Chapter 5

Evaluating Web Sites

Often, developers are faced with upgrading an existing Web site rather than starting from scratch. Being able to fully evaluate the execution of a Web site is an important skill that all developers should strive to master. Site evaluation is also a great way to learn from others. Looking at sites that are well executed may inspire designers, while evaluating those that are broken may show them how to avoid errors. Yet site evaluations are not always easy to conduct. Often, developers focus on what they are familiar with or focus only on surface aspects of sites, such as visual design. As in building a site, an evaluation of a site must focus not only on visuals but also on technology, content, purpose, and delivery. Even when keeping all aspects of Web design in mind, a developer looking at a site may not understand either the initial design considerations or the decisions made that result in what is being evaluated. In this sense, evaluators may have to act as archeologists and try to uncover deeper meaning from basic site characteristics.

The primary method for site evaluation we present in this chapter is often termed *expert evaluation*. The goal is to study a site as informed developers and try to find common execution and usability problems. However, the problem with this type of site evaluation is that developers may not think like users and may assume that things are usable when they are not. Expert evaluation is simply no substitute for real user interviews and testing. Yet don't quickly dismiss expert analysis in favor of usability studies. User testing does little to uncover execution flaws, so we should make sure that sites pass the execution part of our evaluation first before wasting valuable user testing time. Further, many common usability problems are easily observable and user testing simply verifies what a skilled developer may already know to be true through experience. Given these considerations, we will proceed with an overview of expert evaluation first, followed by a discussion of conducting user testing.

The Goals of Expert Evaluation

There are two goals when conducting expert evaluations of Web sites. The first is to uncover obvious execution flaws with sites, such as poor HTML markup, error prone JavaScript, broken links, and other problems (which should be caught during quality assurance but often are not). The second goal is to find obvious usability problems with a site before conducting user testing.

While the use of quality assurance tools and practical knowledge of the various aspects of the Web medium will help us find execution gaffes, usability problems can be more difficult to ferret out. We need to be mindful of how users think when conducting this part of testing. We need to be particularly careful when making assumptions about purpose, audience, creation method, and so on. If these assumptions are incorrect, the associated conclusions could be equally incorrect.

Conducting an Evaluation

When starting an evaluation, it is important to stop and record some basic information. For example, note the URL of the site you are to evaluate, the date, the time, the person conducting the evaluation, and the reason for the evaluation. When you begin the evaluation, you should block out some time to do the evaluation continuously; otherwise, your impressions could be adversely affected. Consider recording your end time to get an idea of how long it took to reach your conclusions. In general, the evaluation will be broken into the following steps:

1. First impression
2. Home page pretesting
3. Sub-page pretesting
4. Navigation pretesting
5. Task analysis
6. Execution Analysis
7. Final Impression

When we have finished with the evaluation, any required supplementary materials should be prepared, and an evaluation summary developed. Appendix B provides a sample form for conducting a site evaluation. Reading the following sections will help you understand the motivation for the various tests and how to conduct them.

First Impression

The first thing to do before you start the detailed evaluation is to stop and write down your first reaction to the site's home page. Just load the home page and look at it for at most five to ten seconds, and write down whatever comes to mind. Ideally, you will not be too familiar with the site, so the first impression will not be tainted. (Be sure to clear your browser's temporary files and cookies to make certain that your results are not skewed by the site already being cached.) If you are very familiar with the site, you might want to get a few other people, show them the site, and ask what they think on a scale from 1 to 5 (where 1 is a negative feeling and 5 is positive). The point here is to gauge a user's initial feeling for a site—remember, people aren't always rational. Unfortunately, a first impression is only just that if it is truly the first time you are looking at a site. Don't discount this part of the test. Even though a first impression may be an emotional reaction heavily influenced by visuals or environment considerations, record it and try to understand what causes your feeling. If users coming to a site have a very positive or negative first impression, it could certainly affect their desire to go further.

Home Page Pretests

The first few pretests conducted will give you a basic sense of the usability of the home page. Some of the pretests will require you to make some logical assumptions that you will later verify to show usability of the site, so don't start using the site yet or you'll spoil this part of the evaluation. Just keep the home page onscreen and your hands off the mouse and keyboard.

