Second Edition

/Theory/In/Practice

The Art of SEO

Mastering Search Engine Optimization

Eric Enge
Stephan Spencer
Jessie Stricchiola
Rand Fishkin
Foreword by John Battelle

O'REILLY®

The Art of SEO, Second Edition
by Eric Enge, Stephan Spencer, Jessie Stricchiola, and Rand Fishkin

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editor: Mary Treseler
Production Editor: Melanie Yarbrough
Copyeditor: Rachel Head
Proofreader: Kiel Van Horn
Indexer: Ellen Troutman Zaig
Cover Designer: Karen Montgomery
Interior Designer: David Futato
Illustrator: Robert Romano

March 2012: Second Edition.

Revision History for the Second Edition:
2012-03-02 First release
See *http://oreilly.com/catalog/errata.csp?isbn=9781449304218* for release details.

ISBN: 978-1-449-30421-8

[LSI]

1330962786

CONTENTS

CHAPTER SIX

Developing an SEO-Friendly Website

In this chapter, we will examine the major elements of how to assess the search engine friendliness of your site. Making your site content accessible to search engines is the first step toward creating visibility in search results. Once your website content is accessed by a search engine, it can then be considered for relevant positioning within the SERPs.

As we discussed in the introduction to Chapter 2, search engine crawlers are basically software programs. This gives them certain strengths and weaknesses. Publishers must adapt their websites to make the job of these software programs easier—in essence, leverage their strengths and make their weaknesses irrelevant. If you can do this, you will have taken a major step forward toward success with SEO.

Developing an SEO-friendly site architecture requires a significant amount of thought, planning, and communication, due to the large number of factors that influence the ways a search engine sees your site and the large number of ways in which a website can be put together. There are hundreds (if not thousands) of tools that web developers can use to build a website, many of which were not initially designed with SEO, or search engine crawlers, in mind.

Making Your Site Accessible to Search Engines

The first step in the SEO design process is to ensure that your site can be found and crawled by the search engines. This is not as simple as it sounds, as there are many popular web design and implementation constructs that the crawlers may not understand.

Indexable Content

To rank well in the search engines, your site's content—that is, the material available to visitors of your site—should be in HTML text form. For example, while the search engines do crawl images and Flash files, these are content types that are difficult for search engines to analyze, and therefore they do not help them determine the topical relevance of your pages. With Flash, for example, while specific *.swf* files (the most common file extension for Flash) can be crawled and indexed—and are often found when the user searches for specific words or phrases that appear in their filenames and indicates that he is searching only for *.swf* files—it is rare that a generic query returns a Flash file or a website generated entirely in Flash as a highly relevant result, due to the lack of "readable" content. This is not to say that websites developed using Flash are inherently irrelevant, or that it is impossible to successfully optimize a website that uses Flash for search; however, in our experience the preference is almost always given to HTML-based files.

The search engines also face challenges with "identifying" images from a relevance perspective, as there are minimal text-input fields for image files in GIF, JPEG, or PNG format (namely the filename, title, and `alt` attribute). While we do strongly recommend accurate labeling of images in these fields, images alone are usually not enough to earn a web page top rankings for relevant queries. However, the search engines are improving in this area, and we expect image identification technology to continue to advance.

In June 2011, Google announced improvements to its image search functionality, offering the ability for users to perform a search using an image as the search query as opposed to text (though users can input text to augment the query). By uploading an image, dragging and dropping an image from the desktop, entering an image URL, or right-clicking on an image within a browser (Firefox and Chrome with installed extensions), users can often find other locations of that image on the Web for reference and research, as well as images that "appear" similar in tone and composition. While this does not immediately change the landscape of SEO for images, it does give us an indication as to how Google is augmenting its current relevance indicators for image content.

Spiderable Link Structures

As we outlined in Chapter 2, search engines use links on web pages to help them discover other web pages and websites. For this reason, we strongly recommend taking the time to build an internal linking structure that spiders can crawl easily. Many sites make the critical mistake of hiding or obfuscating their navigation in ways that limit spider accessibility, thus impacting their ability to get pages listed in the search engines' indexes. Consider the illustration in Figure 6-1 that shows how this problem can occur.

In Figure 6-1, Google's spider has reached Page A and sees links to pages B and E. However, even though pages C and D might be important pages on the site, the spider has no way to reach them (or even to know they exist), because no direct, crawlable links point to those

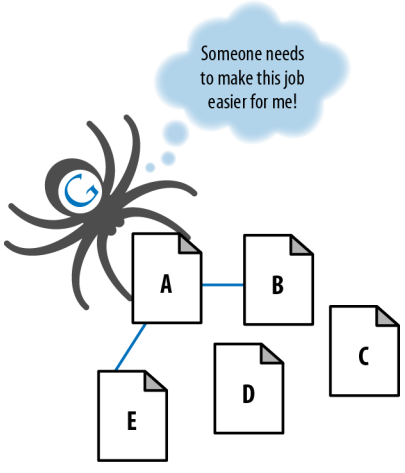

FIGURE 6-1. Providing search engines with crawlable link structures

pages. As far as Google is concerned, they might as well not exist—great content, good keyword targeting, and smart marketing won't make any difference at all if the spiders can't reach those pages in the first place.

To refresh your memory on the discussion in Chapter 2, here are some common reasons why pages may not be reachable:

Links in submission-required forms
: Search spiders will not attempt to "submit" forms, and thus any content or links that are accessible only via a form are invisible to the engines. This even applies to simple forms such as user logins, search boxes, or some types of pull-down lists.

Links in hard-to-parse JavaScript
: If you use JavaScript for links, you may find that search engines either do not crawl or give very little weight to the embedded links.

Links in Flash, Java, or other plug-ins
: Links embedded inside Java and plug-ins are invisible to the engines. In theory, the search engines are making progress in detecting links within Flash, but don't rely too heavily on this.

Links in PowerPoint and PDF files
: PowerPoint and PDF files are no different from Flash, Java, and plug-ins. Search engines sometimes report links seen in PowerPoint files or PDFs, but how much they count for is not easily known.

Links pointing to pages blocked by the meta `robots` *tag,* `rel="NoFollow"`*, or robots.txt*
: The *robots.txt* file provides a very simple means for preventing web spiders from crawling pages on your site. Use of the `NoFollow` attribute on a link, or placement of the meta

`robots` tag on the page containing the link, is an instruction to the search engine to not pass link juice via that link (a concept we will discuss further in "Content Delivery and Search Spider Control" on page 245). For more on this, please reference the following blog post from Google's Matt Cutts: *http://www.mattcutts.com/blog/pagerank-sculpting/*.

Links on pages with many hundreds or thousands of links
: Google has a suggested guideline of 100 links per page before it may stop spidering additional links from that page. This "limit" is somewhat flexible, and particularly important pages may have upward of 150 or even 200 links followed. In general, however, it is wise to limit the number of links on any given page to 100 or risk losing the ability to have additional pages crawled.

Links in frames or iframes
: Technically, links in both frames and iframes can be crawled, but both present structural issues for the engines in terms of organization and following. Unless you're an advanced user with a good technical understanding of how search engines index and follow links in frames, it is best to stay away from them as a place to offer links for crawling purposes. We will discuss frames and iframes in more detail in "Creating an Optimal Information Architecture (IA)" on page 188.

XML Sitemaps

Google, Yahoo!, and Bing (from Microsoft, formerly MSN Search, and then Live Search) all support a protocol known as XML Sitemaps. Google first announced it in 2005, and then Yahoo! and MSN Search agreed to support the protocol in 2006. Using the Sitemaps protocol you can supply the search engines with a list of all the pages you would like them to crawl and index.

Adding a URL to a Sitemap file does not guarantee that it will be crawled or indexed. However, it can result in pages that are not otherwise discovered or indexed by the search engines getting crawled and indexed.

This program is a complement to, not a replacement for, the search engines' normal, link-based crawl. The benefits of Sitemaps include the following:

- For the pages the search engines already know about through their regular spidering, they use the metadata you supply, such as the date when the content was last modified (*lastmod date*) and the frequency at which the page is changed (*changefreq*), to improve how they crawl your site.
- For the pages they don't know about, they use the additional URLs you supply to increase their crawl coverage.
- For URLs that may have duplicates, the engines can use the XML Sitemaps data to help choose a canonical version.
- Verification/registration of XML Sitemaps may indicate positive trust/authority signals.

- The crawling/inclusion benefits of Sitemaps may have second-order positive effects, such as improved rankings or greater internal link popularity.

Matt Cutts, the head of Google's webspam team, has explained Google Sitemaps in the following way:

> Imagine if you have pages A, B, and C on your site. We find pages A and B through our normal web crawl of your links. Then you build a Sitemap and list the pages B and C. Now there's a chance (but not a promise) that we'll crawl page C. We won't drop page A just because you didn't list it in your Sitemap. And just because you listed a page that we didn't know about doesn't guarantee that we'll crawl it. But if for some reason we didn't see any links to C, or maybe we knew about page C but the URL was rejected for having too many parameters or some other reason, now there's a chance that we'll crawl that page C.

Sitemaps use a simple XML format that you can learn about at *http://www.sitemaps.org*. XML Sitemaps are a useful and in some cases essential tool for your website. In particular, if you have reason to believe that the site is not fully indexed, an XML Sitemap can help you increase the number of indexed pages. As sites grow in size, the value of XML Sitemap files tends to increase dramatically, as additional traffic flows to the newly included URLs.

Layout of an XML Sitemap

The first step in the process of creating an XML Sitemap is to create an *.xml* Sitemap file in a suitable format. Since creating an XML Sitemap requires a certain level of technical know-how, it would be wise to involve your development team in the XML Sitemap generation process from the beginning. Figure 6-2 shows an example of some code from a Sitemap.

