content Accessibility Guidelines. Compliance with W3C’s *Web Content Accessibility Guidelines* includes:

1. Providing content that, when presented to the user, conveys essentially the same function or purpose as auditory or visual content.
2. Ensuring that text and graphics are understandable when viewed without color.
3. Mark up documents with the proper structural elements. Control presentation with style sheets rather than with presentation elements and attributes.
4. Using markup that facilitates pronunciation or interpretation of abbreviated or foreign text.
5. Ensuring that tables have necessary markup to be transformed by accessible browsers and other user agents.
6. Ensuring that pages are accessible even when newer technologies are not supported or are turned off.
7. Ensuring that moving, blinking, scrolling, or auto-updating objects or pages may be paused or stopped.
8. Ensuring that the user interface follows principles of accessible design: device-independent access to functionality, keyboard operability, self-voicing, etc.
9. Use features that enable activation of page elements via a variety of input devices.
10. Using interim accessibility solutions so that assistive technologies and older browsers will operate correctly
11. Using W3C technologies (according to specification) and follow accessibility guidelines. Where it is not possible to use a W3C technology, or doing so results in material that does not transform gracefully, provide an alternative version of the content that is accessible.
12. Providing context and orientation information to help users understand complex pages or elements.
13. Providing clear and consistent navigation mechanisms—orientation information, navigation bars, a site map, etc.—to increase the likelihood that users will find what they are looking for at a site.
14. Ensuring that documents are clear and simple so they may be more easily understood.

For more information about these guidelines please go to <http://www.w3.org/WAI>.

Scope of Project

Section I. Content Creation and Editing

Overall, the editing and design using the CMS must be fast, friendly, and flexible. The CMS must be cross-platform compatible; that is, usable in all browsers and on all operating systems. Content developers with varying skill levels must be able to create and publish content quickly and easily for a variety of purposes and audiences.

The HTML/WYSIWYG editor

- The default text editor must allow editing in both HTML code and WYSIWYG views.
- The default text editor must be readily available as soon as login is complete. Content developers should be able to access the editor first, and then open up each section they have authorization to access.
- Content developers must be able to begin creating and editing content with a minimal number of clicks (e.g., editing function buttons should appear both at the top and bottom of screens).
- Advanced content developers must be able to use the text editor as a feature-rich HTML/XML authoring tool.
- Content developers must be able to save content and continue editing without having to exit and reenter the text editor, regardless of the view (WYSIWYG or HTML).
- The text editor must be XML-compliant and browser independent.
- The text editor must support the doctype declaration specified in the page header.
- Content developers must be able to find and replace words or text strings in either WYSIWYG view or HTML view, both in the displayed text and in the source code itself. This is especially helpful for removing artifact code that interferes with display (such as in tables or images), as well as for making editorial or other changes quickly and globally to content. Extra points for being able to limit find-and-replace to content within a given section of the site.
- In HTML view, content developers must be able to view tags in color (as in HotDog, CFStudio, etc.).
- The CMS must accommodate special characters, including euros, smart quotes, and diacritics.

- The CMS must have a wizard or other special feature for creating simple email forms.
- The CMS should allow tables, which enable the architecture of complex pages, to be edited both by dragging rows/columns and by an advanced palette that changes widths by percentages and pixels, in-table text color, borders and spacing, and other characteristics on a par with Dreamweaver’s table editor.
- The CMS must provide a tool for uploading variant file types—Word, PDF, etc.—for the same document, which would then appear as one entity followed by choice of file types; for example, “Board Document [[PDF](#)] [[DOC](#)] [[TXT](#)].”
- Content developers must have the capability to seamlessly copy a page’s published path to the clipboard, even before that page has been completed and published.
- The CMS must allow unrestricted page lengths and quantity of content.

Previewing content

- Content developers must be able to preview new or changed content easily in order to make multiple adjustments without extensive waiting and reloading.
- The CMS must refresh the preview screen to show repeated modifications.
- Content developers must be able to show unpublished work in progress to outside groups, and restrict their access based on established criteria or permissions.

