

AccessEnable: A Tool for Evaluating Compliance with Accessibility Standards

Tom Brinck, Derren Hermann, Brian Minnebo, and Ali Hakim
Diamond Bullet Design and RetroAccess, Inc.

ACCESSENABLE

AccessEnable is an online tool developed at RetroAccess, Inc. for evaluating and improving the accessibility of web sites. AccessEnable is a commercial service available at www.retroaccess.com. It is organized around rulesets that contain guidelines. The principle rulesets are currently built around the Section 508 accessibility standards that apply to U.S. federal government web sites (section508.gov) and the W3C Web Content Accessibility Guidelines (w3c.org/wai/). Some principle aspects to AccessEnable include:

- It is designed around an efficient and comprehensive process to identify and correct accessibility problems to successfully and uniformly comply with standards.
- It supports both the testing aspects that can be automated and the process of correcting those that must be done manually.
- It is able to provide quick summaries that facilitate the manual evaluation of compliance, such as providing overviews of images used throughout the site and their corresponding attributes for review of consistency and completeness in the specification of those attributes.
- It supports the needs of a formal compliance process by incorporating exception-handling and formal documentation of the compliance procedure.
- The rulesets are easily extensible, enabling the tool to be rapidly adapted to other standards and to serve to implement internal corporate branding and style guides.

THE ROLE OF STANDARDS COMPLIANCE IN USABILITY

While standards compliance is obviously not adequate to ensure usability, or accessibility in particular, it is an essential tool to provide toward these ends. Standards are an important,

and often relatively inexpensive, first step toward assuring usability, and the currently accepted accessibility standards ensure a minimal level of accessibility that makes it **possible** for people with disabilities to access information, a necessary precursor to providing information in a **usable** format. Standards compliance can provide supplementary checks that might not be found in user testing, especially low-frequency or situational problems that wouldn't be encountered in the limited scenarios that can be considered in user testing, such as software compatibility issues or difficulties related to a low-frequency disability.

THE ARCHITECTURE OF ACCESSENABLE

AccessEnable is implemented in Perl and designed to be easily extensible by dynamically loading rulesets. By being provided online, new standards and updates to existing standards can be easily provided, as well as continuous enhancement of guideline explanations and tips. This saves the user from the trouble of installation, compatibility issues, and monitoring upgrades. The tool is also capable of evaluating and improving its own interface.

The code evaluation is achieved by spidering a web site and sending each page through a parser which matches the code against rules to identify problems. Record-keeping by the spider itself enables cross-site consistency checks and structural evaluation.

On individual pages, the parser must be robust--violations of HTML syntax standards are reported but they can't stop the progress of parsing or else the number of problems found would be highly limited. Some syntax problems are extremely common, such as the appearance of a less-than symbol, "<", within body text or the misuse of quotes. Both Netscape and

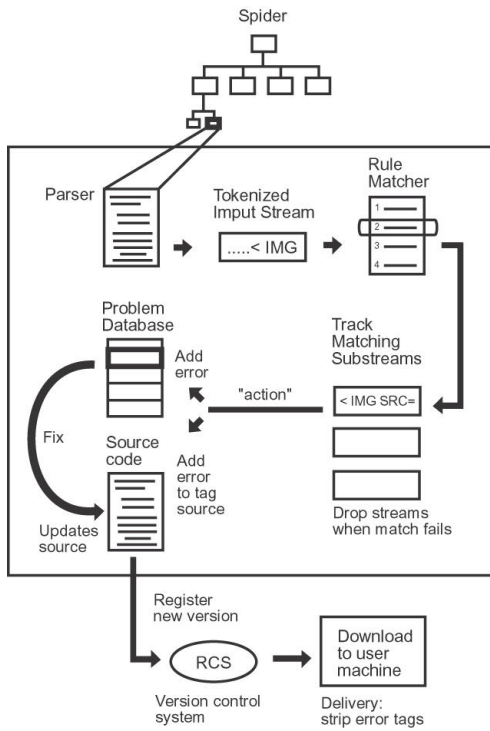
Internet Explorer are robust in their ability to display pages that violate standards, and while AccessEnable reports these problems, it continues to parse according to the interpretations that IE would make of the code. The alternative would be, when encountering an unmatched "<", to report the rest of the page as an error without further assistance.

Rules may check relationships across the entire site or on a single page. Each page-specific rule contains 6 procedures: initialization, start, stop, action, finish, and fix. When each page is loaded, the initialization procedures are called. As AccessEnable parses the code, it passes tokens to a rule matcher. When a match is made with the beginning of a rule template (the start procedure), that rule begins watching the input string for tokens that trigger a full match or a non-match (the stop procedure). When a full match is made, it triggers the action procedure, which typically adds an error marker to the input stream (tagging the source code) and adds an error to the error table. The error table is used to generate the primary report of problems found. When parsing the page finishes, the finish procedure is called.