Identity Pretest

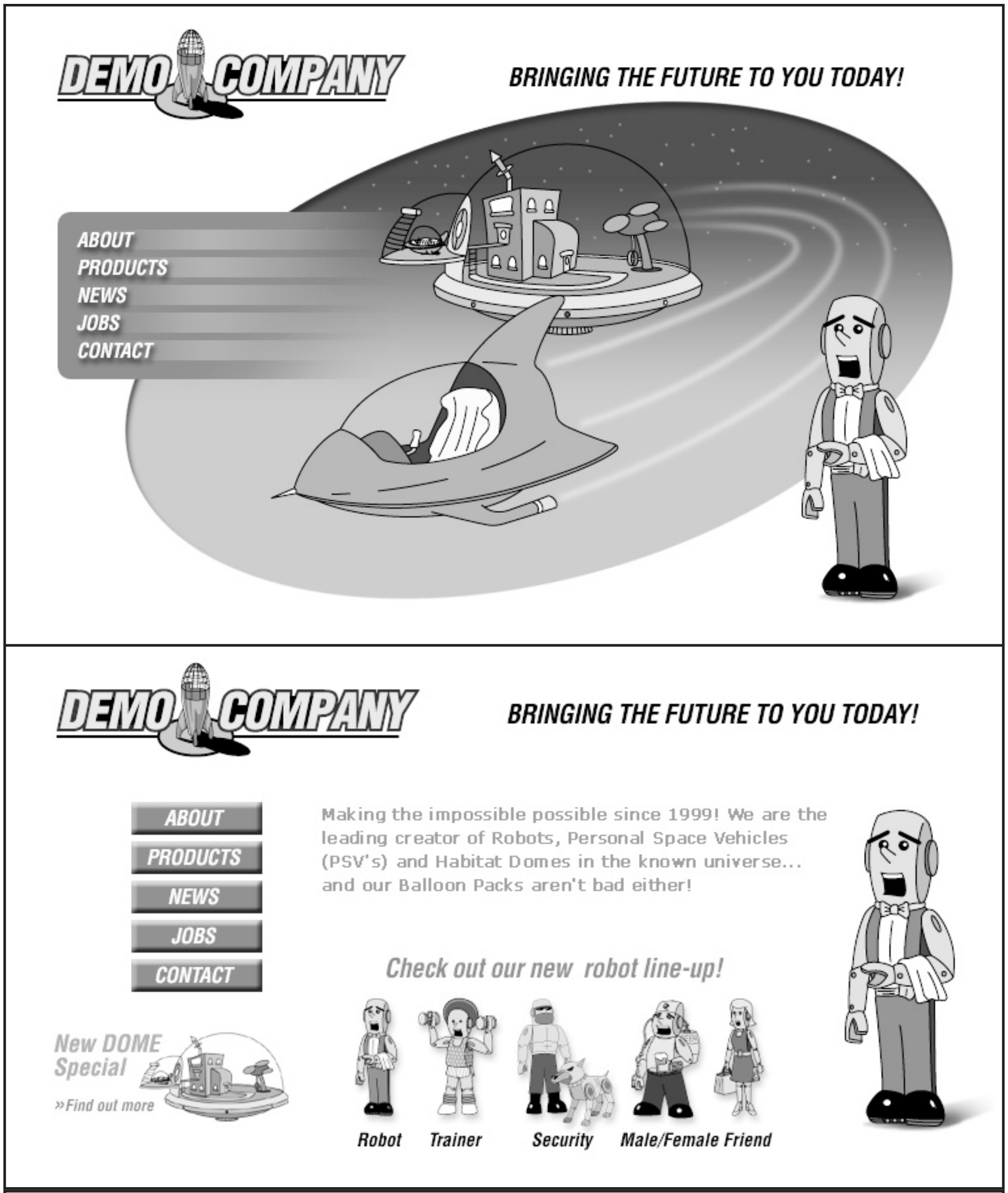
The first pretest to be conducted could be called the identity test. To conduct this test, look at the home page for between 30 seconds to a minute, and see if you can figure out the organization's name, the topic of the home page, and any sense of what the site is about. It would seem obvious that a site should clearly communicate its goals and purpose right away, but often that just isn't the case. Consider the two home pages in Figure 5-1—which passes the home page identity test for you?

Now ask yourself what users are supposed to accomplish at the site. More important—who is the site actually built for? For some sites—particularly those that you may not have much involvement in—performing a site evaluation may be much like an archaeologist looking at an ancient civilization's ruins. The purpose, use, and users of a particular aspect of a Web site will be almost as difficult to discern by a site evaluator as the significance of a few stones from a larger structure by an archaeologist.

Navigation Pretests

The next and probably the most telling is the navigation pretest. In this test, before you use the site, look at the home page and attempt to guess which areas of the screen are clickable. You may consider printing the page and circling the hot spots, conducting what is called a *paper test*. However, given that many pages may not be designed for printing or will remove navigation features in print, it is best just to do a screen test and run your finger, not the mouse, around the screen trying to determine if something is clickable or not. Once you have evaluated the whole page, go back and check your intuition. You will probably find that some clickable areas of the page do not obviously look like they are for purposes of navigation, while other things that *look* clickable actually aren't. Common reasons for failure include inconsistent color usage such as using blue text for labels and logos, removing underlines on links, and trying to make images and supporting materials link together. Note the number of believed links and actual links, determine an accuracy ratio, and record any notable problems for your final report.

The second navigation pretest requires determining the purpose of each clickable zone on the page. Once the links have been identified, record each and write a brief statement about what will happen when the link is pressed. Once finished, check your record by visiting each link and noting whether your guess was correct or not. Surprisingly, this test fails quite often because of poor labels. Often, failed link labels use a metaphor, jargon, or acronym, so make sure that your wording is plain and simple.



Making the impossible possible since 1999! We are the leading creator of Robots, Personal Space Vehicles (PSV's) and Habitat Domes in the known universe... and our Balloon Packs aren't bad either!

What do they do?

Figure 5-1. *Looks can be deceiving*

Once finished with these basic tests, you might want to scan link labels for style and consistency. Make sure that the labels are of similar length, wording, and style, both textually and visually. Observe the rules of the page for what is clickable and what is not, and note any inconsistency in visual style in clickable regions, regardless of whether any such region passed the initial clickable pretest.

Sub-Page Pretests

The primary sub-pages of the site—namely, those that are directly accessible from the home page—should be tested using the same pretests described in the previous two sections. However, for the identity pretest, focus more on the purpose of the page than on the organization. The navigation pretests should proceed normally. While this may seem like a lot of work for an average size site, it should proceed rather quickly if the sub-pages follow a consistent design and navigation pattern. If they do vary greatly, you are probably facing a site that has a high degree of design and navigation inconsistency and deserves significant analysis.