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
   <url>
      <loc>http://www.example.com/</loc>
      <lastmod>2005-01-01</lastmod>
      <changefreq>monthly</changefreq>
      <priority>0.8</priority>
   </url>
</urlset>
```

FIGURE 6-2. Sample XML Sitemap from Google.com

To create your XML Sitemap, you can use any of the following:

An XML Sitemap generator
: This is a simple script that you can configure to automatically create Sitemaps, and sometimes submit them as well. Sitemap generators can create these Sitemaps from a URL list, access logs, or a directory path hosting static files corresponding to URLs. Here are some examples of XML Sitemap generators:

  - SourceForge.net's google-sitemap_gen (*http://sourceforge.net/projects/goog-sitemapgen/files/sitemapgen/*)
  - XML-Sitemaps.com's Sitemap Generator (*http://www.xml-sitemaps.com*)
  - Sitemaps Pal
  - GSite Crawler

Simple text
: You can provide Google with a simple text file that contains one URL per line. However, Google recommends that once you have a text Sitemap file for your site, you use the Sitemap Generator to create a Sitemap from this text file using the Sitemaps protocol.

Syndication feed
: Google accepts Really Simple Syndication (RSS) 2.0 and Atom 1.0 feeds. Note that the feeds may provide information on recent URLs only.

What to include in a Sitemap file

When you create a Sitemap file, you need to take care in situations where your site has multiple URLs that refer to one piece of content. Include *only* the preferred (canonical) version of the URL, as the search engines may assume that the URL specified in a Sitemap file is the preferred form of the URL for the content. You can use the Sitemap file as one way to suggest to the search engines which URL points to the preferred version of a given page.

In addition, be careful about what not to include. For example, do not include multiple URLs that point to identical content, and leave out pages that are simply pagination pages or alternate sort orders for the same content, and/or any low-value pages on your site. Last but not least, make sure that none of the URLs listed in the Sitemap file include any tracking parameters.

Mobile Sitemaps

Mobile Sitemaps should be used for content targeted at mobile devices. Mobile information is kept in a separate Sitemap file, which should not contain any information on nonmobile URLs. Google supports nonmobile markup as well as XHTML Mobile Profile (XHTML MP), WML (WAP 1.2), and cHTML markup. Details on the mobile Sitemap format can be found here: *http://www.google.com/support/webmasters/bin/answer.py?answer=34648*.

Video Sitemaps

Including information on your videos in your Sitemap file will increase their chances of being discovered by the search engines. Google indicates that it supports the following video formats: *.mpg*, *.mpeg*, *.mp4*, *.m4v*, *.mov*, *.wmv*, *.asf*, *.avi*, *.ra*, *.ram*, *.rm*, *.flv*, and *.swf*. You can see the specification of how video Sitemap entries are to be implemented here: *http://www.google.com/support/webmasters/bin/answer.py?answer=80472*.

Image Sitemaps

You can also increase visibility for your images by listing them in your Sitemap file. For each URL you include in your Sitemap file, you can also list the images that appear on that page. You can list up to 1,000 images per page. Specialized image tags are associated with the URL. The details of the format of these tags can be seen at *http://www.google.com/support/webmasters/bin/answer.py?answer=178636*.

Listing images in the Sitemap does increase the chances of those images being indexed. If you list some images on a page and not others, this will be interpreted as a signal that the images not listed are less important.

Where to upload your Sitemap file

When your Sitemap file is complete, upload the file to your site in the highest-level directory you want search engines to crawl (generally, the root directory), such as *www.yoursite.com/sitemap.xml*. You can include more than one subdomain in your Sitemap, provided that you verify the Sitemap for each subdomain in Google Webmaster Tools.

Managing and updating XML Sitemaps

Once your XML Sitemap has been accepted and your site has been crawled, monitor the results and update your Sitemap if there are issues. With Google, you can return to your Google Webmaster Tools account to view the statistics and diagnostics related to your Google Sitemaps; just click the site you want to monitor. You'll also find some FAQs from Google on common issues such as slow crawling and low indexation.

Update your XML Sitemap with Google and Bing when you add URLs to your site. You'll also want to keep your Sitemap file up-to-date when you add a large volume of pages or a group of pages that are strategic.

There is no need to update the XML Sitemap when simply updating content on existing pages. Further, if development resources are not available to update your Sitemap, it is not strictly necessary to immediately update it when pages are deleted, as the search engines will simply not be able to crawl those URLs; however, don't let a significant number of deleted pages remain in your Sitemap for long. You should update your Sitemap file whenever you add any new content, and you can remove any deleted pages at that time.

Updating your Sitemap with Bing. Simply update the *.xml* file in the same location as before.

Updating your Google Sitemap. You can resubmit your Google Sitemap using your Google Sitemaps account, or you can resubmit it using an HTTP request:

From Google Sitemaps
: Sign in to Google Webmaster Tools with your Google account. From the Sitemaps page, select the checkbox beside your Sitemap filename and click the Resubmit Selected button. The submitted date will update to reflect this latest submission.

From an HTTP request
: If you do this, you don't need to use the Resubmit link in your Google Sitemaps account. The Submitted column will continue to show the last time you manually clicked the link, but the Last Downloaded column will be updated to show the last time Google fetched your Sitemap. For detailed instructions on how to resubmit your Google Sitemap using an HTTP request, see *http://www.google.com/support/webmasters/bin/answer.py?answer=183669*.

Google and the other major search engines discover and index websites by crawling links. Google XML Sitemaps are a way to feed to Google the URLs that you want crawled on your site. This enables more complete crawling and indexation, which results in improved long-tail searchability. By creating and updating this *.xml* file, you are helping to ensure that Google recognizes your entire site, and this recognition will help people find your site. It also helps the search engines understand which version of your URLs (if you have more than one URL pointing to the same content) is the canonical version.

Creating an Optimal Information Architecture (IA)

Making your site friendly to search engine crawlers also requires that you put some thought into your site information architecture. A well-designed architecture can bring many benefits for both users and search engines.

The Importance of a Logical, Category-Based Flow

The search engines face myriad technical challenges in understanding your site. Crawlers are not able to perceive web pages in the way that humans do, and thus significant limitations for both accessibility and indexing exist. A logical and properly constructed website architecture can help overcome these issues and bring great benefits in search traffic and usability.

At the core of website information architecture are two critical principles: *usability*, or making a site easy to use; and *information architecture*, or crafting a logical, hierarchical structure for content.

In his book *Information Architects* (Graphis Inc.), Richard Saul Wurman, one of the very early information architecture proponents, developed the following definition for the term:

information architect. 1) the individual who organizes the patterns inherent in data, making the complex clear. 2) a person who creates the structure or map of information that allows others to find their personal paths to knowledge. 3) the emerging 21st century professional occupation addressing the needs of the age focused upon clarity, human understanding, and the science of the organization of information.

Usability and search friendliness

Search engines are trying to reproduce the human process of sorting relevant web pages by quality. If a real human were to do this job, usability and the user experience would surely play a large role in determining the rankings. Given that search engines are machines and they don't have the ability to segregate by this metric quite so easily, they are forced to employ a variety of alternative, secondary metrics to assist in the process. The most well known and well publicized among these is link measurement (see Figure 6-3), and a well-organized site is more likely to receive links.

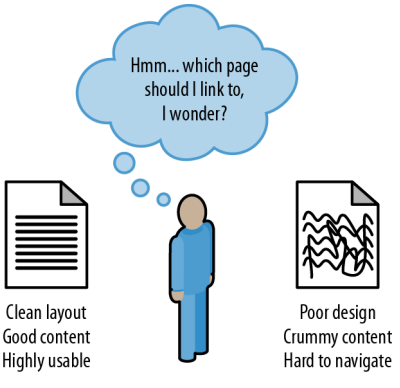

FIGURE 6-3. Making your site attractive to link to

Since Google launched in the late 1990s, search engines have strived to analyze every facet of the link structure on the Web, and they have extraordinary abilities to infer trust, quality, reliability, and authority via links. If you push back the curtain and examine why links between websites exist and why they were put in place, you can see that a human being (or several humans, if the organization suffers from bureaucracy) is almost always responsible for the creation of links.

The engines hypothesize that high-quality links will point to high-quality content, and that sites offering great content and positive user experiences will be rewarded with more links than those providing poor content and poor user experiences. In practice, the theory holds up well. Modern search engines have done a very good job of placing good-quality, usable sites in top positions for queries.

An analogy

Look at how a standard filing cabinet is organized. You have the individual cabinet, drawers in the cabinet, folders within the drawers, files within the folders, and documents within the files (see Figure 6-4).

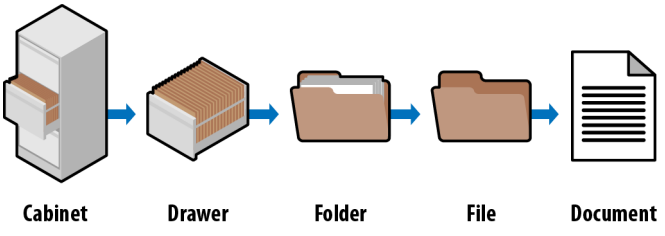

FIGURE 6-4. Similarities between filing cabinets and web pages

There is only one copy of any individual document, and it is located in a particular spot. There is a very clear navigation path to get to it.

If you wanted to find the January 2011 invoice for a client called Amalgamated Glove & Spat, you would go to the cabinet, open the drawer marked Client Accounts, find the Amalgamated Glove & Spat folder, look for the Invoices file, and then flip through the documents until you come to the January 2011 invoice (again, there is only one copy of this; you won't find it anywhere else).

Figure 6-5 shows what it looks like when you apply this logic to the popular website, Craigslist.org.

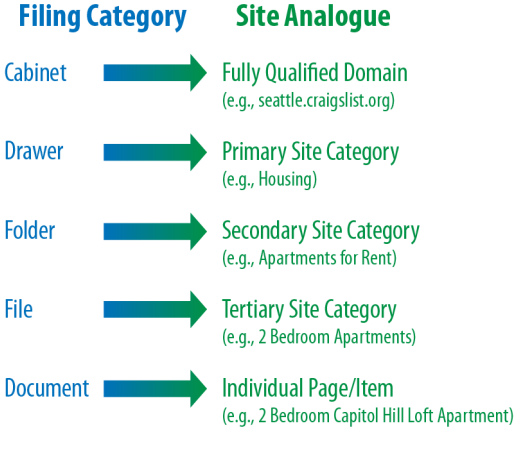

FIGURE 6-5. Filing cabinet analogy applied to Craigslist.org

To get to that final listing, you'd navigate to the Craigslist Seattle site, choose Apts/Housing, narrow your search down to two-bedroom apartments, and pick the two-bedroom loft from the list of available postings. Craigslist's simple, logical information architecture makes it easy to reach the desired post in four clicks, without having to think too hard at any step about where to go. This principle applies perfectly to the process of SEO, where good information architecture dictates:

- As few clicks as possible to get to any given page
- One hundred or fewer links per page (so as not to overwhelm either crawlers or visitors)
- A logical, semantic flow of links from home page to categories to detail pages

Here is a brief look at how this basic filing cabinet approach can work for some more complex information architecture issues.

Subdomains. You should think of subdomains as completely separate filing cabinets within one big room. They may share similar architecture, but they shouldn't share the same content; and, more importantly, if someone points you to one cabinet to find something, she is indicating that that cabinet is the authority, not the other cabinets in the room. Why is this important? It will help you remember that links (i.e., votes or references) to subdomains may not pass all, or any, of their authority to other subdomains within the room (e.g., **.craigslist.com*, wherein * is a variable subdomain name).

Those cabinets, their contents, and their authority are isolated from each other and may not be considered to be associated with one another. This is why, in most cases, it is best to have one large, well-organized filing cabinet instead of several different ones, as the latter arrangement may prevent users and bots from finding what they want.

Redirects. If you have an organized administrative assistant, he probably uses 301 redirects (discussed further in "Redirects" on page 262) inside his literal, metal filing cabinet. If he finds himself looking for something in the wrong place, he might place a sticky note there to remind himself of the correct location the next time he needs to look for that item. Anytime he looks for something in those cabinets, he will always be able to find it because if he navigates improperly, he will inevitably find a note pointing him in the right direction. One copy. One. Only. Ever.

Redirect irrelevant, outdated, or misplaced content to the proper spot in your filing cabinet and both your users and the engines will know what qualities and keywords you think it should be associated with.

URLs. It would be tremendously difficult to find something in a filing cabinet if every time you went to look for it, it had a different name, or if that name resembled "jklhj25br3g452ikbr52k"—a not-so-uncommon type of character string found in dynamic website URLs. Static, keyword-targeted URLs are much better for users and for bots. They can always be found in the same place, and they give semantic clues as to the nature of the content.

These specifics aside, thinking of your site information architecture in terms of a filing cabinet is a good way to make sense of best practices. It'll help keep you focused on a simple, easily navigated, easily crawled, well-organized structure. It is also a great way to explain an often complicated set of concepts to clients and coworkers.

Since search engines rely on links to crawl the Web and organize its content, the architecture of your site is critical to optimization. Many websites grow organically and, like poorly planned filing systems, become complex, illogical structures that force people (and spiders) looking for content to struggle to find what they want.

Site Architecture Design Principles

When planning your website, remember that nearly every user will initially be confused about where to go, what to do, and how to find what he wants. An architecture that recognizes this difficulty and leverages familiar standards of usability with an intuitive link structure will have the best chance of making a visit to the site a positive experience. A well-organized site architecture helps solve these problems and provides semantic and usability benefits to both users and search engines.

As Figure 6-6 demonstrates, a recipes website can use intelligent architecture to fulfill visitors' expectations about content and create a positive browsing experience. This structure not only helps humans navigate a site more easily, but also helps the search engines to see that your content fits into logical concept groups. You can use this approach to help you rank for applications of your product in addition to attributes of your product.

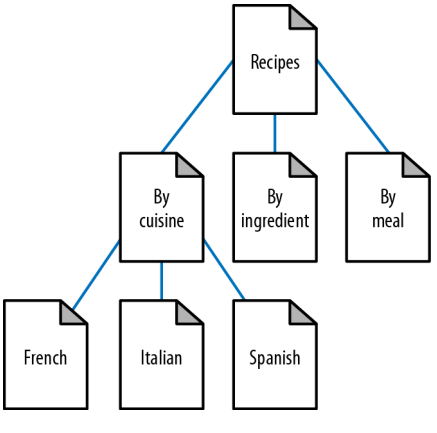

FIGURE 6-6. Structured site architecture

Although site architecture accounts for a small part of the algorithms, the engines do make use of relationships between subjects and give value to content that has been organized in a sensible fashion. For example, if in Figure 6-6 you were to randomly jumble the subpages into incorrect

categories, your rankings could suffer. Search engines, through their massive experience with crawling the Web, recognize patterns in subject architecture and reward sites that embrace an intuitive content flow.

Designing site architecture

Although site architecture—the creation of structure and flow in a website's topical hierarchy—is typically the territory of information architects and is created without assistance from a company's internal content team, its impact on search engine rankings, particularly in the long run, is substantial, thus making it wise to follow basic guidelines of search friendliness. The process itself should not be overly arduous, if you follow this simple protocol:

1. List all of the requisite content pages (blog posts, articles, product detail pages, etc.).
2. Create top-level navigation that can comfortably hold all of the unique types of detailed content on the site.
3. Reverse the traditional top-down process by starting with the detailed content and working your way up to an organizational structure capable of holding each page.
4. Once you understand the bottom, fill in the middle. Build out a structure for subnavigation to sensibly connect top-level pages with detailed content. In small sites, there may be no need for this level, whereas in larger sites, two or even three levels of subnavigation may be required.
5. Include secondary pages such as copyright, contact information, and other nonessentials.
6. Build a visual hierarchy that shows (to at least the last level of subnavigation) each page on the site.

Figure 6-7 shows an example of a structured site architecture.

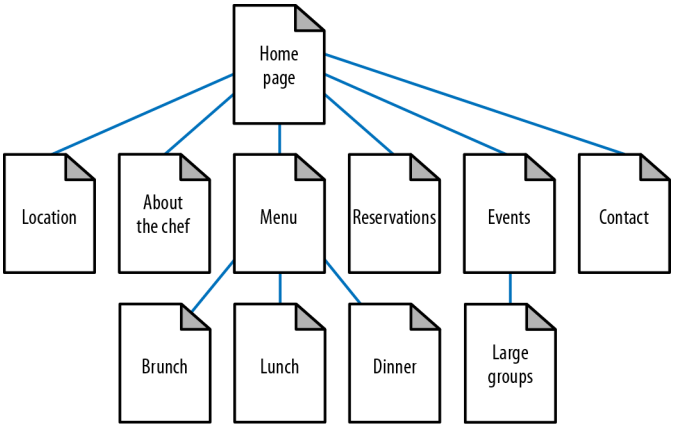

FIGURE 6-7. Second example of structured site architecture

Category structuring

As search engines crawl the Web, they collect an incredible amount of data (millions of gigabytes) on the structure of language, subject matter, and relationships between content. Though not technically an attempt at artificial intelligence, the engines have built a repository capable of making sophisticated determinations based on common patterns. As shown in Figure 6-8, search engine spiders can learn semantic relationships as they crawl thousands of pages that cover a related topic (in this case, dogs).

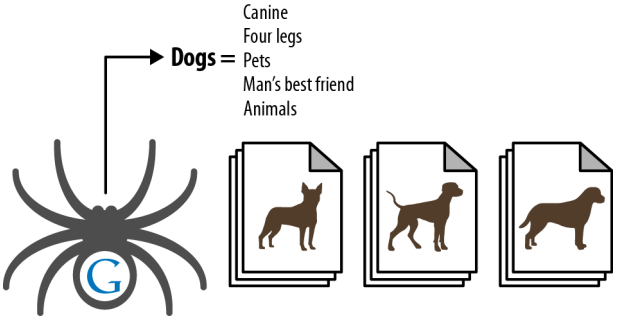

FIGURE 6-8. Spiders learning semantic relationships

Although content need not always be structured along the most predictable patterns, particularly when a different method of sorting can provide value or interest to a visitor, organizing subjects logically assists both humans (who will find your site easier to use) and engines (which will award you greater rankings based on increased subject relevance).

Topical relevance. Naturally, this pattern of relevance-based scoring extends from single relationships between documents to the entire category structure of a website. Site creators can best take advantage of this by building hierarchies that flow from broad, encompassing subject matter down to more detailed, specific content. Obviously, in any categorization system, there is a natural level of subjectivity. Don't get too hung up on perfecting what the engines want here—instead, think first of your visitors and use these guidelines to ensure that your creativity doesn't overwhelm the project.

Taxonomy and ontology

In designing a website, you should also consider its taxonomy and ontology. The taxonomy is essentially a two-dimensional hierarchical model of the architecture of the site. You can think of an ontology as mapping the way the human mind thinks about a topic area. It can be much more complex than a taxonomy, because a larger number of relationship types can be involved.

One effective technique for coming up with an ontology is called *card sorting*. This is a user-testing technique whereby users are asked to group related items together so that you can organize your site as intuitively as possible. Card sorting can help identify not only the

most logical paths through your site, but also ambiguous or cryptic terminology that should be reworded.

With card sorting, you write all the major concepts onto a set of cards that are large enough for participants to read, manipulate, and organize. Your test subjects place the cards in the order they believe provides the most logical flow, as well as into groups that seem to fit together.

By itself, building an ontology is not part of SEO, but when you do it properly it will impact your site architecture, and therefore it interacts with SEO. Coming up with the right site architecture should involve both disciplines.

Flat Versus Deep Architecture

One very strict rule for search friendliness is the creation of a flat site architecture. Flat sites require a minimal number of clicks to access any given page, whereas deep sites create long paths of links required to access detailed content. For nearly every site with fewer than 10,000 pages, all content should be accessible through a maximum of three clicks from the home page and/or sitemap page. At 100 links per page, even sites with millions of pages can have every page accessible in five to six clicks if proper link and navigation structures are employed. If a site is not built to be flat, it can take too many clicks to reach the desired content, as shown in Figure 6-9. In contrast, a flat site (see Figure 6-10) allows users and search engines to reach most content in just a few clicks.

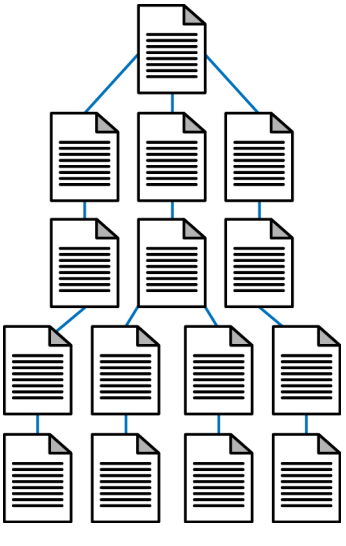

FIGURE 6-9. Deep site architecture

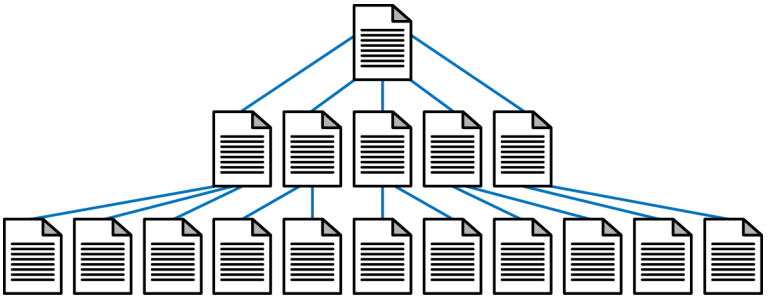

FIGURE 6-10. Flat site architecture

Flat sites aren't just easier for search engines to crawl; they are also simpler for users, as they limit the number of page visits the user requires to reach her destination. This reduces the abandonment rate and encourages repeat visits.

When creating flat sites, be aware that the engines are known to limit the number of links they crawl from a given page. As mentioned earlier, representatives from several of the major engines have said in the past that if a page contains more than 100 individual links, unless that page is of particular importance (i.e., many external sites link to it) it is likely that not all of those links will be followed.

This is not as big a problem today as it once was, as the search engines are able to handle bigger page sizes and larger numbers of links per page (*http://www.mattcutts.com/blog/how-many-links-per-page/*). However, there are still other reasons to avoid too many links per page, including potential usability issues.

The number of links per page issue relates directly to another rule for site architects: avoid excessive pagination wherever possible. *Pagination*, the practice of creating a sequence of pages to break up long lists of elements or long articles (e.g., some ecommerce sites use pagination for product catalogs that have more products than they wish to show on a single page), is problematic for many reasons.

First, pagination provides virtually no topical relevance. Second, pagination can potentially create duplicate content problems or be seen as indicative of poor-quality content. Last, pagination can create spider traps, and having hundreds or thousands of extraneous, low-quality pages can be detrimental to search visibility. There are ways to address the downsides of pagination, as we will discuss in a moment. Figure 6-11 shows an example of pagination.

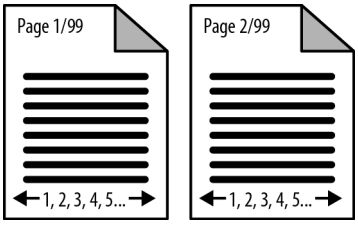

FIGURE 6-11. Pagination structures

So, make sure you implement flat structures and stay within sensible guidelines for the number of links per page, while retaining a contextually rich link structure. This is not always as easy as it sounds. On some sites, building a contextually rich structure may require quite a bit of thought and planning. Consider a site selling 10,000 different men's running shoes. Defining an optimal structure for that site could be a very large effort, but that effort will pay serious dividends in return.

Solutions to pagination problems vary based on the content of the website. Here are a few possibilities, with examples of when they may be useful:

1. Create a `View-All` page and use `rel="canonical"`. You may have lengthy articles that you choose to break into multiple pages. However, this results in links to the pages whose anchor text is something like "1", "2", and so forth. The titles of the various pages may not vary in any significant way, so they tend to compete with each other for search traffic. Finally, if someone links to the article but does not link to the first page, the link juice from that link will largely be wasted.

   One way to handle this problem is to retain the paginated version of the article, but also create a single-page version of the article. This is referred to as a `View-All` page. Then use the `canonical` tag (discussed in more detail later in this chapter, in "The canonical tag" on page 258) to point from the paginated pages to the `View-All` page. This will concentrate all of the link juice and search engine attention on one single page. You should also include a link to the `View-All` page from each of the individual paginated pages. However, if the `View-All` page is too slow in loading because of the page size this may not be the best option for you.
2. Use `rel="next"` and `rel="prev"`. At SMX East in September 2011, Googler Maile Ohye announced Google's support for new link elements called `rel="next"` and `rel="prev"`. The benefit of using these link elements is that it lets Google know when it has encountered a sequence of paginated pages. Once Google recognizes these tags, links to any of the pages will be treated as links to the series of pages as a whole. In addition, Google will show in the index the most relevant page in the series (most of the time this will be the first page, but not always).

While at the time of this writing Bing had not yet announced support for these tags, it is likely that it will do so in the near future. These tags can be used to inform Google about pagination structures, and they can be used whether or not you create a `View-All` page. The concept is simple. The following example outlines how to use the tags for content that is paginated into 12 pages:

a. In the `<head>` section of the first page of your paginated content, implement a `rel="next"` tag pointing to the second page of the content. The tag should look something like this:

```
<link rel="next" href="http://www.yoursite.com/products?prod=qwert&p=2" />
```

b. In the `<head>` section of the last page of your paginated content, implement a `rel="prev"` tag pointing to the second-to-last page of the content. The tag should look something like this:

```
<link rel="prev" href="http://www.yoursite.com/products?prod=qwert&p=11" />
```

c. In the `<head>` section of pages 2 through 11, implement `rel="next"` and `rel="prev"` tags pointing to the following and preceding pages, respectively. The following example shows what the tags should look like on page six of the content:

```
<link rel="prev" href="http://www.yoursite.com/products?prod=qwert&p=5" />
<link rel="next" href="http://www.yoursite.com/products?prod=qwert&p=7" />
```

It should also be noted that if you implement a `View-All` page and do not implement any of these tags, Google will attempt to discover that page and show it instead of the paginated versions in its search results. However, the authors recommend that you make use of one of the above solutions, as Google cannot guarantee that it will discover your `View-All` pages and it is best to provide it with as many clues as possible.

Search-Friendly Site Navigation

Website navigation is something that web designers have been putting considerable thought and effort into since websites came into existence. Even before search engines were significant, navigation played an important role in helping users find what they wanted. It plays an important role in helping search engines understand your site as well.

Basics of search engine friendliness

The search engine spiders need to be able to read and interpret your website's code to properly spider and index the content on your web pages. Do not confuse this with the rules of organizations such as the World Wide Web Consortium (W3C), which issues guidelines on HTML construction. Although following the W3C guidelines can be a good idea, the great majority of sites do not follow these guidelines, so search engines generally overlook violations of these rules as long as their spiders can parse the code.

Unfortunately, as we saw earlier in this chapter
(in "Spiderable Link Structures" on page 182), there are also a number of ways that navigation and content can be rendered on web pages that function for humans, but are invisible to or challenging for search engine spiders. Basic HTML text and HTML links such as those highlighted in Figure 6-12 work equally well for humans and search engine crawlers.

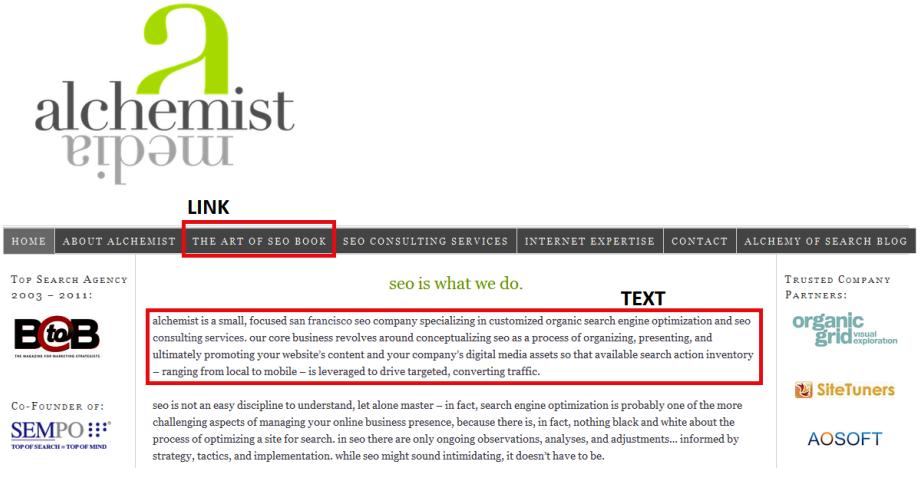

FIGURE 6-12. Example page with simple text and text link

Site elements that are problematic for spiders

While simple HTML is not a problem for the spiders, many other types of content may appear on a web page that work well for humans but not so well for search engines. Here are some of the most common problem areas.

Search and web forms. Many sites incorporate search functionality. These "site search" elements are specialized search engines that index and provide access to one site's content.

This is a popular method of helping users rapidly find their way around complex sites; for example, the Pew Internet website (*http://www.pewinternet.org*) provides a site search box in the top-right corner. This is a great tool for users, but search engines will be stymied by it. Search engines operate by crawling the Web's link structure—they don't submit forms or attempt random queries into search fields, and thus, any URLs or content solely accessible via a form will remain invisible to Google or Bing. In the case of site search tools, this is OK, as search engines do not want to index this type of content (they don't like to serve search results within their search results).

Forms are a popular way to provide interactivity, and one of the simplest applications is the "contact us" form many websites have.

Unfortunately, crawlers will not fill out or submit forms such as these; thus, any content restricted to those who employ them is inaccessible to the engines. In the case of a "contact us" form, this is likely to have little impact, but other types of forms can lead to bigger problems.

Websites that have content behind paywall and/or login barriers will either need to provide text links to the content behind the barrier (which defeats the purpose of the login) or implement First Click Free (discussed in "Content Delivery and Search Spider Control" on page 245).

Java, images, audio, and video. Adobe Shockwave files, Java embeds, audio, and video (in any format) present content that is largely uncrawlable by the major engines. With some notable exceptions that we will discuss later, search engines can read text only when it is presented in HTML format. Embedding important keywords or entire paragraphs in an image or a Java console renders them invisible to the spiders. Likewise, the search engines cannot easily understand words spoken in an audio file or video. However, Google has begun to leverage tools such as Google Voice Search in order to "crawl" audio content and extract meaning (this was first confirmed in the book *In the Plex* by Steven Levy, published by Simon & Schuster). Baidu already has an MP3 search function, and the Shazam and Jaikoz applications show the ability to identify song hashes today as well.

Using `alt` attributes, originally created as metadata for markup and an accessibility tag for vision-impaired users, is a good way to present at least some text content to the engines when displaying images or embedded, nontext content. Note that the `alt` attribute is not a strong signal, and using the `alt` attribute on an image link is no substitute for implementing a simple text link with targeted anchor text. A good alternative is to employ captions and text descriptions in the HTML content wherever possible.

In the past few years, a number of companies offering transcription services have cropped up, providing automated text creation for the words spoken in audio or video files. Providing these transcripts on rich media pages makes your content accessible to the search engines and findable by keyword-searching visitors. You can also use software such as Dragon Naturally Speaking and dictate your "transcript" to your computer.

AJAX and JavaScript. JavaScript enables many dynamic functions inside a website, most of which interfere very minimally with the operations of a search engine spider. The exception comes when a page must use a JavaScript call to reach another page, or to pull content that the spiders can't see in the HTML. In some instances this content is not visible to search engine spiders. However, Google has confirmed that it will attempt to execute JavaScript to access this type of content (*http://googlewebmastercentral.blogspot.com/2011/11/get-post-and-safely-surfacing-more-of.html*).

One example of this is Facebook Comments. Facebook Comments is a system offered by Facebook that allows publishers to collect comments from users on their site. Figure 6-13 shows an example of the Facebook Comments on a page on the TechCrunch website.

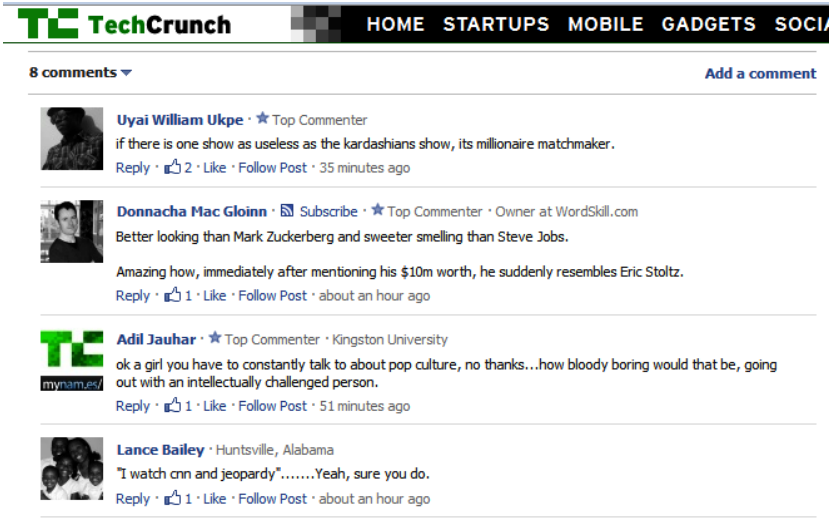

FIGURE 6-13. Facebook Comments on TechCrunch

If you examine the source code for this particular page you will not see any of the text strings for these comments in the HTML. This is because the comments are actually stored on Facebook and are dynamically retrieved by the web server when the page is rendered.

This is an example of the type of content that has not historically been indexed by the search engines, but Google started indexing these comments in October 2011. However, when you use a JavaScript implementation like this, it is not clear what Google or Bing will be able to do with it. Facebook Comments is a broadly used system, and it made sense for the search engines to learn how to read that content. Other uses of JavaScript may or may not be parsable. If your intent is to create content that you want the search engines to see, it is still safest to implement it in a form that is directly visible in the HTML of the web page.

Asynchronous JavaScript and XML (AJAX) presents similar problems, most notably in the delivery of content that search engines may not be able to spider. Since AJAX uses database calls to retrieve data without refreshing a page or changing URLs, the content contained behind these technologies may be completely hidden from the search engines (see Figure 6-14).

If a traditional AJAX implementation is used on your site, you may want to consider implementing an alternative spidering system for search engines to follow. AJAX applications are so user-friendly and appealing that for many publishers foregoing them is simply impractical. With these traditional implementations, building out a directory of links and pages that the engines can follow is a far better solution.

When you build these secondary structures of links and pages, make sure to provide users with access to them as well. Inside the AJAX application itself, give your visitors the option to "directly link to this page" and connect that URL with the URL you provide to search spiders

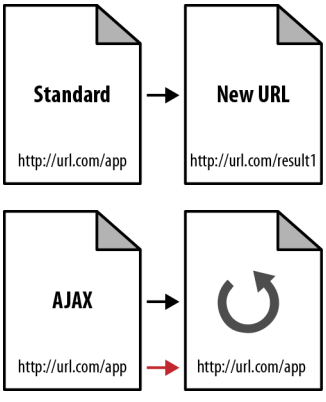

FIGURE 6-14. The problem with AJAX

through your link structures. AJAX apps not only suffer from content that can't be crawled, but often don't receive accurate links from users since the page URL doesn't change.

Newer versions of AJAX use a # delimiter, which acts as a query string into the AJAX application. This does allow you to link directly to different pages within the application. Search engines normally ignore the #, which is used for HTML bookmarking, and everything past it. This is largely because web browsers use what's after the # to jump to the anchor within the page, and that's done locally within the browser. In other words, the browser doesn't send the full URL, so the parameter information (i.e., any text after the #) is not passed back to the server.

Google outlined a method for making these AJAX pages visible to search engines back in 2009: *http://googlewebmastercentral.blogspot.com/2009/10/proposal-for-making-ajax-crawlable.html*. This was later followed up with recommendations made on the Google Code site: *http://code.google.com/web/ajaxcrawling/docs/getting-started.html*.

The solution proposed by Google involves making some slight modifications to the way your AJAX URLs are formatted so that its crawler can recognize when an AJAX URL can be treated like a static page (one that will always return the same content), in which case Googlebot will read the page and treat it like any other static page for indexing and ranking purposes.

Frames. Frames emerged in the mid-1990s as a popular way to make easy navigation systems. Unfortunately, both their usability (in 99% of cases) and their search friendliness (in 99.99% of cases) were exceptionally poor. Today, iframes and CSS can replace the need for frames, even when a site's demands call for similar functionality.

For search engines, the biggest problem with frames and iframes is that they often hold the content from two or more URLs on a single page. For users, the issue is that search engines, which direct searchers to only a single URL, may get confused by frames and direct visitors to single pages (orphan pages) inside a site intended to show multiple URLs at once.

Additionally, since search engines rely on links, and frame pages will often change content for users without changing the URL, external links often unintentionally point to the wrong URL. As a consequence, links to the page containing the frame or iframe may not actually point to the content the linker wanted to point to. Figure 6-15 shows an example page that illustrates how multiple pages are combined into a single URL with frames, which results in link distribution and spidering issues.

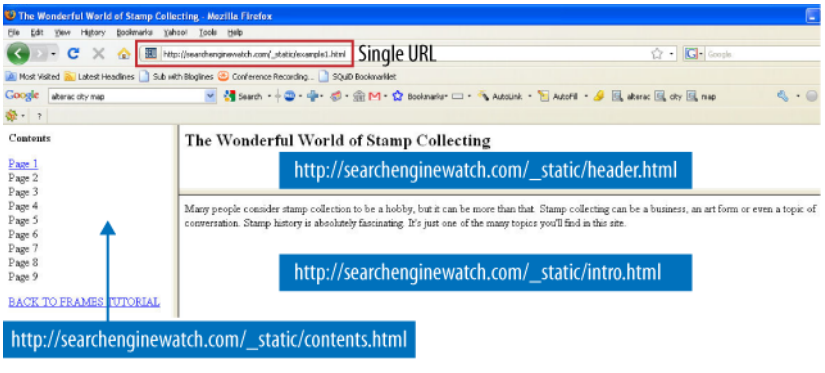

FIGURE 6-15. Sample page using frames

Search engine–friendly navigation guidelines

Although search engine spiders have become more advanced over the years, the basic premise and goals remain the same: spiders find web pages by following links and record the content of the pages they find in the search engine's index (a giant repository of data about websites and pages).

In addition to avoiding the techniques we just discussed, there are some additional guidelines for developing search engine–friendly navigation:

Implement a text link–based navigational structure

If you choose to create navigation in Flash, JavaScript, or some other technology the search engine may be unable to parse, make sure to offer alternative text links in HTML for spiders to ensure that automated robots (and visitors who may not have the required browser plug-ins) can reach your pages.

Beware of "spider traps"

Even intelligently coded search engine spiders can get lost in infinite loops of links that pass between pages on a site. Intelligent architecture that avoids looping 301 or 302 HTTP server codes (or other redirection protocols) should negate this issue, but sometimes having online calendar links, infinite pagination that loops, or massive numbers of ways in which content is accessible or sorted can result in tens of thousands of pages for search engine spiders to crawl when you intended to have only a few dozen true pages of content.

You can read more about Google's viewpoint on this at *http://googlewebmastercentral.blogspot.com/2008/08/to-infinity-and-beyond-no.html*.

Watch out for session IDs and cookies

As we just discussed, if you limit the ability of a user to view pages or redirect based on a cookie setting or session ID, search engines may be unable to crawl your content. The bots do not have cookies enabled, nor can they deal with session IDs properly (each visit by the crawler gets a URL with a different session ID, and the search engine sees these URLs with session IDs as different URLs). Although restricting form submissions is fine (as search spiders can't submit forms anyway), limiting content access via cookies and session IDs is a bad idea.

Be mindful of server, hosting, and IP issues

Server issues rarely cause search engine ranking problems—but when they do, disastrous consequences can follow. The engines are acutely aware of common server problems, such as downtime or overloading, and will give you the benefit of the doubt (though this will mean your content cannot be spidered during periods of server dysfunction). On the flip side, sites hosted on Content Delivery Networks (CDNs) may get crawled more heavily, and CDNs offer significant performance enhancements to a website.

The IP address of your host can be of concern in some instances. IP addresses once belonging to sites that have spammed the search engines may carry with them negative associations that can hinder spidering and ranking. While the engines aren't especially picky about shared hosting versus dedicated servers and dedicated IP addresses, or about server platforms, it should be noted that many hassles can be avoided by going these routes. At the very minimum, you should be cautious and find a host you trust, and inquire into the history and "cleanliness" of the IP address you may be assigned. The search engines keep track of domains, hosting services, IP addresses, and blocks of IP addresses that have a history of being used for spam sites. Their experience tells them that many of these have strong correlations with spam (and thus that removing them from the index can have great benefits for users). As a site owner *not* engaging in these practices, it pays to investigate your web host prior to getting into trouble.

NOTE

You can read more about server and hosting issues in "Identifying Current Server Statistics Software and Gaining Access" on page 118.

Root Domains, Subdomains, and Microsites

Among the common questions that arise when structuring a website (or restructuring one) are whether to host content on a new domain, when to use subfolders, and when to employ microsites.

As search engines scour the Web, they identify four kinds of web structures on which to place metrics:

Individual pages/URLs
: These are the most basic elements of the Web: filenames, much like those that have been found on computers for decades, which indicate unique documents. Search engines assign query-independent scores—most famously, Google's PageRank—to URLs and judge them in their ranking algorithms. A typical URL might look something like: *http://www.yourdomain.com/page.html*.

Subfolders
: The folder structures that websites use can also inherit or be assigned metrics by search engines (though there's very little information to suggest that they are used one way or another). Luckily, they are an easy structure to understand. In the URL *http://www.yourdomain.com/blog/post17.html*, */blog/* is the subfolder and *post17.html* is the name of the file in that subfolder. Engines may identify common features of documents in a given subfolder and assign metrics to these (such as how frequently the content changes, how important these documents are in general, or how unique the content is that exists in these subfolders).

Subdomains/fully qualified domains (FQDs)/third-level domains
: In the URL *http://blog.yourdomain.com/page.html*, three kinds of domain levels are present. The top-level domain (also called the *TLD* or *domain extension*) is *.com*, the second-level domain is *yourdomain*, and the third-level domain is *blog*. The third-level domain is sometimes referred to as a *subdomain*. Common web nomenclature does not typically apply the word *subdomain* when referring to *www*, although technically, this too is a subdomain. A fully qualified domain is the combination of the elements required to identify the location of the server where the content can be found (in this example, *http://blog.yourdomain.com/*).

  These structures can receive individual assignments of importance, trustworthiness, and value from the engines, independent of their second-level domains, particularly on hosted publishing platforms such as WordPress, Blogspot, Wetpaint, and so on.