Interface with other applications

- The CMS must allow copying and pasting code and formatted text from other applications—such as Word, PageMaker, InDesign, or MS Publisher—using methods that preserve formatting and special characters.
- The CMS must allow code to be imported and exported to and from such software as Dreamweaver, HotDog, and other authoring tools.

Dynamic pages

- Content developers must be able to remove links or text, and troubleshoot dynamically generated elements on dynamic pages.

- Content developers must be able to control fonts, formatting, sequencing, and wording (such as “Related links,” which might be called “Other resources,” “Neat stuff on this topic,” or “Board documents”) on dynamic pages.

Validation

- The CMS must be able to clean HTML/XML tags and optimally strip out extraneous Word coding, but should not over-clean. Content developers must have the option to override cleaning.
- Content developers must be able to validate W3C compliance. The CMS validation tool must identify code violations by color or some other visual reference, rather than by line numbers.

Section II. Site Design and Graphic Formats

The CMS should allow advanced WYSIWYG HTML editing that allows source code manipulation, whether that is provided in the default HTML editor or through another process. It should work as seamlessly with third-party development tools (including but not limited to Dreamweaver, Go Live, and ColdFusion) as possible to allow in-house graphic specialists to create custom pages and “sites” (CMS sections cloaked as separate Web sites) at the level asked of them by the association’s units.

General design concerns

- Content developers must be able to copy layout (tables and text) from Dreamweaver and paste directly into the default editor. Layout, headings, and other design elements should be retained entirely, and the view must remain WYSIWYG without reverting to code. It must be possible to copy both as code and as WYSIWYG.
- The CMS must carry over ALT tags on images when content developers copy image coding from Dreamweaver into the HTML/WYSIWYG editor.
- The CMS must keep JavaScript elements (image rollovers, image rotation) intact when coding is copied from Dreamweaver to the HTML/WYSIWYG editor.
- The CMS must keep form elements (form scripts and buttons) intact when copied from Dreamweaver to the CMS editor.
- The CMS must allow source-code editing for content developers.
- Content developers must be able to upload cascading style sheets (CSSs) and link to them from individual pages. Individual pages within the same site or group of

pages (e.g., all pages for one, 10-page campaign site) must be able to support separate style sheets without interference from CSSs on the main site.

Graphics/images

- The CMS must accommodate fast, uncomplicated image uploading, without tedious approvals.
- Content developers must be able to upload images independently of the HTML/WYSIWYG editor.
- Content developers must be able to upload multiple images at once.
- Content developers must be able to overwrite images (i.e., replace graphics with the same file name) and PDFs easily. ALT and other graphic-specific editing tags should be stable during this process.
- Content developers must be able to easily delete graphics and PDFs.
- Content developers must be able to drag and drop images into the CMS. It should take about two clicks to get to the image repository. The image repository should look like a window of folders on a server. The view should not have to refresh when a folder is clicked on.

Special formats

- Content developers must be able to upload any type of media content to the CMS (MOV, EPS, PDF, Powerpoint, DHTML, MPEG, streaming video, Quicktime, .ZIP, etc.)
- Content developers must be able to add Flash presentations.

Special features

- The CMS must enable time-stamping (“page last updated on mm-dd-yyyy”) and “you are visitor number n,nnn” coding.
- The CMS must support use of server-side includes (SSIs).

Section III. Content Properties

This section concerns other features that involve display and functionality. Many of these are incorporated in the “properties page” of our current CMS, but may be configured differently in another system.

Display

- The CMS must allow content developers to specify and change the hierarchical sorting of links, pages, or content records (by date, alphanumeric, custom, system default).
- Publishing must be quick and result in immediate visibility of the new content.
- Dynamic pages must update either immediately after publishing or after resetting the cache (if necessary).
- Content developers must be able to specify when content will be published (i.e., schedule publication for a future date).
- Content developers must be able to specify a “copyright date” (persistent yet modifiable, in case of error) for all pages. The copyright date will not change on republication.

Archiving

- Content developers should be able to temporarily flag content as “unavailable” to the public.
- The CMS must provide a visual flag to site visitors that certain content is archival and resides on a separate Knowledge Management System.