Site Navigation Testing

Once the first layers of the site have been examined, it is time to perform simple tests to probe the quality of the global site navigation. Good sites will provide consistent, well-executed navigation and should provide alternative navigation schemes, such as site maps, indexes, and search engines. First, look to make sure that placement of navigation is consistent from page to page. Subtle shifting may occur, so try browsing the site extremely fast and notice whether the menu items bounce or jump position slightly from page to page. Even this minor variation can break the perceived stability of a site. Next, look to see how robust the navigation is and whether multiple forms of site navigation are supported. Numerous navigation execution questions should be asked during this phase. Is the current location clearly indicated with labels or link path indicators? Does the site have text links at the bottoms of pages? Is alternative text used for graphical navigation buttons? Does the site require excessive scrolling? Are back-to-top links used on longer pages? Does the site have a map or index? The questionnaire in Appendix B presents many of the questions you should be asking during the navigation analysis phase.

One form of navigation that deserves special attention, if present, is the search facility. Very often, search is poorly implemented in a site, despite the fact that more and more users are coming to rely on it. Chapter 9 presents a thorough discussion of how search should be implemented in a site; but for now, focus on how the search is accessed, how it deals with errors, and how both positive and negative results are presented. Search facilities should be clearly marked and easily accessible from every page. A well-implemented search should correct errors or at least clearly indicate them when they occur. Once a positive query is returned, the results should be easy to navigate and refine. All these issues are covered in the sample evaluation; but if you evaluate sites on your own, make sure to enter nonsense queries and “extreme positive” queries, like the organization name, in the search field, to see how the extreme cases are handled.

Task Analysis

The testing so far has concentrated on general navigation of a site, but the goal of navigation is to help a user accomplish some task. Generally, on the Web, users are doing one of three general tasks:

1. Reading
2. Looking for something
3. Performing some interaction

The third task covers user activities like interacting with menus, filling out forms, or other mechanical tasks. Our testing should make sure that the site supports all three of these general task groups. Once we have verified that, we should consider the specific tasks unique to a particular site.

Testing Readability

When thinking about reading Web content, you have to consider both when and how the user will read the content. A user may read content immediately, may print it to read offline, or may bookmark it to read or print at a later date. Web content should be readable both onscreen and on paper.

Testing printing is easy: just print each page in the site. Be careful, though; some pages may purposely not be designed for printing. Also, you may have special print buttons or Adobe Acrobat files for printing. If this is the case, make sure to note the approach and whether it is effective.

Testing the screen readability of content is a little more difficult. Of course, reading content is the best test, but it tends to take a long time. You will almost certainly find, as you perform this test, that content is too long or complex to be easily read onscreen. Even when content is written for screen use, page layout and contrast may make it difficult to read. One way to test page layouts and contrast is to perform what the author dubs the “fuzzy eye” test. In this test, squint and look at the page. If you can still discern the general sense of the page structure easily, the layout and contrast is probably adequate; if you cannot, the items may be too close together or contrast may not be strong enough.

Testing Findability

Of course, information is only useful if site visitors can find it. In order to test the findability of information in a site, you first need to have at least some familiarity with the content in the site before attempting to find an item likely to be there. The simplest findability test would be to look for something required in just about any site—for example, contact information. Once a generic item has been determined, try to find the information from an arbitrary point in the site. You may find that even this test requires numerous clicks once beyond the home page. You can also try the same test using the site’s secondary navigation facilities, such as the site map and search facility.

The other findability tests are similar to the simple one just described, but they require that you find a particular item that is very specific to the site. For example, if products are sold, try to find the price of a particular product, the cheapest product, and the most expensive product. If the organization is a corporation, try to find information about the management team or, if it is publicly traded, its current stock price or last reported revenue figure. There are many possible information tasks, and you may want to record not only whether the task was successful or not, but also the time it takes or the number of clicks required to find something—as well as your feelings about the ease of use and adequacy of results.

Testing Interactivity

The final task-related test concerns the various interactive features of the site. This testing is primarily related to filling out forms for performing tasks such as ordering products, making contacts, creating memberships, and so on. Each primary feature of the site should be tested in three ways: correct usage, extreme negative, and extreme positive. Correct usage means following the steps—filling out a form and so on—to buy a product in the basic, obviously correct manner. You may find that it is difficult to figure out what to do during this test. If so, make sure to note down frustrations. Extreme negative and extreme positive tests make mistakes on purpose during interactive tasks. In extreme negative testing, obviously false or blank answers are provided to see if the site handles these properly. Extreme positive testing goes in the opposite direction and tests for out-of-range values and things that would be obviously beyond the capacity of the site. Well-designed sites should limit errors, so, ideally, interactive tests will cause frustration rather than raise execution issues. Unfortunately, given the state of Web development procedures (as discussed in Chapter 4), many execution errors may exist in tested sites. We will discuss a few things to look for in the next section.

Execution Analysis

Execution testing focuses on trying to make sure the site is built correctly. Execution includes issues with content, visuals, technology, and delivery. For example, with content, you might look to see if site content is up-to-date or if there are spelling and grammar errors in pages. Technical execution would focus on whether the site follows standards for HTML, CSS, XML, and other technologies. Visual execution would be concerned with image quality and file size. Delivery would be focused on speed and server capacity. The next few sections detail a few of the things to keep in mind as you evaluate each area, with the Appendix B checklist providing a set of specific questions to try to answer.