A typical simple example is identification of a single tag. When the parser encounters an image tag, the start procedure of an image tag based rule, such as the requirement of ALT tags, is simultaneously matched with the stop procedure. This then checks that the ALT tag is present and records an error if it is not. The source code has an error attribute added at that point for easy location of the problem during fixing.

The initialization and finish procedures are used to handle whole-page rules, such as the requirement of a title tag or metatags.

When the user is reviewing the error report, correctable problems can be fixed. The fix procedure is called, and when necessary, the user is prompted for additional information (such as the text of an ALT tag). When a specific error is fixed, the file or files in which it occurs are parsed to locate the error tag and the correction is made. Each variation on the file is checked into RCS (a version control system) so that the user can easily recover from undesired changes. When the user downloads the final changes, they are checked out of RCS and any remaining error tags are stripped from the code.



The Architecture of AccessEnable

Dynamically-generated pages can be evaluated and repaired, but automatic correction of original scripts that generate the code requires integration with each and every relevant scripting tool, and it would never be possible to fix every kind of script. Thus, the process we support and recommend is that template pages of code be evaluated before integrating them into scripts. Beyond the template, the content of pages still needs to be evaluated, but fixes need to be repaired by hand. Database integration is a logical first step in expanding the tool to dynamic environments.

THE PROCESS OF USING ACCESSABLE

AccessEnable is designed to fit into the work process of web design and development. The tool would be used at an early stage of HTML creation when initial page templates are constructed and at a later stage when pages are nearly complete, usually as part of a quality assurance process. It would also be used during site maintenance (when only revised pages would normally be checked) and in repairing a site that had not originally been designed for compliance.

The process of checking standards and making corrections can be very time-consuming, so the tool is designed around optimizing that process. The process follows these steps:

1. Sign in and enter the address of your site (option to evaluate one page or the whole site).



2. The processing can take a while, so you have the option of coming back later for the complete report or reviewing the pages that are complete at any time.



3. You start at a summary page, which identifies the number of errors and warnings for each ruleset you are evaluating against.

Summary Report	Warnings	Errors	Total	Corrected
Section 508 Report	96	3	99	0
WCAG Report	101	7	108	0
Section 508 (Legacy) Report	8	18	26	0
Section 508 (Legacy) Report	37	0	37	0
Brink, Beigh, Wood Usability Report	8	0	8	0

Errors are problems we have detected that require correction.
 Warnings are potential problems we have detected. These items require your assessment as to whether a correction is needed.
 Offline Evaluation and Correction (This may apply to your site. Follow the "View Tip" links to view correction tips and learn how to address these items.

4. Then drilldown to review detailed information on the problems found. Alternate sorting of the error report is possible, but for practical use, the most common approach is to sort first by ruleset (e.g. the 508 guidelines) then by individual guideline, rather than by the order of occurrence of problems in the file. Since well-designed pages usually have few errors, this best fits the need of methodically verifying that each guideline is complied with. This also provides warnings of possible problems that can't be fully identified automatically. These warnings include information to facilitate the manual

identification of those problems.

Standard	Warnings	Errors	Total
Section 508	96	3	99
Total Corrected: 0 of 31			



5. We also provide review pages that facilitate manual inspection of certain problems that can't be automatically identified. These are visualizations designed to allow more rapid and consistent problem identification than code review would allow. (At the time of this writing, these are in prototype stage and not in the online product.)

6. If repair of the errors is done online, then there are several alternatives:

- Some detected errors or warnings can be "skipped" when they are not applicable to the site. AccessEnable provides the facility to document why these exceptions are being skipped.



- Some fixes can be done automatically. When applicable, the user can select to make these changes only to the current page being reviewed or to the entire site.
- Some fixes require more information, such as the text of an ALT tag. Individual fix procedures provide forms to elicit this information and make the fix.
- The remaining problems will need to be fixed by hand. Instruction is provided to facilitate manual correction.

7. Download the repaired site.



The following sessions are available to download:

Main session: www.section508.gov -- 02/02/02

Special note about Dynamic Pages

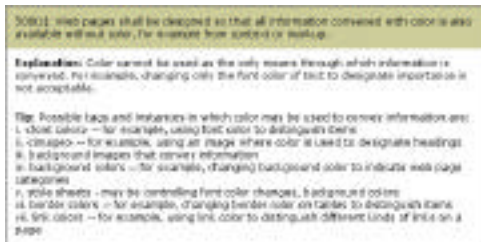
AccessEnable analyzes static HTML pages. If the content on any of the pages of your website is generated dynamically using CGI, ASP, PHP, CFM, etc., AccessEnable will be analyzing and correcting a static snapshot of the HTML produced NOT the source code.