Page manipulation

- Content developers must be able to completely delete single or multiple content records (HTML, PDF, image, and others) permanently, after agreeing to one or two “Are you sure?” messages.
- The CMS must alert content developers to the fact that a page they are attempting to remove is linked to from other pages within the system.
- Content developers must have the option to clone or duplicate pages by making either a one-time or permanent template from existing content.
- Content developers must be able to move or relocate content without deleting and recreating it.

Site navigation

- Content developers must be able to navigate the CMS quickly and intuitively, perhaps using a pull-down menu to get to specific areas.

Navigation menu

- Content developers must be able to indicate which items should be included in the navigation menu and which should be excluded.

CMS search

- The CMS should provide a repository search engine for content developers that offers functionalities useful to site maintenance.
- Any search engine used by content developers should provide the capability of acting (editing, deleting) on specific items within the result set.
- Any search engine used by content developers should have the capability of searching all metadata and conducting fielded and combined-field searching using logical operators.
- Any search engine used by content developers must minimize multiple hits on what is essentially the same page with different dynamic URLs.
- Any search engine used by content developers should allow searching the CMS by navigation terms to identify pages created by previous content developers.

Error handling

- The CMS must offer appropriate “Are you sure?” messages throughout the system to ensure a minimum of deletion errors.
- Error messages to content developers from within the CMS must be clear, specific, and free of jargon.

URLs

- The CMS must encourage use of short static URLs, either when they are automatically generated or when a content developer assigns them.
- The CMS must allow static URLs with the same name to exist in different sections or folders within the system (i.e., two different johnson.pdf’s can reside in both /acrl/may04/ and /acrl/july04).

- The CMS must not create multiple versions of a static URL (such as “Default2446.htm” for a content developer–assigned file named “outsourcing.htm”), because the Default files are not revised when the original is changed, and links often inadvertently point to the Default.
- The CMS should keep the number of dynamic URL versions it creates for a static URL to an absolute minimum.

Section IV: Metadata

The CMS must provide for the collection and appropriate use of metadata elements related to content. In some cases, the system should not only prompt the input of metadata elements, but also require it. Some of the elements and features discussed below may need to allow override by appropriately-empowered users.

Date-Related Metadata Elements

The CMS should accommodate the need to enter the following types of date-related metadata elements:

- Date of content creation
- Date of content copyright
- Date of an event or deadline contained in the content
- Date for future publication
- Date on which the content creator should be prompted to review
- Date of content obsolescence

The CMS should prompt content developers to enter date-related information. Some fields, such as “date to review” or “obsolescence date” should be required. There should also be an option to override required date information, in essence making some content “evergreen.”

Date-related metadata elements should be used by the system to:

- create compiled pages of date-sensitive event pages
- publish pages at a specified future date
- prompt content creators to renew or remove content
- delete content if no action is taken to retain it
- move content to an off-site archive (KM system)*

Date-related metadata elements (e.g., event date) should be used to remove or change links to content from compiled pages (e.g., future conferences to past conferences) without intervention by content developers.

* Display of such content should be visibly flagged as archival and not readily available.

Subject-Related Metadata Elements

The CMS should provide the means to identify content by its creating unit(s) and content developers as well as the subject(s) of the content.

Unit-identifying metadata elements (e.g., ACRL) must be applied automatically to all content created by a unit.

Subject-identifying metadata elements (e.g., academic library) should be able to be applied automatically for specified content, with the ability of the content developer to override when necessary.

The CMS should accommodate an ALA-derived taxonomy and prompt content developers to select approved subject-identifying metadata elements. Some advanced content developers should be allowed to add items to the taxonomy.

The CMS should require content developers to complete a free-form description of the content, with a non-display option available to the content creator.

Metadata Management Features

The CMS must accommodate the creation and maintenance of multiple domain names, and must retain the display of variant domain names to the deepest levels of content.

Content developers should have the ability to create and maintain directory names of their content, as well as file names and “vanity” URLs (essentially redirects).