Content Execution

The quality of a site is heavily influenced by the freshness and quality of the content presented. A site's content should be appropriate in quantity—not too much that it is difficult to find appropriate information easily, but not so little that the user is left wanting more. The content should also be up-to-date and accurate. Execution issues, such as spelling, grammar, and tone, should also be well considered. Last, the details of the site should be very carefully examined. Truly, with Web sites, the devil is in the details. Copyright dates, trademarks, product names, and very small formatting errors are often glaring to the user and may ruin an otherwise excellent experience.

A good way to evaluate content is to do a careful screen and paper walk-through. Printing pages and going over each one very carefully is probably the best way to find typos and consistency issues. However, many Web maintenance tools and even page editors can be used to spell-check pages. When looking for details, it is tough to spot everything; fortunately, some Web site maintenance tools can be used to evaluate consistency of terminology through the use of custom rules that look for the inclusion of certain key phrases.

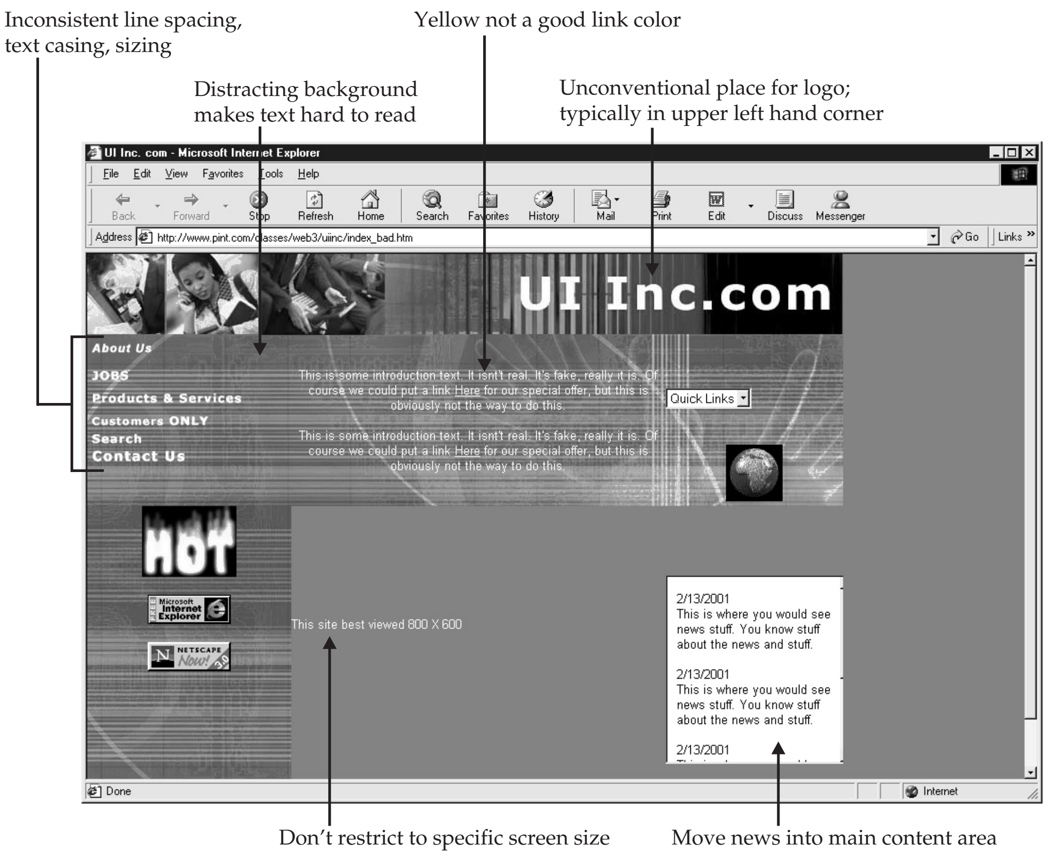
Visual Execution

Evaluating the look and feel of a site can be difficult because doing so is, to a great degree, a matter of personal taste. However, execution of images and layout should be evaluated regardless of your personal take on a site's aesthetics. Images may not be used properly or optimized correctly. There may be color problems in the site, font sizing issues, and page layout problems. In many cases, the page layout may not even fit the screen resolution or print correctly. Pay particular attention to tests of the site under less than ideal conditions, such as lower resolution. In many cases, a site layout will completely fall apart when images are turned off or font sizes modified. When doing the visual portion of a site evaluation, it is important to print out a screen capture of the evaluated page, as it may change over time. Screen printouts can be marked up to draw attention to problem areas as well as interesting features. Figure 5-2 shows an example of a marked-up page with visual and navigation execution notes.

Technical Execution

Web design relies heavily on technology, ranging from simple markup languages to complex programming approaches. When evaluating a site you have full access to, it is possible not only to look at client-side technologies, such as HTML, but also to examine server-side technologies, such as CGI programs or databases. Unfortunately, when examining sites externally, you may be limited to looking only at technology easily viewed at the browser or the effect of technology executed on a server. For some evaluators, it may be appropriate to call in a professional programmer to evaluate the quality of examined code, as glaring errors may escape those who know CGI or JavaScript only just enough to use provided scripts. We overview a few of the more common technologies here for evaluation and leave the rest for Appendix B.