Complete root domains/host domains/pay-level domains (PLDs)/second-level domains
: The domain name you need to register and pay for, and the one you point DNS settings toward, is the second-level domain (though it is commonly improperly called the "top-level" domain). In the URL *http://www.yourdomain.com/page.html*, *yourdomain.com* is the second-level domain. Other naming conventions may refer to this as the "root" or "pay-level" domain.

Figure 6-16 shows some examples.

FIGURE 6-16. Breaking down some example URLs

When to Use a Subfolder

If a subfolder will work, it is the best choice 99.9% of the time. Keeping content on a single root domain and single subdomain (e.g., *http://www.yourdomain.com*) gives the maximum SEO benefits, as the engines will maintain all of the positive metrics the site earns around links, authority, and trust and will apply these to every page on the site.

Subfolders have all the flexibility of subdomains (the content *can*, if necessary, be hosted on a unique server or a completely unique IP address, through post-firewall load balancing) and none of the drawbacks. Subfolder content will contribute directly to how search engines (and users, for that matter) view the domain as a whole. Subfolders can be registered with the major search engine tools and geotargeted individually to specific countries and languages as well.

Although subdomains are a popular choice for hosting content, they are generally not recommended if SEO is a primary concern. Subdomains *may* inherit the ranking benefits and positive metrics of the root domain they are hosted underneath, but they do not always do so (and in these scenarios, content can underperform).

When to Use a Subdomain

If your marketing team decides to promote a URL that is completely unique in content or purpose and would like to use a catchy subdomain to do it, using a subdomain can be practical. Maps.google.com is an example of where the marketing considerations make a subdomain an acceptable choice. One good reason to use a subdomain is in a situation where doing so can look more authoritative to users, as a result of creating separation from the main domain.

Be wary of press and media attention to the domains, as unsavvy users often don't understand the concept of subdomains or that domains can be on the "World Wide Web" without a "www." It is much less expensive to use a subfolder and have slightly less marketing panache than it is to educate through branding and advertising.

Subdomains may also be a reasonable choice if keyword usage in the domain name is of critical importance. It appears that search engines do weight keyword usage in the URL, and give slightly higher weight to exact keyword matches in the subdomain (or third-level domain name) than subfolders.

When to Use a Separate Root Domain

If you have a single, primary site that has earned links, built content, and attracted brand attention and awareness, it is very rarely advisable to place any new content on a completely separate domain. There are rare occasions when this can make sense, and we'll walk through these, as well as explaining how singular sites benefit from collecting all of their content in one root domain location.

Splitting similar or relevant content from your organization onto multiple domains can be likened to a store taking American Express Gold cards and rejecting American Express Corporate or American Express Blue—it is overly segmented and dangerous for the consumer mindset. If you can serve web content from a single domain, that domain will earn branding in the minds of your visitors and references from them, as well as links from other sites and bookmarks from your regular customers. Switching to a new domain forces you to rebrand and to earn all of these positive metrics all over again.

Microsites

There is a lot of debate about microsites, and although we generally recommend that you do not saddle yourself with the hassle of dealing with multiple sites and their SEO risks and disadvantages, it is important to understand the arguments, even if there are only a few, in favor of doing so.

Making the case for microsites

Optimized properly, a microsite may have dozens or even hundreds of pages. If your site is likely to gain more traction and interest with webmasters and bloggers by being at arm's length

from your main site, this approach may be worth considering—for example, if you have a very commercial main site and you want to create some great content that does not fit on that site, perhaps in the form of articles, podcasts, and RSS feeds.

When should you consider a microsite?

When you own a specific keyword search query domain
: For example, if you own usedtoyotatrucks.com, you might do very well to pull in search traffic for the specific term *used toyota trucks* with a microsite.

When you plan to sell the domains
: It is very hard to sell a folder or even a subdomain, so this strategy is understandable if you're planning to churn the domains in the second-hand market.

As discussed earlier, if you're a major brand building a "secret" or buzz-worthy microsite
: In this case, it can be useful to use a separate domain. However, you really should 301 the pages of that domain back to your main site after the campaign is over so that the link juice continues to provide long-term benefit—just as the mindshare and branding do in the offline world.

You should never implement a microsite that acts as a doorway page to your main site, or that has substantially the same content as you have published on your main site. Consider this only if you are willing to invest in putting rich original content on the site, and if you are willing to invest the time to promote the site as an independent site.

Such a site may gain more links by being separated from the main commercial site. A microsite may also have the added benefit of bypassing some of the legal and PR department hurdles and internal political battles. This could be a key consideration if you're at a monolithic or low risk-tolerance organization.

However, a microsite on a brand new domain may wallow in the Google sandbox for months (for more about the Google sandbox, see "Determining Searcher Intent and Delivering Relevant, Fresh Content" on page 46). So, what to do if you want to launch a microsite? Consider buying an aged, reputable "aftermarket" domain—one that has had a quality site on it for a while (parking pages don't count!)—and then change the domain registration information slowly so that the site's PageRank doesn't get reset to zero. Or start the clock running as soon as possible on your new domain by posting at least a few pages to the URL and then getting a few links to it, as far in advance of the official launch as possible.

Here are the reasons for not using a microsite:

Search algorithms favor large, authoritative domains
: Take a piece of great content about a topic and toss it onto a small, mom-and-pop website, point some external links to it, optimize the page and the site for the target terms, and get it indexed. Now, take that exact same content and place it on Wikipedia or CNN.com. You're virtually guaranteed that the content on the large, authoritative domain will

outrank the content on the small niche site. The engines' current algorithms favor sites that have built trust, authority, consistency, and history.

Multiple sites split the benefits of links

As suggested in Figure 6-17, a single good link pointing to a page on a domain positively influences the entire domain and every page on it. Because of this phenomenon, it is much more valuable to have any links you can possibly get pointing to the same domain to help boost the rank and value of the pages on it. Having content or keyword-targeted pages on other domains that don't benefit from the links you earn to your primary domain only creates more work.

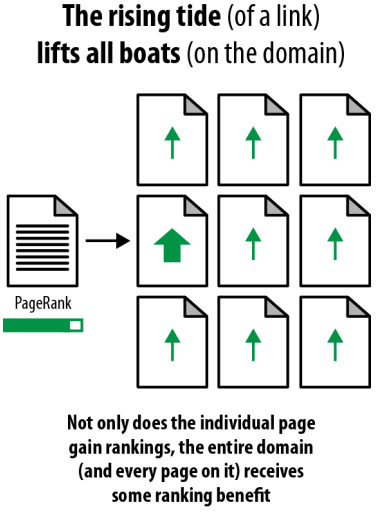

FIGURE 6-17. How links can benefit your whole site

100 links to Domain A ≠ 100 links to Domain B + 1 link to Domain A (from Domain B)

In Figure 6-18, you can see how earning lots of links to Page G on a separate domain is far less valuable than earning those same links to a page on the primary domain. Due to this phenomenon, even if you interlink all of the microsites or multiple domains that you build, the value you get still won't be close to the value you could get from those links if they were to point directly to the primary domain.

A large, authoritative domain can host a huge variety of content

Niche websites frequently limit the variety of their discourse and content matter, whereas broader sites can target a wider range of foci. This is valuable not just for targeting the long tail of search and increasing potential branding and reach, but also for viral content, where a broader focus is much less limiting than that of a niche focus.

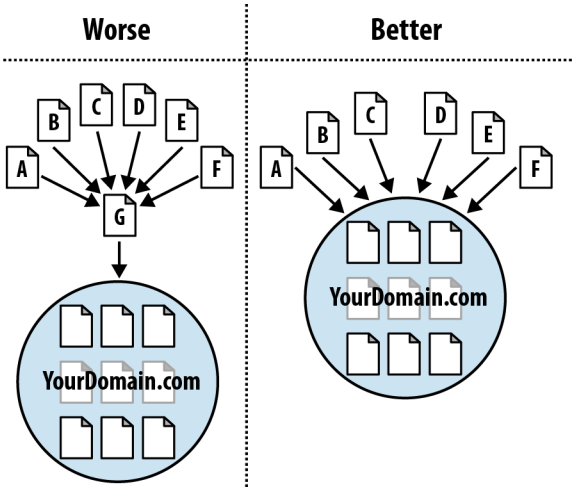

FIGURE 6-18. How direct links to your domain are better

Time and energy are better spent on a single property

If you're going to pour your heart and soul into web development, design, usability, user experience, site architecture, SEO, public relations, branding, and so on, you want the biggest bang for your buck. Splitting your attention, time, and resources amongst multiple domains dilutes that value and doesn't let the natural order of building on your past successes on a single domain assist with that process. As shown in Figure 6-18, every page on a site receives benefit from inbound links to a site. The page receiving the link gets the most benefit, but other pages also benefit.

When to Use a TLD Other than .com

There are only a few rare situations in which you should consider using a TLD other than *.com*:

- When you own the *.com* and want to redirect to a *.org*, *.tv*, *.biz*, etc., possibly for marketing/branding/geographic reasons. Do this only if you already own the *.com* and can redirect.
- When you can use a *.gov*, *.mil*, or *.edu* domain (*.jobs*, though technically restricted to HR and hiring organizations, is available to anyone who hires and doesn't have any special search benefits).
- When you are serving only a single geographic region and are willing to permanently forego growth outside that region (e.g., *.co.uk*, *.de*, *.it*, etc.).
- When you are a nonprofit and want to distance your organization from the commercial world. In this case, *.org* may be for you.

Optimization of Domain Names/URLs

Two of the most basic parts of any website are the domain name and the URLs for the pages of the website. This section will explore guidelines for optimizing these important elements.

Optimizing Domains

When a new site is being conceived or designed, one of the critical items to consider is the naming of the domain, whether it is for a new blog, a company launch, or even just a friend's website. Here are 12 tips that will be indispensable in helping you select a great domain name:

1. **Brainstorm five top keywords.** When you begin your domain name search, it helps to have five terms or phrases in mind that best describe the domain you're seeking. Once you have this list, you can start to pair them or add prefixes and suffixes to create good domain ideas. For example, if you're launching a mortgage-related domain, you might start with words such as *mortgage*, *finance*, *home equity*, *interest rate*, and *house payment*, and then play around until you can find a good match.
2. **Make the domain unique.** Having your website confused with a popular site that someone else already owns is a recipe for disaster. Thus, never choose a domain that is simply the plural, hyphenated, or misspelled version of an already established domain. For example, for years Flickr did not own Flicker.com, and probably lost traffic because of that. They recognized the problem and bought the domain, and as a result *http://flicker.com* now redirects to *http://flickr.com*.
3. **Choose only .com-available domains or the ccTLD for your country.** If you're not concerned with type-in traffic, branding, or name recognition, you don't need to worry about this one. However, if you're at all serious about building a successful website over the long term, you should be worried about all of these elements, and although directing traffic to a *.net* or *.org* is fine, owning and 301'ing the *.com*, or the ccTLD for the country your website serves (e.g., *.co.uk* for the United Kingdom), is critical. With the exception of the very tech-savvy, most people who use the Web still make the automatic assumption that *.com* or the ccTLD for their country is all that's out there, or that these domains are more trustworthy. Don't make the mistake of locking out or losing traffic from these folks.
4. **Make it easy to type.** If a domain name requires considerable attention to type correctly due to spelling, length, or the use of unmemorable words or sounds, you've lost a good portion of your branding and marketing value. Usability folks even tout the value of having the words include easy-to-type letters (which we interpret as avoiding *q*, *z*, *x*, *c*, and *p*).
5. **Make it easy to remember.** Remember that word-of-mouth marketing relies on the ease with which the domain can be called to mind. You don't want to be the company with the terrific website that no one can ever remember to tell their friends about because they can't remember the domain name.

6. **Keep the name as short as possible.** Short names are easy to type and easy to remember (see the previous two rules). Short names also allow more of the URL to display in the SERPs and are a better fit on business cards and other offline media.
7. **Create and fulfill expectations.** When someone hears about your domain name for the first time, he should be able to instantly and accurately guess what type of content he might find there. That's why we love domain names such as Hotmail.com, CareerBuilder.com, AutoTrader.com, and WebMD.com. Domains such as Monster.com, Amazon.com, and Zillow.com required far more branding because of their nonintuitive names.
8. **Avoid trademark infringement.** This is a mistake that isn't made too often, but it can kill a great domain and a great company when it is. To be sure you're not infringing on anyone's registered trademark with your site's name, visit the US Patent and Trademark office site (*http://www.uspto.gov/trademarks/index.jsp*) and search before you buy. Knowingly purchasing a domain that includes a trademarked term with bad-faith intent is a form of cybersquatting referred to as *domain squatting*.
9. **Set yourself apart with a brand.** Using a unique moniker is a great way to build additional value with your domain name. A "brand" is more than just a combination of words, which is why names such as Mortgageforyourhome.com and Shoesandboots.com aren't as compelling as branded names such as Yelp.com and Gilt.com.
10. **Reject hyphens and numbers.** Both hyphens and numbers make it hard to convey your domain name verbally and fall down on being easy to remember or type. Avoid Roman or spelled-out numerals in domains, as both can be confusing and mistaken for the other.
11. **Don't follow the latest trends.** Website names that rely on odd misspellings (as do many Web 2.0–style sites), multiple hyphens (such as the SEO-optimized domains of the early 2000s), or uninspiring short adjectives (such as "top...*x*," "best...*x*," "hot...*x*") aren't always the best choice. This isn't a hard and fast rule, but in the world of naming conventions in general, just because everyone else is doing it doesn't mean it is a surefire strategy. Just look at all the people who named their businesses "AAA...*x*" over the past 50 years to be first in the phone book; how many Fortune 1,000s are named "AAA company"?
12. **Use an AJAX domain selection tool**. Websites such as Nameboy and Domjax make it exceptionally easy to determine the availability of a domain name. Just remember that you don't have to buy through these services. You can find an available name that you like, and then go to your registrar of choice.

Picking the Right URLs

Search engines place some weight on keywords in your URLs. Be careful, however, as the search engines can interpret long URLs with numerous hyphens in them (e.g., *Buy-this-*

awesome-product-now.html) as a spam signal. What follows are some guidelines for selecting optimal URLs for the pages of your site(s).

Describe your content

An obvious URL is a great URL. If a user can look at the address bar (or a pasted link) and make an accurate guess about the content of the page before ever reaching it, you've done your job. These URLs get pasted, shared, emailed, written down, and yes, even recognized by the engines.

Keep it short

Brevity is a virtue. The shorter the URL, the easier it is to copy and paste, read over the phone, write on a business card, or use in a hundred other unorthodox fashions, all of which spell better usability and increased branding.

Static is the way

The search engines treat static URLs differently than dynamic ones. Users also are not fond of URLs in which the big players are *?*, *&*, and =. They are just harder to read and understand.

Descriptives are better than numbers

If you're thinking of using */114/cat223/* you should go with */brand/adidas/* instead. Even if the descriptive isn't a keyword or particularly informative to an uninitiated user, it is far better to use words when possible. If nothing else, your team members will thank you for making it that much easier to identify problems in development and testing.

Keywords never hurt

If you know you're going to be targeting a lot of competitive keyword phrases on your website for search traffic, you'll want every advantage you can get. Keywords are certainly one element of that strategy, so take the list from marketing, map it to the proper pages, and get to work. For pages created dynamically through a CMS, try to configure it so you include keywords in the URL.