Content developers should have the ability to clone entire navigation/content elements, including all metadata elements, not just textual content.

Content developers should have the power to determine whether or not an item appears in navigation, as well as in automatically compiled links to content.

Content developers should have the ability to create and maintain a non-displaying “notes” field for their content.

Content developers should have the power to define the sort order of their content (alpha, date, custom).

Select content developers should have access via the CMS to style sheets that govern the display of their content.

Select content developers should have access via the CMS to templates that govern the display of their content.

The CMS must allow for the uploading/entering of associated files and links to related content. Content developers must have the ability to name the rubrics under which files and links appear.

The CMS must allow content to be restricted by member or product code to constituents of those groups (e.g., LITA members, Annual Conference registrants, etc.).

The CMS must provide content developers the ability to search navigation items and metadata for specific content elements (e.g., field searching and combined-field searching for metadata elements for retrieval of “lost” content).

The CMS should have the ability to maintain and display accurate crumb trail information.

Rights and Permissions

ALA’s content management system will be available to ALA staff and its members. ALA staff and members can be authenticated against our iMIS database. All ALA staff and members have a login and password stored in iMIS that is used with our Web site; the login and password should be the same for the CMS. ALA Web policy requires that all applications authenticate against our iMIS database, and after the first successful login, that all applications will share authenticated credentials such that the user log in only once per session.

Content developers will login with their iMIS/CMS credentials and gain access to their assigned areas. Authority to view restricted content should take into consideration the content developer's authority to publish content in the CMS. For example, content developers who are authorized to create content restricted to members of a unit, but who are not themselves members of that unit, should be able to see the content they have created. Likewise, someone who is not registered for the LITA National Forum (for example), but is authorized to create content restricted to registrants, should be able to view the content she or he has created.

The system must provide automated workflows that route content through an approval process, but also allows for editing at each level as outlined in the matrix below.

	Content Record Mgr	Navigation Item Mgr	Forms Mgr	Tagged Page Mgr	Tags Mgr	Template Mgr	Image Mgr	Archive Mgr	Redirect Mgr	User Mgr	Authority / Workflow Mgr
Create	Create and save content record	Create and save navigation item	Create and save forms	Create and save tagged page	Create and save dynamic tags	Create and save template	Create and save images and image maps	Create and save archive	Create and save redirect	Create and save CMS user	Create and save CMS authority levels and workflow functions
Edit	Edit content record	Edit navigation item	Edit forms	Edit tagged page	Edit dynamic tags	Edit template	Edit images and image maps	Edit archive	Edit redirect	Edit CMS user	Edit CMS authority levels and workflow functions

Revert	Revert to previous version of content record	Revert to previous version of navigation item	Revert to previous version of forms	Revert to previous version of tagged page	Revert to previous version of dynamic tags	Revert to previous version of template	Revert to previous version of images and image maps	Revert to previous version of archive	Revert to previous version of redirect	Revert to previous version of CMS user	Revert to previous version of CMS authority levels and workflow functions
Delete	Delete content record	Delete navigation item	Delete forms	Delete tagged page	Delete dynamic tags	Delete template	Delete images and image maps	Delete archive	Delete redirect	Delete CMS user	Delete CMS authority levels and workflow functions
Move	Move content record	Move navigation item	Move forms	Move tagged page		Move template	Move images and image maps	Move archive	Retarget Redirect		
Approve	Approve content record	Approve navigation item	Approve forms	Approve tagged page	Approve dynamic tags	Approve template	Approve images and image maps	Approve archive	Approve redirect		
Publish	Publish content record	Publish navigation item	Publish forms	Publish tagged page	Publish dynamic tags	Publish template	Publish images and image maps	Publish archive	Publish redirect	Activate CMS User	Activate CMS Authority Level
Batch	Batch maintenance of content records	Batch maintenance of navigation items	Batch maintenance of forms	Batch maintenance of tagged pages	Batch maintenance of dynamic tags	Batch maintenance of templates	Batch maintenance of images and image maps	Batch maintenance of archives	Batch maintenance of redirects	Batch maintenance of CMS users	Batch maintenance of CMS authority levels and workflow functions

See Appendix A for a Glossary of Terms

Administrative Guidelines

Deadlines

All work must be completed and ready for beta-test by January 31, 2006.