HTML/XHTML Because HTML serves as the bedrock of a Web site, particular attention should be paid to the accuracy and quality of HTML. With the rise of XHTML, use of the doctype indicator and strict compliance are becoming particularly critical. Compliance with the various HTML or XHTML standards should be examined by validating key pages in the site. Online validators, such as http://validator.w3.org, can be used, but readers may



Inconsistent line spacing, text casing, sizing

Yellow not a good link color

Distracting background makes text hard to read

Unconventional place for logo; typically in upper left hand corner

UI Inc. com - Microsoft Internet Explorer

Address http://www.pint.com/classes/web3/uiinc/index_bad.htm

This is some introduction text. It isn't real. It's fake, really it is. Of course we could put a link Here for our special offer, but this is obviously not the way to do this.

This is some introduction text. It isn't real. It's fake, really it is. Of course we could put a link Here for our special offer, but this is obviously not the way to do this.

This site best viewed 800 X 600

2/13/2001
This is where you would see news stuff. You know stuff about the news and stuff.

2/13/2001
This is where you would see news stuff. You know stuff about the news and stuff.

2/13/2001

Don't restrict to specific screen size

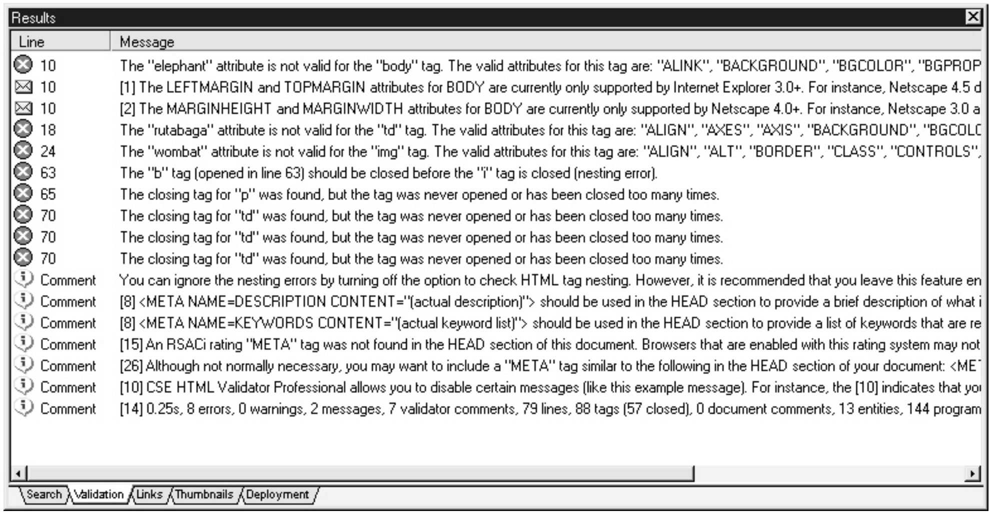
Move news into main content area

Figure 5-2. *Printed page marked up*

find stand alone validation tools like CSE Validator (http://www.htmlvalidator.com) to be superior. Figure 5-3 shows this validator in action.

Proprietary tag usage or trick HTML should be carefully noted. Inspection **<meta>** tags, comments, and other small signs such as consistent page formats should be noted to help determine how HTML was created—such as with a tool or by hand. In some cases, telltale signs like indentation patterns of markup may indicate creation by a particular HTML editor; but if it is possible to directly query the developer, ask which tools were used and what standards were followed if any.

CSS Cascading Style Sheets are rapidly becoming an important technology for presenting Web pages. CSS use provides a major benefit in allowing separation of document structure from presentation. However, unless the site uses external style sheets, this benefit is reduced. Document-wide style sheets or inline styles are adequate,

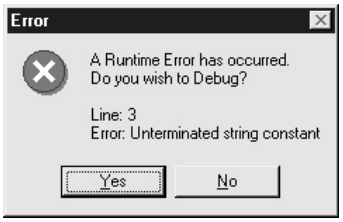


```
Results
Line      Message
10        The "elephant" attribute is not valid for the "body" tag. The valid attributes for this tag are: "ALINK", "BACKGROUND", "BGCOLOR", "BGPROP
10        [1] The LEFTMARGIN and TOPMARGIN attributes for BODY are currently only supported by Internet Explorer 3.0+. For instance, Netscape 4.5 d
10        [2] The MARGINHEIGHT and MARGINWIDTH attributes for BODY are currently only supported by Netscape 4.0+. For instance, Netscape 3.0 a
18        The "rutabaga" attribute is not valid for the "td" tag. The valid attributes for this tag are: "ALIGN", "AXES", "AXIS", "BACKGROUND", "BGCOLC
24        The "wombat" attribute is not valid for the "img" tag. The valid attributes for this tag are: "ALIGN", "ALT", "BORDER", "CLASS", "CONTROLS",
63        The "b" tag (opened in line 63) should be closed before the "i" tag is closed (nesting error).
65        The closing tag for "p" was found, but the tag was never opened or has been closed too many times.
70        The closing tag for "td" was found, but the tag was never opened or has been closed too many times.
70        The closing tag for "td" was found, but the tag was never opened or has been closed too many times.
70        The closing tag for "td" was found, but the tag was never opened or has been closed too many times.
Comment   You can ignore the nesting errors by turning off the option to check HTML tag nesting. However, it is recommended that you leave this feature en
Comment   [8] <META NAME=DESCRIPTION CONTENT="(actual description)"> should be used in the HEAD section to provide a brief description of what i
Comment   [8] <META NAME=KEYWORDS CONTENT="(actual keyword list)"> should be used in the HEAD section to provide a list of keywords that are re
Comment   [15] An RSACi rating "META" tag was not found in the HEAD section of this document. Browsers that are enabled with this rating system may not
Comment   [26] Although not normally necessary, you may want to include a "META" tag similar to the following in the HEAD section of your document: <ME
Comment   [10] CSE HTML Validator Professional allows you to disable certain messages (like this example message). For instance, the [10] indicates that yo
Comment   [14] 0.25s, 8 errors, 0 warnings, 2 messages, 7 validator comments, 79 lines, 88 tags (57 closed), 0 document comments, 13 entities, 144 program
```