Subdomains aren't always the answer

First off, never use multiple subdomains (e.g., *product.brand.site.com*); they are unnecessarily complex and lengthy. Second, consider that subdomains have the potential to be treated separately from the primary domain when it comes to passing link and trust value. In most cases where just a few subdomains are used and there's good interlinking, it won't hurt, but be aware of the downsides. For more on this, and for a discussion of when to use subdomains, see "Root Domains, Subdomains, and Microsites" on page 204 earlier in this chapter.

Use fewer folders

A URL should contain no unnecessary folders (or words or characters, for that matter). They do not add to the user experience of the site and can in fact confuse users.

Hyphens separate best

When creating URLs with multiple words in the format of a phrase, hyphens are best to separate the terms (e.g., */brands/dolce-and-gabbana/*), but you can also use plus signs (+).

Stick with conventions

If your site uses a single format throughout, don't consider making one section unique. Stick to your URL guidelines once they are established so that your users (and future site developers) will have a clear idea of how content is organized into folders and pages. This can apply globally as well for sites that share platforms, brands, and so on.

Don't be case-sensitive

Since URLs can accept both uppercase and lowercase characters, don't ever, ever allow any uppercase letters in your structure. Unix/Linux-based web servers are case-sensitive, so *http://www.domain.com/Products/widgets/* is technically a different URL from *http://www.domain.com/products/widgets/*. Note that this is not true in Microsoft IIS servers, but there are a lot of Apache web servers out there. In addition, this is confusing to users, and potentially to search engine spiders as well. If you have them now, 301-redirect them to all-lowercase versions to help avoid confusion. If you have a lot of type-in traffic, you might even consider a 301 rule that sends any incorrect capitalization permutation to its rightful home.

Don't append extraneous data

There is no point in having a URL exist in which removing characters generates the same content. You can be virtually assured that people on the Web will figure it out, link to you in different fashions, confuse themselves, their readers, and the search engines (with duplicate content issues), and then complain about it.

Keyword Targeting

The search engines face a tough task: based on a few words in a query (or sometimes only one), they must return a list of relevant results, order them by measures of importance, and hope that the searcher finds what she is seeking. As website creators and web content publishers, you can make this process massively simpler for the search engines and, in turn, benefit from the enormous traffic they send by employing the same terms users search for in prominent positions on your pages.

This practice has long been a critical part of search engine optimization, and although other metrics (such as links) have a great deal of value in the search rankings, keyword usage is still at the core of targeting search traffic.

The first step in the keyword targeting process is uncovering popular terms and phrases that searchers regularly use to find the content, products, or services your site offers. There's an art and science to this process, but it consistently begins with a list of keywords to target (see Chapter 5 for more on this topic).

Once you have that list, you'll need to include these keywords in your pages. In the early days of SEO, the process involved stuffing keywords repetitively into every HTML tag possible. Now, keyword relevance is much more aligned with the usability of a page from a human perspective.

Since links and other factors make up a significant portion of the search engines' algorithms, they no longer rank pages with 61 instances of "free credit report" above pages that contain only 60. In fact, *keyword stuffing*, as it is known in the SEO world, can actually get your pages devalued via search engine penalties. The engines don't like to be manipulated, and they recognize keyword stuffing as a disingenuous tactic. Figure 6-19 shows an example of a page utilizing accurate keyword targeting.

Appropriate keyword usage includes creating titles, headlines, and content designed to appeal to searchers in the results (and entice clicks), as well as building relevance for search engines to improve your rankings.

Building a search-friendly site requires prominently employing the keywords searchers use to find content. This section explores some of the more prominent places where a publisher can place those keywords.

Title Tags

For keyword placement, title tags are the most critical element for search engine relevance. The title tag is in the `<head>` section of an HTML document, and it is the only piece of "meta" information about a page that influences relevancy and ranking.

The following eight rules represent best practices for title tag construction. Do keep in mind, however, that a title tag for any given page must directly correspond to that page's content. You may have five different keyword categories and a unique site page (or section) dedicated to each, so be sure to align a page's title tag content with its actual visible content as well.

1. **Incorporate keyword phrases.** This one may seem obvious, but it is critical to prominently include in your title tag whatever your keyword research shows as being the most valuable keywords for capturing searches.
2. **Place your keywords at the beginning of the title tag.** This provides the most search engine benefit. If you're doing this and you also want to employ your brand name in the title tag, you should place that at the end. There is a tradeoff here between SEO benefit and branding benefit that you should think about and make an explicit decision on. Major brands may want to place their brand at the start of the title tag as it may increase click-through rates. To decide which way to go you need to consider which need is greater for your business.
3. **Limit length to 65 characters (including spaces).** Content in title tags after 65 characters is probably given less weight by the search engines. At a minimum, the title tag

FIGURE 6-19. Title and heading tags—powerful for SEO

shown in the SERPs gets cut off at 65 characters. Watch this number carefully, though, as Google in particular is now supporting up to 70 characters in some cases.

4. **Target longer phrases if they are relevant.** When choosing what keywords to include in a title tag, use as many as are completely relevant to the page at hand while remaining accurate and descriptive. It can be much more valuable to have a title tag such as "SkiDudes | Downhill Skiing Equipment & Accessories" rather than simply "SkiDudes | Skiing Equipment"—including those additional terms that are both relevant to the page and receive significant search traffic can bolster your page's value.

   However, if you have separate landing pages for "skiing accessories" versus "skiing equipment," don't include one term in the other's title. You'll be cannibalizing your rankings by forcing the engines to choose which page on your site is more relevant for each phrase, and they might get it wrong. We will discuss the cannibalization issue in more detail shortly.

5. **Use a divider.** When splitting up the brand from the descriptive, options include | (a.k.a. the pipe), >, -, and :, all of which work well. You can also combine these where appropriate—for example, "Major Brand Name: Product Category - Product." These characters do not bring an SEO benefit, but they can enhance the readability of your title.

6. **Focus on click-through and conversion rates.** The title tag is exceptionally similar to the title you might write for paid search ads, only it is harder to measure and improve because the stats aren't provided for you as easily. However, if you target a market that is relatively stable in search volume week to week, you can do some testing with your title tags and improve the click-through rate.

   Watch your analytics and, if it makes sense, buy search ads on the page to test click-through and conversion rates of different ad text as well, even if it is for just a week or two. You can then look at those results and incorporate them into your titles, which can make a huge difference in the long run. A word of warning, though: don't focus entirely on click-through rates. Remember to continue measuring conversion rates.

7. **Target searcher intent.** When writing titles for web pages, keep in mind the search terms your audience employed to reach your site. If the intent is browsing or research-based, a more descriptive title tag is appropriate. If you're reasonably sure the intent is a purchase, download, or other action, make it clear in your title that this function can be performed at your site. Here is an example from *http://www.bestbuy.com/site/Cameras-Camcorders/Digital-Cameras/abcat0401000.c?id=abcat0401000*: "Digital Cameras: Buy Digital Cameras & Accessories - Best Buy."

8. **Be consistent.** Once you've determined a good formula for your pages in a given section or area of your site, stick to that regimen. You'll find that as you become a trusted and successful "brand" in the SERPs, users will seek out your pages on a subject area and will have expectations that you'll want to fulfill.

Meta Description Tags

Meta descriptions have three primary uses:

- To describe the content of the page accurately and succinctly
- To serve as a short text "advertisement" to click on your pages in the search results
- To display targeted keywords, not for ranking purposes, but to indicate the content to searchers

Great meta descriptions, just like great ads, can be tough to write, but for keyword-targeted pages, particularly in competitive search results, they are a critical part of driving traffic from the engines through to your pages. Their importance is much greater for search terms where the intent of the searcher is unclear or where different searchers might have different motivations.

Here are seven good rules for meta descriptions:

1. **Tell the truth**. Always describe your content honestly. If it is not as "sexy" as you'd like, spice up your content; don't bait and switch on searchers, or they'll have a poor brand association.
2. **Keep it succinct**. Be wary of character limits—currently Google displays up to 160 characters, Yahoo! up to 165, and Bing up to 200+ (they'll go to three vertical lines in some cases). Stick with the smallest—Google—and keep those descriptions at 160 characters (including spaces) or less.
3. **Author ad-worthy copy**. Write with as much sizzle as you can while staying descriptive, as the perfect meta description is like the perfect ad: compelling and informative.
4. **Test, refine, rinse, and repeat**. Just like an ad, you can test meta description performance in the search results, but it takes careful attention. You'll need to buy the keyword through paid results (PPC ads) so that you know how many impressions critical keywords received over a given time frame. Then you can use analytics to see how many clicks you got on those keywords and calculate your click-through rate.
5. **Analyze psychology**. The motivation for a natural-search click is frequently very different from that of users clicking on paid results. Users clicking on PPC ads may be very directly focused on making a purchase, whereas people who click on a natural result may be more interested in research or learning about the company or its products. Don't assume that successful PPC ad text will make for a good meta description (or the reverse).
6. **Include relevant keywords**. It is extremely important to have your keywords in the meta description tag—the boldface that the engines apply can make a big difference in visibility and click-through rate. In addition, if the user's search term is not in the meta description, chances are reduced that the meta description will be used as the description in the SERPs.

7. **Don't employ descriptions universally**. You shouldn't always write a meta description. Conventional logic may hold that it is usually wiser to write a good meta description yourself to maximize your chances of it being used in the SERPs, rather than letting the engines build one out of your page content; however, this isn't always the case.

   If the page is targeting one to three heavily searched terms/phrases, go with a meta description that hits users performing those searches. However, if you're targeting longer-tail traffic with hundreds of articles or blog entries or even a huge product catalog, it can sometimes be wiser to let the engines themselves extract the relevant text. The reason is simple: when engines show a page in the SERPs, they always display the keywords (and surrounding phrases) that the user searched for. If you try to force a meta description, you can end up creating one that is not appropriate for the search phrase your page gets matched to, which is not uncommon in a large, complex site. In some cases, the search engines will overrule your meta description anyway and create their own, but since you can't consistently rely on this behavior, opting out of meta descriptions is OK (and for massive sites, it can save hundreds or thousands of man-hours).

Heading (H1, H2, H3) Tags

The H*x* tags in HTML (`<h1>`, `<h2>`, `<h3>`, etc.) are designed to indicate a headline hierarchy in a document. Thus, an `<h1>` tag might be considered the headline of the page as a whole, whereas `<h2>` tags would serve as subheadings, `<h3>`s as tertiary-level headlines, and so forth. The search engines have shown a slight preference for keywords appearing in heading tags, notably the `<h1>` tag (which is the most important of these to employ).

In some cases, you can use the title tag of a page, containing the important keywords, as the `<h1>` tag. However, if you have a longer title tag, you may want to use a more focused, shorter heading tag incorporating the most important keywords from the title tag. When a searcher clicks a result in the SERPs, reinforcing the search term he just typed in with the prominent headline helps to indicate that he has arrived on the right page with the same content he sought.

Many publishers assume that what makes the `<h1>` a stronger signal is the size at which it is displayed. For the most part, the styling of your heading tags is not a factor in the SEO weight of the heading tag. You can style the tag however you want, as shown in Figure 6-20, provided that you don't go to extremes (e.g., making it too small to read).

FIGURE 6-20. Headings styled to match the site

Document Text

The HTML text on a page was once the center of keyword optimization activities. Metrics such as keyword density and keyword saturation were used to measure the perfect level of keyword usage on a page. As far as the search engines are concerned, however, the text in a document—and particularly the frequency with which a particular term or phrase is used—has very little impact on how happy a searcher will be with that page.

In fact, quite often a page laden with repetitive keywords in an attempt to please the engines will provide a very poor user experience; thus, although some SEO professionals today do claim to use *term weight* (a mathematical equation grounded in the real science of information retrieval) or other, more "modern" keyword text usage methods, nearly all optimization can be done very simply.

The best way to ensure that you've achieved the greatest level of targeting in your text for a particular term or phrase is to use it in the title tag, in one or more of the section headings (within reason), and in the copy on the web page. Equally important is to use other related phrases within the body copy to reinforce the context and the relevance of your main phrase to the page.

Although it is possible that implementing more instances of the key phrase on the page may result in some increase in ranking, this is increasingly unlikely to happen as you add more instances of the phrase. In addition, it can ruin the readability of some documents, which could hurt your ability to garner links to your site. Furthermore, testing has shown that document text keyword usage is such a small factor with the major engines that even one link of very low quality is enough to outweigh a page with perfect keyword optimization versus one that simply includes the targeted phrase naturally on the page (2 to 10 times, depending on the page length).

This doesn't mean keyword placement on pages is useless—you should always strive to include the keyword you're targeting at least a few times, and perhaps more, depending on the document length—but it does mean that aiming for "perfect" optimization on every page for every term is not generally the best use of your SEO time.

Image Filenames and alt Attributes

Incorporation of images on web pages can substantively enrich the user experience. However, the search engines cannot read the images directly. There are two elements that you can control to give the engines context for images:

The filename
: Search engines look at the image filename to see whether it provides any clues to the content of the image. Don't name your image *example.com/img4137a-b12.jpg*, as this name tells the search engine nothing at all about the image, and you are passing up the opportunity to include keyword-rich text.

  If the image is a picture of Abe Lincoln, name the file *abe-lincoln.jpg* and/or have the `src` URL string contain it, as in *example.com/abe-lincoln/portrait.jpg*.

Image `alt` *text*
: Image tags in HTML permit you to specify an attribute known as `alt`. This is a place where you can provide more information about what is in the image, and again where you can use your targeted keywords. Here is an example for the picture of Abe Lincoln:

  ```
  <img alt="Abe Lincoln photo" src="http://example.com/abe-lincoln.jpg" >
  ```

  Use the quotes if you have spaces in the text string of the `alt` content! Sites that have invalid `img` tags frequently lump a few words without quotes into the `img` tag, intended for the `alt` content—but with no quotes, all terms after the first word will be lost.

This usage of the image filename and of the `alt` attribute permits you to reinforce the major keyword themes of the page. This is particularly useful if you want to rank in image search. Make sure the filename and the `alt` text reflect the content of the picture, though, and do not artificially emphasize keywords unrelated to the image (even if they are related to the page). Although the `alt` attribute and the image filename are helpful, you should not use image links

as a substitute for text links with rich anchor text, as these carry much more weight from an SEO perspective.

Presumably, your picture will relate very closely to the content of the page, and using the image filename and the `alt` text will help reinforce the page's overall theme.

While not essential, it is worth mentioning that while Google has stated it places more emphasis on the `alt` attribute (*http://googlewebmastercentral.blogspot.com/2007/12/using-alt-attributes-smartly.html*), the `title` attribute is another area that can be used to describe an image's content. We recommend judicious use of the `title` attribute—specifically, using it only if it adds more guidance to users as opposed to simply repeating the content found in the `alt` attribute.

Boldface Text

Some SEO professionals who engage in considerable on-page optimization testing have noticed that, all else being equal, a page that includes the targeted keyword(s) in `<b>` or `<strong>` tags (HTML elements that boldface text visually) outrank their counterparts that do not employ boldface. Thus, although this is undoubtedly a very small factor in modern SEO, it may be worth leveraging, particularly for those looking to eke every last bit of optimization out of keyword usage.

Avoiding Keyword Cannibalization

As we discussed earlier, you should not use common keywords across multiple page titles. This advice applies to more than just the title tags.

One of the nastier problems that often crops up during the course of a website's information architecture, *keyword cannibalization* refers to a site's targeting of popular keyword search phrases on multiple pages, forcing the engines to pick which one is most relevant. In essence, a site employing cannibalization competes with itself for rankings and dilutes the ranking power of internal anchor text, external links, and keyword relevancy.

Avoiding cannibalization requires strict site architecture with attention to detail. Plot out your most important terms on a visual flowchart (or in a spreadsheet file, if you prefer), and pay careful attention to what search terms each page is targeting. Note that when pages feature two-, three-, or four-word phrases that contain the target search phrase of another page, linking back to that page within the content with the appropriate anchor text will avoid the cannibalization issue.

For example, if you had a page targeting "mortgages" and another page targeting "low-interest mortgages," you would link back to the "mortgages" page from the "low-interest mortgages" page using the anchor text "mortgages" (see Figure 6-21). You can do this in the breadcrumb or in the body copy. The *New York Times* (*http://www.nytimes.com*) does the latter, where keywords in the body copy link to the related resource page on the site.

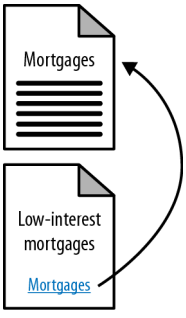

FIGURE 6-21. Adding lots of value with relevant cross-links

Keyword Targeting in CMSs and Automatically Generated Content

Large-scale publishing systems, or those that produce automatically generated content, present some unique challenges. If hundreds of pages are being created every day, it is not feasible to do independent keyword research on each and every page, making page optimization an interesting challenge.

In these scenarios, the focus turns to methods/recipes for generating unique titles, `<h1>` headings, and page content for each page. It is critical to educate the writers on ways to implement titles and headings that capture unique, key aspects of the articles' content. More advanced teams can go further with this and train their writing staff on the use of keyword research tools to further optimize this process.

In the case of automatically generated material (such as that produced from algorithms that mine data from larger textual bodies), the key is to automate a means for extracting a short (fewer than 70 characters) description of the article and making it unique from other titles generated elsewhere on the site and on the Web at large.

SEO Copywriting: Encouraging Effective Keyword Targeting by Content Creators

Very frequently, someone other than an SEO professional is responsible for content creation. Content creators often do not have an innate knowledge as to why keyword targeting is important—and therefore, training for effective keyword targeting is a critical activity. This is particularly important when dealing with large websites and large teams of writers.

Here are the main components of SEO copywriting that your writers must understand:

- Search engines look to match up a user's search queries with the keyword phrases on your web pages. If a search phrase does not appear on your page, chances are good that the page will never achieve significant ranking for that search phrase.

- The search phrases users may choose to use when looking for something are infinite in variety, but certain phrases will be used much more frequently than others.
- Using the more popular phrases you wish to target on a web page in the content for that page is essential to SEO success for that page.
- The title tag is the most important element on the page. Next up is the first header (`<h1>`), and then the main body of the content.
- Tools exist (as outlined in Chapter 5) that allow you to research and determine what the most interesting phrases are.

If you can get these five points across, you are well on your way to empowering your content creators to perform solid SEO. The next key element is training them on how to pick the right keywords to use.

This can involve teaching them how to use keyword research tools similar to the ones we discussed in Chapter 5, or having the website's SEO person do the research and provide the terms to the writers.

The most important factor to reiterate to the content creator is that content quality and the user experience still come first. Then, by intelligently making sure the right keyphrases are properly used throughout the content, they can help bring search engine traffic to your site. Reverse these priorities, and you can end up with keyword stuffing or other spam issues.

Long-Tail Keyword Targeting

As we outlined in Chapter 5, the small-volume search terms, when tallied up, represent 70% of all search traffic, while the more obvious, high-volume terms represent only 30% of the overall search traffic.

For example, if you run a site targeting searches for *new york pizza* and *new york pizza delivery*, you might be surprised to find that the hundreds of single searches each day for terms such as *pizza delivery on the corner of 57th & 7th*, or *Manhattan's tastiest Italian-style sausage pizza*, when taken together, will actually provide considerably more traffic than the popular phrases you've researched. This concept is called the *long tail of search*.

Targeting the long tail is another aspect of SEO that combines art and science. In Figure 6-22, you may not want to implement entire web pages for a history of pizza dough, pizza with white anchovies, or Croatian pizza. You may get traffic on these terms, but they are not likely to convert into orders for pizza.

Finding scalable ways to chase long-tail keywords is a complex topic. Perhaps you have a page for ordering pizza in New York City, and you have a good title and `<h1>` header on the page (e.g., "New York City Pizza: Order Here"), as well as a phone number and a form for ordering the pizza, and no other content. If that is all you have, that page is not competing effectively for rankings on long-tail search terms. To fix this, you need to write additional content for the

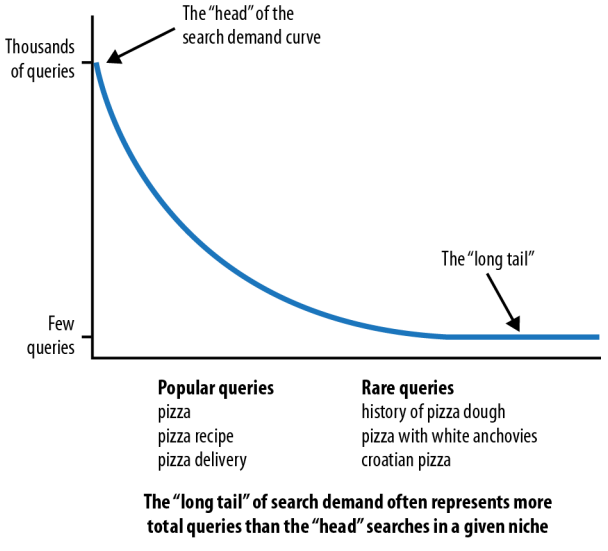

FIGURE 6-22. Example of the long-tail search curve

page. Ideally, this would be content that talks about the types of pizza that are popular in New York City, the ingredients used, and other things that might draw in long-tail search traffic.

If you also have a page for ordering pizza in San Jose, the picture gets even more complicated. You don't really want your content on the San Jose page to be the same as it is on the New York City page. You run the risk of running into potential duplicate content problems, as we will outline in "Duplicate Content Issues" on page 234, or the keyword cannibalization issues we discussed earlier in this chapter.

To maximize your success, find a way to generate different content for those two pages, ideally tuned to the specific needs of the audience that will arrive at them. Perhaps the pizza preferences of the San Jose crowd are different from those in New York City. And of course, the geographic information is inherently different between the two locations, so driving directions from key locations might be a good thing to include on each page.

If you have pizza parlors in 100 cities, this can get very complex indeed. The key here is to remain true to the diverse needs of your users, yet use your knowledge of the needs of search engines and searcher behavior to obtain that long-tail traffic.

Content Optimization

Content optimization relates to how the presentation and architecture of the text, image, and multimedia content on a page can be optimized for search engines. Many of these recommendations are second-order effects. Having the right formatting or display won't boost

your rankings directly, but through it, you're more likely to earn links, get clicks, and eventually benefit in search rankings. If you regularly practice the techniques in this section, you'll earn better consideration from the engines and from the human activities on the Web that influence their algorithms.

Content Structure

Because SEO has become such a holistic part of website development and improvement, it is no surprise that *content formatting*—the presentation, style, and layout choices you select for your content—is a part of the process. Choosing browser-safe sans serif fonts such as Arial and Helvetica is a wise choice for the Web; Verdana in particular has received high praise from usability/readability experts, such as that WebAIM offered in an article posted at *http://webaim.org/techniques/fonts/*.

Verdana is one of the most popular of the fonts designed for on-screen viewing. It has a simple, straightforward design, and the characters (or glyphs) are not easily confused. For example, the uppercase *I* and the lowercase *L* have unique shapes, unlike in Arial, in which the two glyphs may be easily confused (see Figure 6-23).

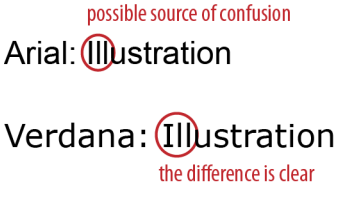

FIGURE 6-23. Arial versus Verdana font comparison

Another advantage of Verdana is the amount of spacing between letters. Conversely, one consideration to take into account with Verdana is that it is a relatively large font. The words take up more space than words in Arial, even at the same point size (see Figure 6-24).

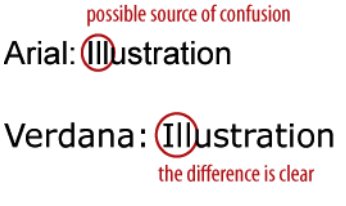

FIGURE 6-24. How fonts impact space requirements

The larger size improves readability but also has the potential of disrupting carefully planned page layouts.

Font choice is accompanied in importance by sizing and contrast issues. Type that is smaller than 10 points is typically very challenging to read, and in all cases, relative font sizes are recommended so that users can employ browser options to increase/decrease the size if necessary. Contrast—the color difference between the background and text—is also critical; legibility usually drops for anything that isn't black (or very dark) on a white background.

Content length and word count

Content length is another critical piece of the optimization puzzle that's mistakenly placed in the "keyword density" or "unique content" bucket of SEO. In fact, content length can have a big role to play in terms of whether your material is easy to consume and easy to share. Lengthy pieces often don't fare particularly well on the Web (with the exception, perhaps, of the one-page sales letter), whereas short-form and easily digestible content often has more success. Sadly, splitting long pieces into multiple segments frequently backfires, as abandonment increases while link attraction decreases. The only benefit is in the number of page views per visit (which is why many sites that get their revenue from advertising employ this tactic).

Visual layout

Last but not least in content structure optimization is the display of the material. Beautiful, simplistic, easy-to-use, and consumable layouts instill trust and garner far more readership and links than poorly designed content wedged between ad blocks that threaten to overtake the page. For more on this topic, you might want to check out "The Golden Ratio in Web Design" from NetTuts (*http://net.tutsplus.com/tutorials/other/the-golden-ratio-in-web-design/*), which has some great illustrations and advice on laying out web content on the page.

CSS and Semantic Markup

Cascading Style Sheets (CSS) is commonly mentioned as a best practice for general web design and development, but its principles provide some indirect SEO benefits as well. Google used to recommend keeping pages smaller than 101 KB, and it used to be a common belief that there were benefits to implementing pages that were small in size. Now, however, search engines deny that code size is a factor at all, unless it is really extreme. Still, keeping file size low means faster load times, lower abandonment rates, and a higher probability of the page being fully read and more frequently linked to.

CSS can also help with another hotly debated issue: code-to-text ratio. Some SEO professionals (even among the authors, opinions vary) swear that making the code-to-text ratio smaller (so there's less code and more text) can help considerably on large websites with many thousands of pages. Your experience may vary, but since good CSS makes it easy, there's no reason not to make it part of your standard operating procedure for web development. Use table-less CSS stored in external files, keep JavaScript calls external, and separate the content layer from the

presentation layer, as shown on CSS Zen Garden, a site that offers many user-contributed stylesheets formatting the same HTML content.

You can use CSS code to provide emphasis, to quote/reference, and to reduce the use of tables and other bloated HTML mechanisms for formatting, which can make a positive difference in your SEO.

Schema.org and Microformats

In June 2011, Google, Bing, and Yahoo! came together to announce a new standard for markup called Schema.org. You can see a copy of the announcement at *http://googleblog.blogspot.com/2011/06/introducing-schemaorg-search-engines.html*. This represented a new level of commitment by the search engines to the concept of marking up content, or more broadly, allowing the publisher to provide information about the content to the search engines. When we refer to "marking up content," we are referring to the concept of tagging your content using XML tags that categorize the contents of a block of content. For example, you may label one block of content as containing a recipe and another as containing a review.

This notion of advanced markup was not new, as all of the search engines have supported semantic markup at a limited level for some time and have used this markup to show rich snippets, as described below.

One of the original ways publishers had to communicate information about a web page to search engines was with meta tags. Unfortunately, these were so badly abused by spammers that Google stopped using them as a ranking signal. Google confirmed this publicly in a post in 2009 that noted that "Google has ignored the keywords meta tag for years and currently we see no need to change that policy" (*http://googlewebmastercentral.blogspot.com/2009/09/google-does-not-use-keywords-meta-tag.html*).

Google continues to indicate that markup is not used as a ranking signal: "Google doesn't use markup for ranking purposes at this time" (*http://www.google.com/support/webmasters/bin/answer.py?answer=1211158*). However, there are important SEO benefits to using markup.

Markup in search results

As mentioned above, search engines sometimes use markup to create *rich snippets*. Figure 6-25 shows an example of rich snippets in the search results, returned for a search on a recipe for a Cambodian dish called Loc Lac.

Based on the markup that Google found in the HTML, it has enhanced the results by showing information such as the average rating by reviewers (the number of stars), the required cooking time, and the number of calories in the meal. The type of markup used for this example is called *microformats*. Figure 6-26 shows what the source code looks like for this example.

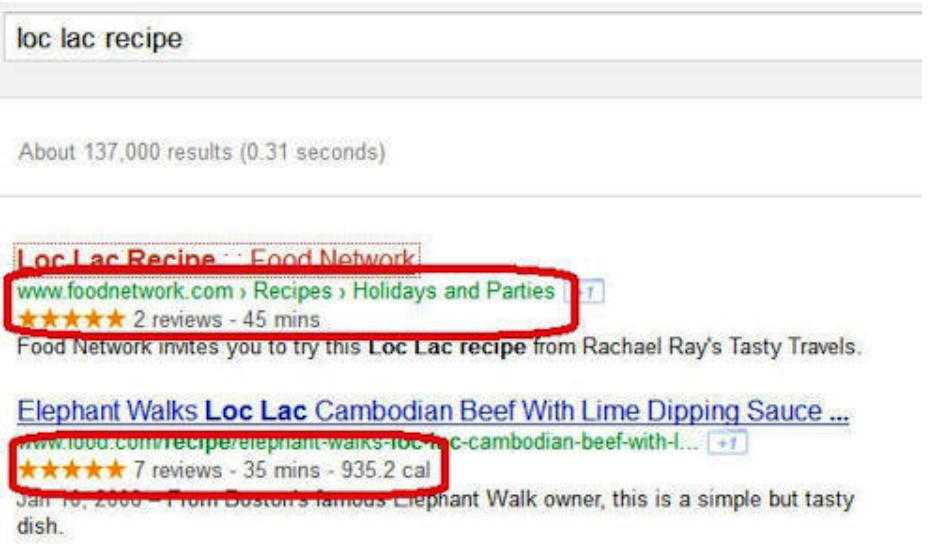

FIGURE 6-25. Example of a recipe rich snippet on Google