Testing

ALA's complex organizational structure necessitates a minimum of three weeks of testing for each deliverable. The beta-test will likely result in extensive feedback and changes for the Consultant, which should then be followed by acceptance testing..

A schedule for both beta and acceptance testing will be included in the final contract between ALA and the Consultant.

Training

Consultant should provide initial training for ALA staff, customized to ALA's needs based on the final specific product.

Consultant should also provide online help text/FAQs, manuals, etc.

Payment

ALA will pay invoices received from Consultant within 30 days from the date of receipt. If ALA disputes one or more items contained in an invoice, ALA would within 15 days of receipt of such invoice, notify Consultant of the item or items under dispute and the reasons for the dispute. Any undisputed amounts would be paid within the 30 days.

Out-of-pocket expense agreed to by ALA and Consultant will be reimbursed based on actual cost and upon receipt of documentation of expenses.

Work performed by Consultant outside of the Scope of Project and the contract must be approved in advance by ALA, in writing, after review of the rate for the work to be performed and the time frame for completion.

Legal

The laws of the State of Illinois will govern any Agreement or Contract with Consultant.

Consultant agrees to keep and maintain strictly confidential all data, information, and activities of ALA and/or its affiliated organizations, which may be revealed to Consultant during the course of work or contained in this RFP. Consultant also agrees to defend, indemnify and hold ALA harmless from any claim or action resulting from a breach of this confidentiality obligation.

To guarantee that the confidentiality of specific information contained in ALA's Online Communities, the Consultant must be willing to sign a confidentiality agreement.

In the performance of all work, Consultant is an independent contractor and will not be considered an employee or agent of ALA. As an independent contractor, Consultant is responsible for any and all employment related taxes and workers' compensation coverage. Consultant also agrees to accept liability for and will indemnify ALA against the payment of any and all contributions, assessments, rates and taxes, of whatsoever kind or nature, which might be imposed or attempted to be imposed upon ALA pertaining to the compensation paid or to be paid in connection with the services rendered to ALA, including but not limited to federal, state, county, city or other income, unemployment (FUTA), social security (FICA) taxes.

Proposed Work and Services

Submission of Proposal

Please submit an electronic copy of your response to the RFP by July 1, 2005, sent to rcarlson@ala.org, delewis@ala.org, and svanyek@ala.org . Quotations received after this date/time will not be considered.

COMPANY PROFILE

Please provide all of the following information about your company in your proposal.

1. Number of years in business.
 2. Primary and secondary business.
 3. Is the company wholly owned?
 4. Location/address.
 5. Hours.
 6. Total number of full-time employees.
 7. Individual(s) to be assigned this project.
 8. Experience level and biographies of individual(s) assigned to this project.
-
1. Describe your experience with and expertise in working with Web-based, database-driven Content Management Systems.
 2. Describe your experience with and expertise in working with iMIS, in particular with iMIS implementations that utilize extensive customization and take advantage of the majority of the available modules. Your work with the iBOs is also of interest to ALA.
 3. What differentiates your service from that of other providers?
 4. Furnish ALA with your company's service standards. Include your company's Service Level Contract(s) and Mission Statement.
 5. Describe the support provided during project implementation, including technical assistance, user manuals, instructional and/or educational materials, on-site visits, or other assistance.
 6. Describe how your company will solicit ALA feedback about your performance. Please provide sample(s) of your company's customer satisfaction surveys.
 7. What penalties and/or corrective measures are contractually memorialized for product defects that are discovered pre- and post-implementation?
 8. What penalties are contractually memorialized for situations where your company is not able to achieve agreed-upon quality standards?
 9. Describe any unique features that your company or software can offer that we should consider.

PROJECT WORK

Provide detailed implementation plans for the system you are proposing based on a project start date of September 1, 2005, and a "go live" date of November 30, 2005. Include detailed deliverables and ownership of those deliverables.