Figure 5-3. *Example of HTML validation*

but their use should be considered less than ideal. Regardless of the method of including style rules, extreme care must be taken with CSS because of all the browser bugs and rendering differences. Compliance with the CSS1 and CSS2 standards may not be as important as making sure the various CSS properties work under common browsers. However, a CSS checker, such as http://jigsaw.w3.org/css-validator/, should be used. Close attention should also be paid to the types of rules used and whether or not there is any problem with browsers that do not support CSS. Testing with an older browser or with the CSS facilities turned off should be performed.

JavaScript JavaScript is a very important part of many Web pages, but far too often it is not used in a reliable manner. Well-executed JavaScript-laden pages will employ the **<noscript>** tag to address scripting being turned off, and may even restrict usage without script enabled. Scripts also should be able to address browser incompatibilities and should not throw error messages like the one shown here:

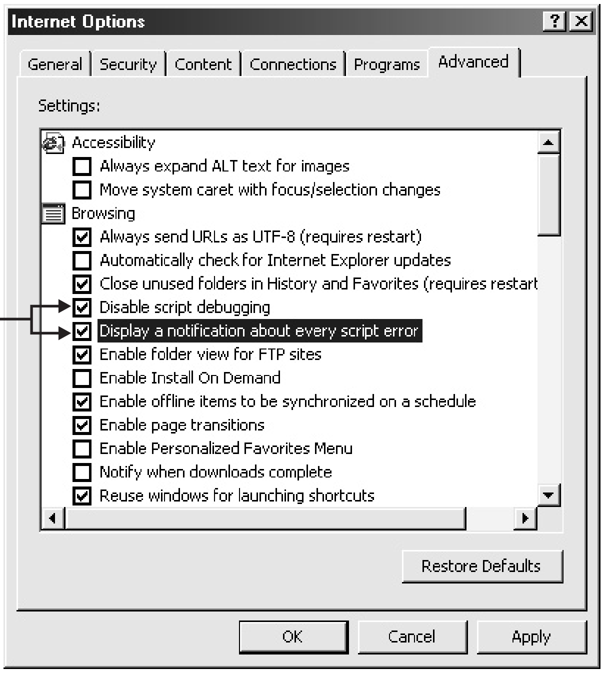


Error

A Runtime Error has occurred.
Do you wish to Debug?

Line: 3
Error: Unterminated string constant

Fortunately for Web site visitors, most browsers are shipped with the default to turn off JavaScript error notifications, since otherwise you would probably see a great number of them. Set your browser's preferences to show errors, as shown here in Internet Explorer.



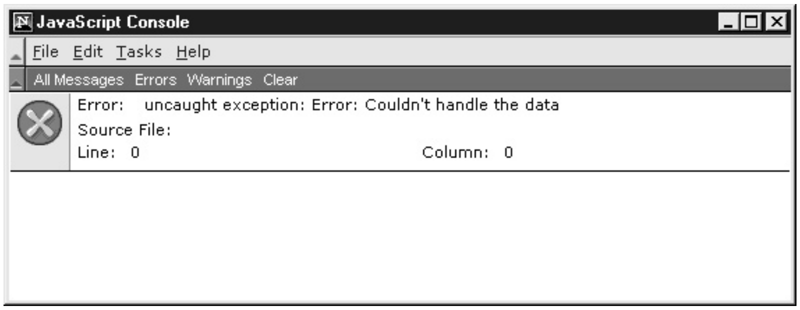
Settings:

Accessibility
- ☐ Always expand ALT text for images
- ☐ Move system caret with focus/selection changes

Browsing
- ☑ Always send URLs as UTF-8 (requires restart)
- ☐ Automatically check for Internet Explorer updates
- ☑ Close unused folders in History and Favorites (requires restart
- ☑ Disable script debugging
- ☑ Display a notification about every script error
- ☑ Enable folder view for FTP sites
- ☐ Enable Install On Demand
- ☑ Enable offline items to be synchronized on a schedule
- ☑ Enable page transitions
- ☐ Enable Personalized Favorites Menu
- ☐ Notify when downloads complete
- ☑ Reuse windows for launching shortcuts

To show errors, make sure the Disable Script Debugging check box is unchecked and Display Error Notification is checked.

In Netscape, you should check the JavaScript console for the error message shown here:



```
Error:   uncaught exception: Error: Couldn't handle the data
Source File:
Line:  0                                    Column:  0
```

Cookies For many, the use of cookies is an invasion of personal privacy. The reality is that cookies are very useful to get around programming limitations caused mainly by HTTP protocol limitations. However, regardless of your personal take on cookies, it is important to know whether a site uses cookies and what they are used for. Some sites may even issue multiple cookies per visit, each with a different purpose. Careful inspection of cookie data can yield valuable clues to how a site works. If cookies are used, it is important to verify the site still works with cookies off. Also, if cookies are used, a statement indicating what they are used for should be available on the site.