```
<div id="fn-w" class="hrecipe"><div class="breadcrumb clrfix">
  <div class="bc-links">
    <a href="/">Home</a>
    <span>&gt;</span>
    <a href="/recipes/index.html">Recipes</a>
    <span>&gt;</span>
    <a href="/recipe-collections/holidays-and-parties/index.html">Holidays and Parties</a>
    <span>&gt;</span>
  </div>
  <div class="bc-desc">Loc Lac Recipe</div>
</div>
```

FIGURE 6-26. Sample of microformats code for a recipe

Supported types of markup

There are a few different standards for markup. The most common ones are *microdata*, *microformats*, and *RDFa*. Schema.org is based on the microdata standard. However, the search engines have implemented rich snippets, based on some (but not all) aspects of microformats, prior to the announcement of Schema.org, and they will likely continue to support these for some period of time.

It is likely that any new forms of rich snippets implemented by the search engines will be based on Schema.org (microdata) and not microformats or RDFa. Some of the formats already supported by Google include:

- People: *http://www.google.com/support/webmasters/bin/answer.py?answer=146646*
- Products: *http://www.google.com/support/webmasters/bin/answer.py?answer=146750#product_properties*

- Events: *http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=164506*
- Business and organizations: *http://www.google.com/support/webmasters/bin/answer.py?answer=146861*
- Video: *http://www.google.com/support/webmasters/bin/answer.py?answer=162163*

In June 2011, Google also announced support for the `rel="author"` tag. This is a form of markup that identifies the author of a piece of content. When Google sees this tag it may choose to place the author's picture in the search results next to search listings for the articles that person has written. Figure 6-27 shows what this looks like in the search results.

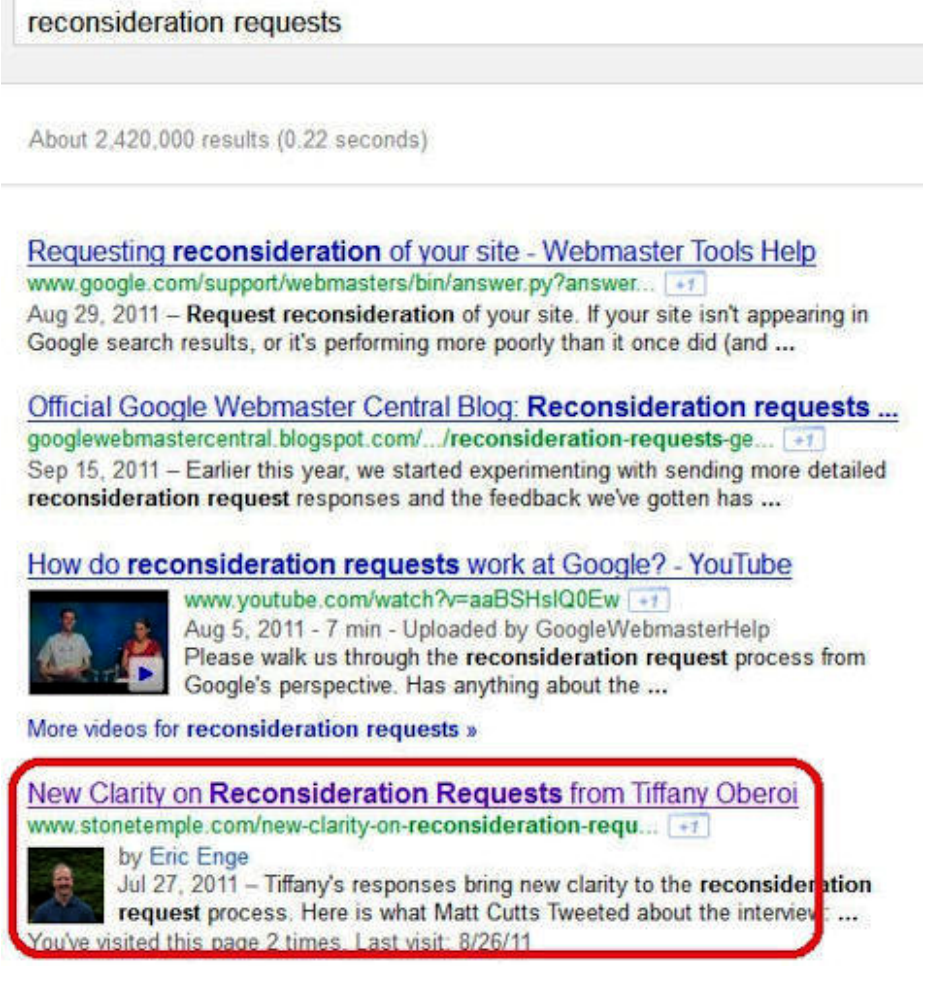

FIGURE 6-27. rel="author" rich snippet in Google results

Google originally required you to implement a fairly complex amount of HTML to recognize you as an author. You can learn how to implement this using the `rel="author"` HTML tag in this article by A.J. Kohn: *http://www.blindfiveyearold.com/how-to-implement-rel-author*. However, you can also accomplish the same thing using an email verification process, as documented at *http://www.google.com/support/webmasters/bin/answer.py?answer=1408986*.

Be aware that inclusion of your image is not automatic. It only happens for certain authors, and some time is required after implementation before the image shows up in the search results. However, setting this up is not hard, and Google is likely to expand the number of authors that are shown over time.

Impact of rich snippets

The key reason that the search engines are pursuing rich snippets is that they have done extensive testing that has shown them that they increase click-through rates. Searchers like seeing more information about the page in the search results. Based on this, you can expect that they will continue to implement support for more of these types of search result enhancements based on markup.

From an SEO perspective, increasing click-through rate is highly desirable: it brings more relevant traffic to the site. In addition, we know that search engines measure user interaction with the search results and that click-through rate is a ranking factor. This was first publicly confirmed in an interview that Bing's Duane Forrester did with Eric Enge, which you can view at *http://www.stonetemple.com/search-algorithms-and-bing-webmaster-tools-with-duane-forrester*.

So, while the search engines do not directly use semantic markup as a ranking signal, the indirect impact of rich snippets providing a higher click-through rate does act as a ranking signal.

Content Uniqueness and Depth

Few can debate the value the engines place on robust, unique, value-added content—Google in particular has had several rounds of kicking "low-quality-content" sites out of its indexes, and the other engines have followed suit.

The first critical designation to avoid is "thin content"—an insider phrase that (loosely) refers to content the engines do not feel contributes enough unique material for a page to merit a high ranking (or even any inclusion) in the search results. How much content is enough content to not be considered thin? The criteria have never been officially listed, but many examples/discussions from engineers and search engine representatives place the following on the list:

- At least 30 to 50 unique words, forming unique, parsable sentences that other sites/pages do not have (for many pages much more is appropriate; consider this a minimum).

- Unique HTML text content that differs from that of other pages on the site in more than just the replacement of key verbs and nouns (yes, this means all those sites that build the same page and just change the city and state names, thinking this makes them "unique," are mistaken).
- Unique titles and meta description elements. If you can't write unique meta descriptions, just exclude them. The search engine algorithms may boot pages from the index simply for having near-duplicate meta tags.
- Unique video/audio/image content. The engines have started getting smarter about identifying and indexing for vertical search pages that wouldn't normally meet the "uniqueness" criteria.

> NOTE
>
> **You can often bypass these limitations if you have a good quantity of high-value external links pointing to the page in question (though this is very rarely scalable), or an extremely powerful, authoritative site (note how many one-sentence Wikipedia stub pages still rank).**

The next criterion from the engines demands that websites "add value" to the content they publish, particularly if it comes (wholly or partially) from a secondary source.

A word of caution to affiliates

This word of caution most frequently applies to affiliate sites whose republishing of product descriptions, images, and so forth has come under search engine fire numerous times. In fact, it is best to anticipate manual evaluations here even if you've dodged the algorithmic sweep. The basic tenets are:

- Don't simply republish something that's found elsewhere on the Web unless your site adds substantive value to users, and don't infringe on others' copyrights or trademarks.
- If you're hosting affiliate content, expect to be judged more harshly than others, as affiliates in the SERPs are one of users' top complaints about search engines.
- Small changes such as a few comments, a clever sorting algorithm or automated tags, filtering, a line or two of text, simple mashups, or advertising do *not* constitute "substantive value."

For some exemplary cases where websites fulfill these guidelines, check out the way sites such as CNET (*http://reviews.cnet.com*), Urbanspoon (*http://www.urbanspoon.com*), and Metacritic (*http://www.metacritic.com*) take content/products/reviews from elsewhere, both aggregating *and* "adding value" for their users.

Last but not least, Google has provided a guideline to refrain from trying to place "search results in the search results." For reference, look at the post from Google's Matt Cutts at

http://www.mattcutts.com/blog/search-results-in-search-results/. Google's stated feeling is that search results generally don't "add value" for users, though others have made the argument that this is merely an anticompetitive move.

Sites can benefit from having their "search results" transformed into "more valuable" listings and category/subcategory landing pages. Sites that have done this have had great success recovering rankings and gaining traffic from Google.

In essence, you want to avoid the potential for your site pages being perceived, both by an engine's algorithm and by human engineers and quality raters, as search results. Refrain from:

- Pages labeled in the title or headline as "search results" or "results"
- Pages that appear to offer a query-based list of links to "relevant" pages on the site without other content (add a short paragraph of text, an image, and formatting that makes the "results" look like detailed descriptions/links instead)
- Pages whose URLs appear to carry search queries (e.g., *?q=miami+restaurants* or *?search=Miami+restaurants* versus */miami-restaurants*)
- Pages with text such as "Results 1 through 10"

Though it seems strange, these subtle, largely cosmetic changes can mean the difference between inclusion in and removal from the index. Err on the side of caution and dodge the appearance of presenting search results.

Content Themes

A less-discussed but still important issue is the fit of each piece of content to your site. If you create an article about pizza, but the rest of your site is about horseshoes, your article is unlikely to rank for the keyword *pizza*. Search engines do analyze and understand what sites, or sections of sites, focus on for topic matter.

You can think of this as being the "theme" of the site (or section). If you start creating content that is not on the same theme, that content will have a very difficult time ranking. Further, your off-topic content could potentially weaken the theme of the rest of the site.

One site can support multiple themes, but each themed section needs to justify its own existence by following good SEO practices, including getting third parties to implement links from the pages of their sites to that section. Make sure you keep your content on topic; this will help the SEO for all of the pages of your site.

Copyblogger has created a tool to help measure the fit of a given article to your site, known as Scribe (*http://www.copyblogger.com/scribe-seo/*). In addition, Scribe will offer a more general look at the consistency of the content across your site as a whole.

Duplicate Content Issues

Duplicate content can result from many causes, including licensing of content to or from your site, site architecture flaws due to non-SEO-friendly CMSs, or plagiarism. Over the past five years, however, spammers in desperate need of content have begun the now much-reviled process of scraping content from legitimate sources, scrambling the words (through many complex processes), and repurposing the text to appear on their own pages in the hopes of attracting long-tail searches and serving contextual ads (and various other nefarious purposes).

Thus, today we're faced with a world of "duplicate content issues" and "duplicate content penalties." Here are some definitions that are useful for this discussion:

Unique content
: This is content that is written by humans; is completely different from any other combination of letters, symbols, or words on the Web; and has clearly not been manipulated through computer text-processing algorithms (such as Markov chain–employing spam tools).

Snippets
: These are small chunks of content such as quotes that are copied and reused; these are almost never problematic for search engines, especially when included in a larger document with plenty of unique content.

Shingles
: Search engines look at relatively small phrase segments (e.g., five to six words), checking for the presence of the same segments on other pages on the Web. When there are too many "shingles" in common between two documents, the search engines may interpret them as duplicate content.

Duplicate content filter
: This is when the search engine removes substantially similar content from a search result to provide a better overall user experience. This is by far the most common action taken by a search engine when it detects duplicate content. Search engines recognize that there are many reasons why duplicate content may occur that are not the result of malicious intent, and they simply look to filter out the copies.

Duplicate content penalty
: Penalties are applied rarely and only in egregious situations. Engines may devalue or ban other web pages on the site, too, or even the entire website.

Consequences of Duplicate Content

Assuming your duplicate content is a result of innocuous oversights on your developer's part, the search engine will most likely simply filter out all but one of the pages that are duplicates, because the search engine only wants to display one version of a particular piece of content in a given SERP. In some cases, the search engine may filter out results prior to including them

in the index, and in other cases the search engine may allow a page in the index and filter it out when it is assembling the SERPs in response to a specific query. In this latter case, a page may be filtered out in response to some queries and not others.

Searchers want diversity in the results, not the same results repeated again and again. Search engines therefore try to filter out duplicate copies of content, and this has several consequences:

- A search engine bot comes to a site with a *crawl budget*, which is counted in terms of the number of pages it plans to crawl in each particular session. Each time it crawls a page that is a duplicate (which is simply going to be filtered out of search results), you have let the bot waste some of its crawl budget. That means fewer of your "good" pages will get crawled. This can result in fewer of your pages being included in the search engine index.
- Even though search engines attempt to filter out duplicate content, links to pages with duplicated content still pass link juice to those pages. Duplicated pages can therefore gain PageRank or link juice, but since it does not help them rank, that resource has been misspent.
- No search engine has offered a clear explanation for how its algorithm picks which version of a page to show. In other words, if it discovers three copies of the same content, which two does it filter out, and which one does it show? Does it vary based on the search query? The bottom line is that the search engine might not favor the version you want.

Although some SEO professionals may debate some of the preceding specifics, the general structure will meet with near-universal agreement. However, there are a couple of problems around the edge of this model.

For example, on your site you may have a bunch of product pages and also offer print versions of those pages. The search engine might pick just the printer-friendly page as the one to show in its results. This does happen at times, and it can happen even if the printer-friendly page has lower link juice and will rank less well than the main product page.

One fix for this is to apply the `canonical` URL tag to all the duplicate versions of the page, pointing back to the master copy.