Provide an overview of the programs that will need to be written and how they will interact with existing software and systems to ensure the capabilities described in the Scope of Work section of this RFP.

1. Discuss your capability and experience with building systems that interface with the Core Business Applications and run on the Network outlined in the Technical Specifications section of this RFP. The iMIS system is the primary interface.
2. Will any third-party software (not currently in use by ALA as listed in the technical specifications) be required in the implementation of this system? Provide information on any third-party software requirement.
3. Describe how a registered member or non-member would be authenticated against iMIS.
4. Describe how a registered member would authenticate against iMIS using member types, dues, and subscription products for member-only areas of the site.
5. Describe the steps a member or non-member would go through to create an account.
6. Describe your proposed system's flexibility and available options for add-on capabilities (include adding variables to existing fields and creating new fields and query parameters, as well as modifiability in terms of adding new modules).
7. Describe your interface design features and options for integrating graphics/artwork, logos, and/or departmental designations.
8. Describe your company's communication methods for reporting technical problems to ALA IT and non-IT staff.
9. Describe your contingency plan for any situation in which a key member of your development team would become unavailable to continue work on this project.
10. Describe a plan for maintenance and enhancement of the ALA Content Management System.
11. Provide samples of standard and customized (if any) documentation, including tutorials, help files, and FAQs.

PROJECT ESTIMATE

Please provide quotes (cost, time, resources, or procedures as indicated) for each component of the project as described in the Scope of Project section above.

1. Cost of project (include hourly fees and time frame for project completion. If more than one person is involved, include fee schedule of each person).
2. Number of hours (broken down by implementation time line).
3. Number of personnel involved.

REFERENCES

Please provide complete information on three to five clients for whom you have implemented a content management system. Include the following information about each reference:

1. Name, address, contact, and telephone number.
2. Description of work performed and tenure of service to client.
3. Description of the level of complexity involved in integrating the client's Web application with their iMIS database.

Please include at least one reference from a not-for-profit organization.

Appendix A: Glossary

Archive: Older content no longer viable or necessary for display on ALA's Web site. Archived content is kept for historical purposes even while not in active use.

Authority: The ability to designate who may have access to what Administrative functions and in what capacity they may subsequently work within the system.

Content Developer: Any ALA staff or member who adds, edits, maintains, or delivers content to ALA's Web site. See Appendix B for profiles of ALA Content Developers.

Content Record: Every individual record of information created for publishing on the Web site constitutes a content record. Content records can be HTML pages, document files, uploads, links or any other specific item to be seen on the front end by a user.

Form: A piece of content created with various fields or attributes, into which a user may enter their own information. Data collected is stored in a database for later analysis or processing.

Images: Photos or drawings saved in one of a few standard formats. Images for ALA's use include .jpg, .gif, .swf, and a few others

Navigation Item: A navigation menu item acts as a "holder" for content displayed on the front-end. Settings that determine format, security, etc are specified while creating these menu items. Navigation menu items share a 'one-to-one' relationship with a navigation link at the front-end.

Redirect: A specially-created URL meant to work in the same manner as the original URL while appearing shorter or otherwise easier to remember

Style sheet: A series of commands which dictate how text and other content appear within a template or other page

Tags: Words or phrases that act as 'markers'/'flags' and are used to identify or mark a record. A record is 'tagged' to facilitate dynamic retrieval and display.

Tagged Page: Tagged pages are special pages created to pull content from other areas of the site and display them in specially created templates.

Template: Shell around which content is added to give a standard appearance. Templates are often applied to many related pages, such as all the pages belonging to a unit or a division.

User: Anyone with rights to make changes of any sort to ALA's Web site.

W3C: Stand for the “World Wide Web Consortium”. The W3C leads the world in developing specifications and guidelines for Internet technologies. ALA is committed to adhering to the W3C’s Level II guidelines for accessibility for all ALA Web content. Please visit <http://www.w3.org/TR/WCAG20/> for more information.

Workflow: Steps involved in the process of accomplishing a desired goal. Workflow is often necessary to define the steps for publishing content, archiving, approving, and performing other standard tasks.