Browser Support Probably the most well-known aspect of site testing is browser support. Many site testing protocols simply advise designers to test in as many browsers as possible. The reality is that you should attempt to create a matrix of the various browsers and perform the technology and layout tests within each browser individually. Oddly, you may find that there are subtle rendering differences in each browser, as well as numerous bugs. A large matrix showing all the different versions of each browser and operating system is the best way to conduct a browser test. Unfortunately, you may find that there are literally dozens of versions of just the 4.*x* generation of Netscape. Because of the difficulty of testing so many combinations, you may want to focus on those browsers that are known to use your site. In some cases, such as with an intranet, the browser being used may be obvious; but before guessing what browsers a site's users commonly use, consider accessing the log files to make sure.

Delivery Execution

How the site is delivered is extremely important to understanding the site's usability. Users appreciate fast downloads, but, as will be discussed in Chapter 17, speed of delivery is often influenced by many factors beyond the size of files being delivered. It is important to understand the server resources used to deliver a site, including both hardware and software used. It is also important to understand how the site is hosted. How the site eventually connects to the Internet can impact performance greatly. Using even simple network tools like "ping," it is possible to determine the responsiveness of a server. Many operating systems provide this tool; for example, under Windows, access the DOS prompt and type **ping** and a host name. If you typed **ping www.webdesignref.com**, you might see something like this:

```
C:\WINDOWS>ping www.webdesignref.com

Pinging www.webdesignref.com [66.45.42.235] with 32 bytes of data:

Reply from 66.45.42.235: bytes=32 time=32ms TTL=114
Reply from 66.45.42.235: bytes=32 time=66ms TTL=114
Reply from 66.45.42.235: bytes=32 time=27ms TTL=114
Reply from 66.45.42.235: bytes=32 time=95ms TTL=114

Ping statistics for 66.45.42.235:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 27ms, Maximum = 95ms, Average = 55ms
```

The round-trip time of data can be used to get a general sense of the responsiveness of the server. It is also possible to acquire other server and network information using tools like WHOIS, traceroute, nslookup, and others. On Windows, most of these tools are included in the operating system, can be found in the public domain, or are nicely packaged in network tools like WS_Ping ProPack (http://www.ipswitch.com/).

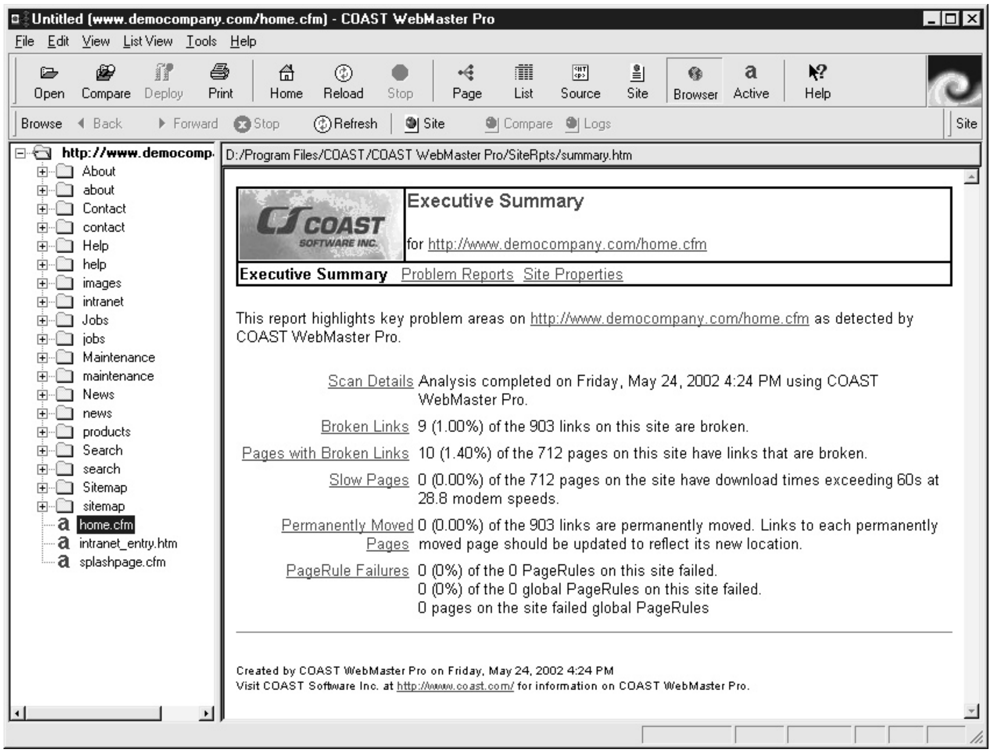
After server and network issues, the size of the pages delivered should be considered. Most site analysis tools will identify pages that are considered large. You can set the threshold for what is considered large, byte-wise, in most of the programs, but some consider anything over 30–50K (including any graphics in the page) as a large page, despite the rising popularity of faster Internet access. Theoretical download times under a variety of line speeds can also be determined with a site analysis tool, and most Web page editors like Dreamweaver even provide facilities to determine page weight and download speed. However, do not rely solely on theoretical times; test the site under actual conditions, if possible. Since network conditions are always changing site delivery, test results may vary greatly from moment to moment.