Therefore, you **MUST NOT** replace any dynamic code with the corrected HTML files that you download from AccessEnable.

Other available sessions:

- o www.chriscassell.net -- 02/11/02
- o www.section508.gov -- 02/02/02
- o spiderman.diamondbullet.com -- 11/19/01
- o www.msn.com -- 10/25/01
- o www.videoculture.org -- 10/24/01
- o chriscassell.net -- 10/09/01
- o www.aafic.com -- 10/09/01
- o www.chriscassell.net -- 10/09/01
- o spiderman.diamondbullet.com -- 10/09/01
- o www.chriscassell.net -- 10/08/01
- o chriscassell.net -- 10/08/01
- o spiderman.diamondbullet.com -- 10/08/01
- o chriscassell.net -- 10/05/01
- o diamondbullet.com -- 10/05/01

8. Review guidelines that could not be automated. Fix any remaining problems by hand or with the appropriate authoring tool. If manual fixes are necessary, another online analysis of the site will be needed for regression testing.



9. The tool provides documentation of any remaining errors and any justifications that have been provided.



Home - AccessEnable - Consulting Services - About Us - Resources - Contact Us - Us Today - Logout

Section 508 Summary

The error numbered 29 on page _index.html has been ignored with attached note: "Logical tab order"

The error numbered 63 on page _index.html has been ignored with attached note: "Logical tab order"

The error numbered 69 on page _index.html has been ignored with attached note: "Logical tab order"

The error numbered 72 on page _index.html has been ignored with attached note: "Logical tab order"

The error numbered 77 on page _index.html has been ignored with attached note: "Logical tab order"

The tabindex attribute for the form element tag in line 256 on page www.section508.gov/index.html has been changed to: 12

The tabindex attribute for the form element tag in line 254 on page www.section508.gov/index.html has been changed to: 13

The tabindex attribute for the form element tag in line 267 on page www.section508.gov/index.html has been changed to: 14

Section 508 Summary

The main goals of this process are:

- to provide a rapid end-to-end process, not simply identifying problems, but taking people completely through to documented compliance.
- to treat non-automated guideline checking and repair as an essential part of the overall process.
- to provide clear documentation appropriate to meeting legal standards.
- to provide reports that support the typical types of users: summary views for status checking by management and "auditors", and detailed support for quality assurance and software development personnel.

THE REQUIREMENTS FOR A GUIDELINES-BASED AUTOMATED EVALUATION TOOL

Based on our experience, the following is a summary of the types of requirements a guidelines-based tool needs to support.

Assessment and Reporting

Does it help you focus on the compliance process without distraction by other guidelines (e.g. focus on Section 508 only, if that's your need)? Are reports meaningful, clear, and functional? Does it assist you in manually assessing violations that can't be done automatically? Does it provide appropriate compliance documentation?

Precision & Completeness

Does it find everything that can be automated (this is a matter of degree, so some way of evaluating the completeness of alternative tools would be valuable)? Does it work with every web technology and across your whole website? Is it extensible for future guidelines?

Correction

Does it automatically correct errors? Can it correct an error everywhere across a site at once? Does it allow interactive correction, facilitating semi-automatic fixes? Does it remember correction decisions from one use to the next?

Usability

Is it easy to use, accessible, and designed around the work process of a QA specialist, developer, or auditor? Does it provide detailed embedded help and tips for interpreting its results.

Customization and Flexibility

Can custom guidelines be easily constructed?
Are there options that help you specifically address site-specific problems and corrections?
Does it enable you to turn off rules and to document why a rule was not applicable?

Technical Completeness

Is the tool fast and reliable? Does it handle secure websites (when passwords are provided)? Does it integrate with dynamically-generated sites or with other web development tools? Can it spider an entire site and can it be restricted to a subsite or to omit certain pages?

Tom Brinck is a co-founder and officer at both Diamond Bullet Design, a web design and usability consulting company, and RetroAccess, which provides an online service for automatically evaluating and repairing the accessibility and usability of web sites (www.retroaccess.com). He is an adjunct lecturer at the University of Michigan, where he teaches user interface design and web usability. Tom is the principal author of the book "Usability for the Web: Designing Web Sites That Work". He has worked at Apple Computer, Toshiba, and Bellcore, where he's done research on user interfaces for education, speech interfaces, and groupware.