Appendix B: CMS User Profiles

To say there is no typical ALA CMS user is quite an understatement. ALA's new CMS must accommodate a wide range of users, who have widely varying CMS needs and skill levels. In an attempt to give responders to this RFP a sample of this variety, we asked a cross-section of ALA staff to complete a questionnaire about their use of the current CMS. Their responses are found below.

1. How often do you use the CMS?

User A: I use CMS at least 25% of the time. I develop on line press kits which incorporate multiple links, graphic and audio files.

User B: I use the CMS once or twice daily, sometimes just for quick fixes (someone left a committee), but sometimes for major projects (setting up basic information for the LITA National Forum). Every week I publish job ads, so a good chunk of every Wednesday is spent in the CMS. I guess if you glommed it all together and smoothed it out, it would add up to about one day out of 5, or 20% of my time.

User C: Infrequently ... mostly if other staff are out and I must get a page up that day.

User D: At 7 hours a week on average, that would be 20 percent.

User E: I'd say only about ten percent of my week is spent working on the CMS.

User F: 50-75%. It is the main part of my duties

User G: I use CMS weekly. When I post releases, I use it for two to three hours. When I'm making corrections, I may use it for 1 to 2 hours/week.

Users H-L: The Production Services editors spend about 25% of their time in the CMS. We design and sometimes maintain sites for divisions, offices, initiatives, and programs, delivering artwork as HTML that ITTS converts to CF templates. We also design and maintain online publications, and work on graphic- or labor-intensive short-term projects for the divisions' web sites.

User M: I'm currently avoiding it like the plague and the time I use it is hard to gauge since if we had a better system it would be decidedly less. Ultimately though, it would probably be about 15-20% of my time.

User N: At least once a day, although it depends on the volume and flow of new or changed content. I mostly use it to "FTP" files created using Cold Fusion Studio editor or to add DOC or PDF files to a page.

User O: At least once a week, although it depends on the volume and flow of new or changed content. It's safe to say that I would use the CMS more if it did not take as much as 15 minutes just to change one linked content record.

User P: I do mainly administrative and troubleshooting work within the CMS on a daily basis. Probably at least 25% of my time is taken up by CMS work

2. What other software/tools do you use to accomplish your CMS work?

User A: I copy and paste content into CMS from existing Microsoft Word files that I have converted into text files (cleanest code). --

User B: Very little Cold Fusion. WordPad and Notepad, mostly. PDFs.

User C: Just Word

User D: Hotdog

User E: none.

User F: I do the code in Dreamweaver, some graphics in Firefox and MS Paint

User G: I copy and paste content into CMS from existing Microsoft Word files that I have converted into text files (cleanest code).

Users H-L: We use Photoshop to create artwork, Adobe ImageReady to construct navigation and art with basic java and animation, and Dreamweaver to create custom design layouts, CSS, interactive pages, and forms. We are also called upon to add content from Word, create and upload PDF files, upload movies and downloadable artwork in a variety of formats. We also export content from page layout programs (Quark, InDesign) as HTML for web publication.

User M: I use GoLive as needed. And Adobe Acrobat, Illustrator, Photoshop, etc.

User N: Used to use HotDog but it doesn't like XP or something ... Cold Fusion Studio editor suffices for most HTML formatting work. Also, Topstyle for handling .CSS aspects of formats. I've stopped using Fireworks and avoid image maps as much as possible for navigation banners. Corel Draw is preferred software for handling graphics and signage.

User O: I used Dreamweaver to design our pages, but at present our pages are simple enough to change by just using the provided HTML program, or occasionally copying an existing page and adding new content. We would like to experiment in the future with new content and new designs, and undoubtedly I would use Dreamweaver and Fireworks for that.

User P: I write cold fusion applications and build templates using CFStudio, and occasionally am called upon to copy content into the CMS from existing Microsoft Word files

3. Are you responsible for any tagged (dynamic) pages?

User A: Yes I also troubleshoot tagging issues for pages within PIO's section of the site.

User B: I manage one dynamic page, Information Technology and Library Systems.

User C: Probably, but I haven't been able to focus on which and how, etc.