The Final Question

Now that you have evaluated many aspects of a site, consider what you would give the site as a final score. You don’t have to be very scientific about your final rating. Given how much you know now about the site, do you think it is a great site or not? Were you able to accomplish the tests easily? Would you take away a positive, neutral, or negative feeling about the site? Consider listing a few of the reasons that made you skew one way or another.

Evaluation Reports

After finishing your evaluation, you should put together a report summarizing your findings. Make sure to illustrate your findings with as many frame grabs and diagrams as possible. Also, try to provide as many specific details as possible, as well as indications of where the errors are in the site and how they might be fixed. Complete reports should include a detailed analysis of a site, including the number of pages, the page weights, broken links, technology usage, and so on. Because of the tedious nature of compiling such information, we leave this part of the evaluation to tools. Consider using a maintenance or quality assurance tool to analyze the basic characteristics of the site. Quality maintenance tools such as Coast Webmaster (http://www.coast.com) can produce high-quality reports like the one shown in Figure 5-4. However, do not substitute tool use for a real expert evaluation because tools will miss many usability and execution errors.



D:/Program Files/COAST/COAST WebMaster Pro/SiteRpts/summary.htm

Executive Summary

for http://www.democompany.com/home.cfm

Executive Summary Problem Reports Site Properties

This report highlights key problem areas on http://www.democompany.com/home.cfm as detected by COAST WebMaster Pro.

Scan Details	Analysis completed on Friday, May 24, 2002 4:24 PM using COAST WebMaster Pro.
Broken Links	9 (1.00%) of the 903 links on this site are broken.
Pages with Broken Links	10 (1.40%) of the 712 pages on this site have links that are broken.
Slow Pages	0 (0.00%) of the 712 pages on the site have download times exceeding 60s at 28.8 modem speeds.
Permanently Moved Pages	0 (0.00%) of the 903 links are permanently moved. Links to each permanently moved page should be updated to reflect its new location.
PageRule Failures	0 (0%) of the 0 PageRules on this site failed. 0 (0%) of the 0 global PageRules on this site failed. 0 pages on the site failed global PageRules

Created by COAST WebMaster Pro on Friday, May 24, 2002 4:24 PM
Visit COAST Software Inc. at http://www.coast.com/ for information on COAST WebMaster Pro.

Figure 5-4. *Site Quality Report from a maintenance tool*

User Testing

While the evaluation process just described is useful to uncover many types of site problems, it is important not to limit evaluations just to inspection. Developers may focus on certain things and completely miss problems commonly encountered by users. Further, this form of evaluation does not adequately reflect how users actually use a site.

Looking at log files can provide valuable insight into how a site is used. Log files will show who is looking at a site (by IP address or domain name, mostly), what pages users commonly look at, when they look at these pages, the paths users take through a site, the links followed to get to a site, and even what kind of browsers are being used. The log file really does show if content is popular and may provide a great deal of information related to site usability. For example, a tremendous number of users leaving the site from a certain page may indicate a problem. Log files can be used to verify assumptions or even show places to look for problems.

While log files provide a great deal of useful information, they really say very little about a user's feelings about a site. An invaluable way to evaluate a site is to watch how users actually use a site and try to solicit feedback from them. Conducting a user test can be difficult. Be careful to focus more on what users do and not on what the say. Users typically don't want to look stupid and will often indicate that they understand something when they don't.

Rule: Pay attention more to what users do than to what they say.

The best way to deal with this problem is not to let users know that they are taking a test; you might even try to casually watch them without their knowledge. If you ask users to take a usability test, you may find that they pay more attention or try harder to figure things out than they might usually. The assumption almost seems to be that test administrators will be pleased at how proficient they are. At the opposite end of spectrum, on occasion testers will purposefully look for errors. In either case, it should be evident that testing conditions may not always be the same as user conditions.

A very important aspect of testing is making sure not to get too involved. For example, if you ask users to evaluate a site, don't guide them through it. If you co-pilot the users' browsing sessions, they will uncover only what you want them to and maybe not use the site as they might normally. If you talk too much, showing off the features of the site, you may not give users a chance to say what they think. User testing can be very difficult for site designers who want to put their work in the best light possible, and they may be very unwilling to listen to user criticisms.

Suggestion: Consider having a person not involved in the site design process conduct a user test.

You can certainly be very scientific about user testing: using two-way mirrors, recording mouse travel and keystrokes, and even monitoring pauses or mistakes made by the user during a typical task. Some might go so far as to watch facial expressions or even monitor the blood pressure of the test subject. However, the end result is often really the most important aspect of the test. Remember that, in the final analysis, probably the only real important things to users are whether they were successful in their mission and enjoyed the visit. This does not mean that the study of usability lacks reasonably measurable characteristics; it just suggests that, as imperfect creatures, humans may not always act logically and may even quickly forget the difficulty of performing a task if there is a wonderful reward at the end. Readers interested in understanding more about user testing and usability, particularly the theory and practice of conducting usability tests, should visit http://www.useit.com and http://www.usableweb.com.

Summary

Site evaluations serve both to provide quality assurance and to increase the skills and knowledge of developers. This chapter provided an overview of designer-directed evaluation, while focusing on execution and usability. The tips provided here in conjunction with the detailed checklist presented in Appendix B should uncover many of the common problems in Web sites. However, users may uncover more, and user evaluations should always be performed if possible because, in the end, the acceptability of the site will be determined by the users. However, do not discount developer evaluation, since it makes no sense to have users evaluate a site that is obviously built incorrectly or that exhibits known usability problems.