User D: No.

User E: yes - the Washington Office press releases

User F: No

User G: Yes, for ALA News releases and sometimes for ACRL and ALSC.

Users H-L: Yes, but few. We currently oversee the PIO Press Releases page.

User M: I am not directly responsible for any tagged pages.

User N: Mostly off CMS, but there are some dynamic pages such as committee rosters.

User O: Our 14 most recent news stories are tagged to appear on our front page in chronological order. Further tagged news stories can appear by hitting the MORE NEWS STORIES link, theoretically all the way back to 1997. Our five most recent stories appear on the ALA News page.

User P: I manage the ALA Home Page, which is built dynamically, and answer questions/troubleshoot tagging issues for all the dynamic pages on the site

4. What is your technical skill level?

User A: I'm not sure what my technical skill level is. Most of the CMS functions I implement on a daily basis are self-taught, so I'm not up on the technical jargon. I'm not sure how to respond to this question.

User B: Quite high, though I'm more on the management side of things now. I was once a Unix system manager on a server that ran the ALA website, search engine, and list processor, as well as DNS and basic mail functions. I'm minimally competent in IIS,

despite finding it an abomination. I teach a basic "internet in libraries" class for Dominican's GSLIS, including basic HTML, and one of our member's sig file says, "To teach is to have learned twice." I have intermediate skills with Cold Fusion, strong HTML knowledge, good understanding of SQL and database management in general (though I don't find it all that useful in my daily use of the CMS), strong knowledge of web delivery of library content and services. I'm very well versed in usability and information architecture, and user-centered design, having recently redesigned a large public content site (World Book Online).

User C: Absolutely marginal

User D: HTML

User E: I have basic html skills (would be nice to hand-code sometimes, when things get tricky) and experience using Dreamweaver & CSS.

User F: My other skills (some design & CSS) are not used in this system.

User G: very basic

Users H-L: We possess advanced graphic design skills in Photoshop and Dreamweaver. Editors have a solid understanding of and ability to troubleshoot HTML code, although we work predominantly in a WYSIWYG environment. We have a working understanding of CF processes and are often called upon to design work for divisions and then turn that work over to ITTS, who adds the final level of functionality. We work as a middleman in this way between divisions and ITTS, and often provide input regarding content, organization, and feasibility.

User M: I have an intermediate understanding of HTML.

User N: I'll be getting the Advanced Cold Fusion course soon. Prolly ought to learn more about SQL as Intermediate level understanding of Access hasn't helped me understand the "math" of Cold Fusion coding. Advanced HTML user, measured experientially as well as with courses. Way back before there was an IT department, I used to pop a computer open and don't get me started!

User O: My HTML expertise is a few years old. With a new CMS system, I would get out of the rut of routine and bone up on XML, CSS, JavaScript, RSS, and other functionalities.

User P: I am a cold fusion developer, write SQL queries and do limited management of SQL databases via SQL Enterprise Manager, am comfortable with limited management of IIS and the Cold Fusion Application Server, have intermediate graphic design skills in Adobe Photoshop, and have a solid big-picture view of the integration of ALA's AMS into all of our major Web applications

5. What is your level of supervision over other content developers?

User A: I am one of three content developers in PIO. We don't really have a chain of command.

User B: LITA has a large stable of content managers, whom Michelle Frisque and I coordinate. All of them have authority to publish, but we are ever watchful and quick to fix content gone awry.

User C: I supervise a content developer who has full rights to publish, etc. I oversee-- and sometimes prepare the content.

User D: I do not supervise any other content developers.

User E: I supervise [User F's] updating of the web pages (but she has publishing authority). Carol does the bulk of the updating for the Washington Office.

User F: None

User G: I do not supervise any other content developers.

Users H-L: We do not supervise any other content developers.

User M: I do not supervise any other content developers.

User N: I facilitate about a half dozen member contributors. One staff member can also work on pages in a pinch (has her own password and CMS training).

User O: A few other people in our department serve as backup or supplementary CMSers. They all have publishing authority.

User P: I do not supervise any other content developers