A second version of this can occur when you syndicate content to third parties. The problem is that the search engine may boot your copy of the article out of the results in favor of the version in use by the person republishing your article. The best fix for this, other than `NoIndex`ing the copy of the article that your partner is using, is to have the partner implement a link back to the original source page on your site. Search engines nearly always interpret this correctly and emphasize your version of the content when you do that.

How Search Engines Identify Duplicate Content

Some examples will illustrate the process for Google as it finds duplicate content on the Web. In the examples shown in Figures 6-28 through 6-31, three assumptions have been made:

- The page with text is assumed to be a page containing duplicate content (not just a snippet, despite the illustration).
- Each page of duplicate content is presumed to be on a separate domain.
- The steps that follow have been simplified to make the process as easy and clear as possible. This is almost certainly not the exact way in which Google performs (but it conveys the effect).

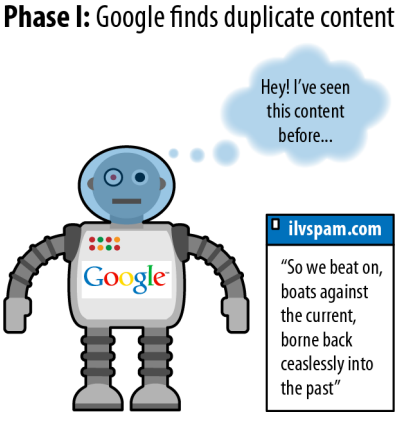

FIGURE 6-28. Google finding duplicate content

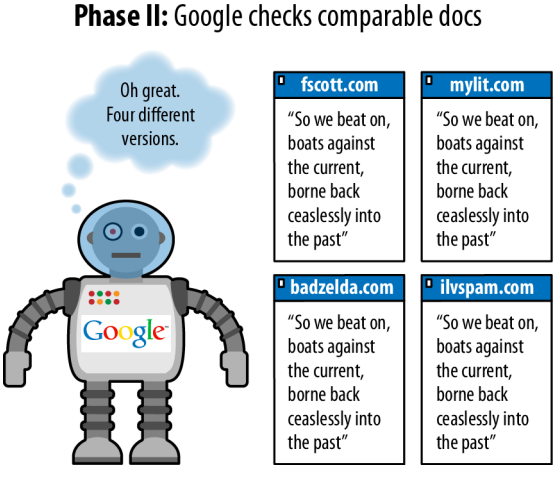

FIGURE 6-29. Google comparing the duplicate content to the other copies

There are a few facts about duplicate content that bear mentioning, as they can trip up webmasters who are new to the duplicate content issue:

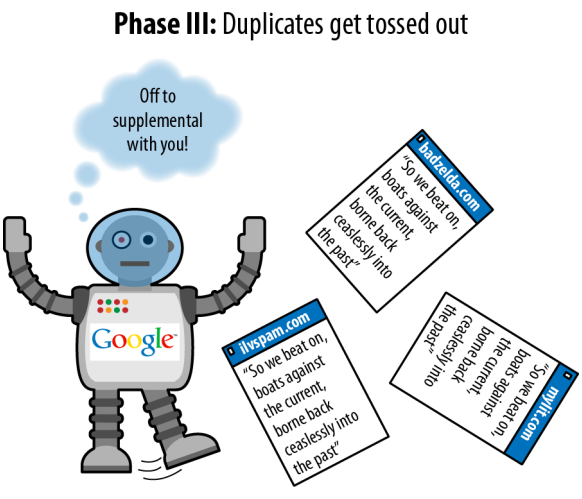

FIGURE 6-30. Duplicate copies getting tossed out

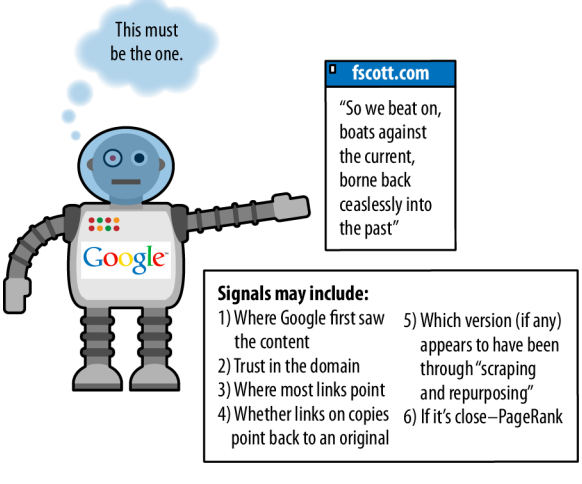

FIGURE 6-31. Google choosing one as the original

Location of the duplicate content

Is it duplicated content if it is all on my site? Yes, in fact, duplicate content can occur within a site or across different sites.

Percentage of duplicate content

What percentage of a page has to be duplicated before you run into duplicate content filtering? Unfortunately, the search engines would never reveal this information because it would compromise their ability to prevent the problem.

It is also a near certainty that the percentage at each engine fluctuates regularly and that more than one simple direct comparison goes into duplicate content detection. The bottom line is that pages do not need to be identical to be considered duplicates.

Ratio of code to text

What if your code is huge and there are very few unique HTML elements on the page? Will Google think the pages are all duplicates of one another? No. The search engines do not really care about your code; they are interested in the content on your page. Code size becomes a problem only when it becomes extreme.

Ratio of navigation elements to unique content

Every page on my site has a huge navigation bar, lots of header and footer items, but only a little bit of content; will Google think these pages are duplicates? No. Google (and Yahoo! and Bing) factors out the common page elements such as navigation before evaluating whether a page is a duplicate. It is very familiar with the layout of websites and recognizes that permanent structures on all (or many) of a site's pages are quite normal. Instead, it pays attention to the "unique" portions of each page and often will largely ignore the rest.

Licensed content

What should I do if I want to avoid duplicate content problems, but I have licensed content from other web sources to show my visitors? Use `meta name = "robots" content="noindex, follow"`. Place this in your page's header and the search engines will know that the content isn't for them. This is a general best practice, because then humans can still visit the page and link to it, and the links on the page will still carry value.

Another alternative is to make sure you have exclusive ownership and publication rights for that content.

Identifying and Addressing Copyright Infringement

One of the best ways to monitor whether your site's copy is being duplicated elsewhere is to use CopyScape.com, a site that enables you to instantly view pages on the Web that are using your content. Do not worry if the pages of these sites are in the supplemental index or rank far behind your own pages for any relevant queries—if any large, authoritative, content-rich domain tried to fight all the copies of its work on the Web, it would have at least two 40-hour-per-week jobs on its hands. Luckily, the search engines have placed trust in these types of sites to issue high-quality, relevant, worthy content, and therefore recognize them as the original issuers.

If, on the other hand, you have a relatively new site or a site with few inbound links, and the scrapers are consistently ranking ahead of you (or if someone with a powerful site is stealing

your work), you've got some recourse. One option is to file a Digital Millennium Copyright Act (DMCA) infringement request with Google, with Yahoo!, and with Bing (you should also file this request with the site's hosting company).

The other option is to file a legal suit (or threaten such) against the website in question. If the site republishing your work has an owner in your country, this course of action is probably the wisest first step. You may want to try to start with a more informal communication asking them to remove the content before you send a letter from the attorneys, as the DMCA motions can take months to go into effect; but if they are nonresponsive, there is no reason to delay taking stronger action.

An actual penalty situation

The preceding examples have to do with duplicate content filters and not actual penalties (although, for all practical purposes, they have the same impact as a penalty: lower rankings for your pages). However, there are also scenarios where an actual penalty can occur.

For example, sites that aggregate content from across the Web can be at risk, particularly if those sites themselves add little unique content. In this type of scenario, you might see the site actually penalized.

The only fixes for this are to reduce the number of duplicate pages accessible to the search engine crawler, either by deleting them or `NoIndex`ing the pages themselves, or to add a substantial amount of unique content.

One example of duplicate content that may get filtered out on a broad basis is that on a *thin affiliate* site. This nomenclature frequently describes a site that promotes the sale of someone else's products (to earn a commission), yet provides little or no information that differentiates it from other sites selling the product. Such a site may have received the descriptions from the manufacturer of the products and simply replicated those descriptions along with an affiliate link (so that it can earn credit when a click/purchase is made).

The problem arises when a merchant has thousands of affiliates generally promoting websites using the same descriptive content, and search engineers have observed user data suggesting that, from a searcher's perspective, these sites add little value to their indexes. Thus, the search engines attempt to filter out this type of site, or even ban it from their indexes. Plenty of sites operate affiliate models but also provide rich new content, and these sites generally have no problem. It is when duplication of content and a lack of unique, value-adding material come together on a domain that the engines may take action.

How to Avoid Duplicate Content on Your Own Site

As we outlined earlier, duplicate content can be created in many ways. Internal duplication of material requires specific tactics to achieve the best possible results from an SEO perspective. In many cases, the duplicate pages are pages that have no value to either users or search

engines. If that is the case, try to eliminate the problem altogether by fixing the implementation so that all pages are referred to by only one URL. Also, 301-redirect the old URLs to the surviving URLs (as discussed in more detail in "Redirects" on page 262) to help the search engines discover what you have done as rapidly as possible, and preserve any link juice the removed pages may have had.

If that process proves to be impossible, there are many options, as we will outline in "Content Delivery and Search Spider Control" on page 245. Here is a summary of the guidelines on the simplest solutions for dealing with a variety of scenarios:

- Use the `canonical` tag. This is the next best solution to eliminating the duplicate pages.
- Use *robots.txt* to block search engine spiders from crawling the duplicate versions of pages on your site.
- Use the `robots NoIndex` meta tag to tell the search engine not to index the duplicate pages.

Be aware, however, that if you use *robots.txt* to prevent a page from being crawled, using `NoIndex` or `NoFollow` on the page itself will not make sense: the spider can't read the page, so it will never see the `NoIndex` or `NoFollow` tag.

With these tools in mind, let's look at some specific duplicate content scenarios:

HTTPS pages
: If you make use of *SSL* (encrypted communications between the browser and the web server, often used for ecommerce purposes), you will have pages on your site that begin with *https:* instead of *http:*. The problem arises when the links on your *https:* pages link back to other pages on the site using relative instead of absolute links, so (for example) the link to your home page becomes *https://www.yourdomain.com* instead of *http://www.yourdomain.com*.

  If you have this type of issue on your site, you may want to use the `canonical` URL tag, which we describe in "Content Delivery and Search Spider Control" on page 245, or 301 redirects to resolve problems with these types of pages. An alternative solution is to change the links to absolute links (e.g., *http://www.yourdomain.com/content.html* instead of */content.html*), which also makes life more difficult for content thieves that scrape your site.

CMSs that create duplicate content
: Sometimes sites have many versions of identical pages because of limitations in the CMS, where it addresses the same content with more than one URL. These are often unnecessary duplications with no end user value, and the best practice is to figure out how to eliminate the duplicate pages and 301 the eliminated pages to the surviving pages. Failing that, fall back on the other options listed at the beginning of this section.

Print pages or multiple sort orders
: Many sites offer "print" pages to provide the user with the same content in a more printer-friendly format, and some ecommerce sites offer lists of their products in multiple

sort orders (such as by size, color, brand, and price). These pages do have end user value, but they do not have value to the search engine and will appear to be duplicate content. For that reason, use one of the options listed previously in this subsection, or set up a print CSS stylesheet such as the one outlined in this post by Yoast: *http://yoast.com/added-print-css-style-sheet/*.

Duplicate content in blogs and multiple archiving systems (pagination, etc.)
: Blogs present some interesting duplicate content challenges. Blog posts can appear on many different pages, such as the home page of the blog, the permalink page for the post, date archive pages, and category pages. Each instance of the post represents a duplicate of the other instances. Few publishers attempt to address the presence of the post on the home page of the blog and also at its permalink, and this is common enough that it is likely that the search engines deal reasonably well with it. However, it may make sense to show only snippets of the post on the category and/or date archive pages.

User-generated duplicate content (repostings, etc.)
: Many sites implement structures for obtaining user-generated content, such as a blog, forum, or job board. This can be a great way to develop large quantities of content at a very low cost. The challenge is that users may choose to submit the same content on your site and on several other sites at the same time, resulting in duplicate content among those sites. It is hard to control this, but there are two things you can do to reduce the problem:

  - Have clear policies that notify users that the content they submit to your site must be unique and cannot be, or cannot have been, posted to other sites. This is difficult to enforce, of course, but it will still help some to communicate your expectations.
  - Implement your forum in a different and unique way that demands different content. Instead of having only the standard fields for entering data, include fields that are likely to be unique with respect to what other sites do, but that will still be interesting and valuable for site visitors to see.

Controlling Content with Cookies and Session IDs

Sometimes you want to more carefully dictate what a search engine robot sees when it visits your site. In general, search engine representatives will refer to the practice of showing different content to users than crawlers as *cloaking*, which violates the engines' Terms of Service (TOS) and is considered spammy behavior.

However, there are legitimate uses for this concept that are not deceptive to the search engines or malicious in intent. This section will explore methods for doing this with cookies and session IDs.

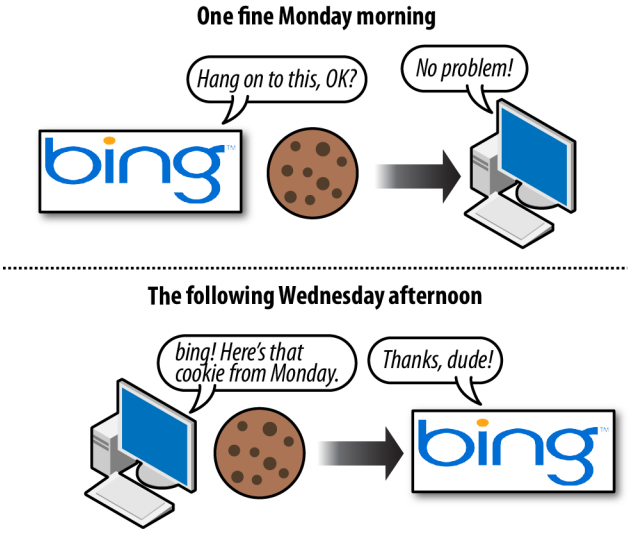

FIGURE 6-32. Using cookies to store data

What's a Cookie?

A *cookie* is a small text file that websites can leave on a visitor's hard disk, helping them to track that person over time. Cookies are the reason Amazon.com remembers your username between visits and the reason you don't necessarily need to log in to your Hotmail account every time you open your browser. Cookie data typically contains a short set of information regarding when you last accessed a site, an ID number, and, potentially, information about your visit (see Figure 6-32).

Website developers can create options to remember visitors using cookies for tracking purposes or to display different information to users based on their actions or preferences. Common uses include remembering a username, maintaining a shopping cart, and keeping track of previously viewed content. For example, if you've signed up for an account with SEOmoz, it will provide you with options on your My Account page about how you want to view the blog and will remember your settings the next time you visit.

What Are Session IDs?

Session IDs are virtually identical to cookies in functionality, with one big difference: when you close your browser (or restart), session ID information is (usually) no longer stored on your hard drive. Figure 6-33 illustrates. The website you were interacting with may remember your data or actions, but it cannot retrieve session IDs from your machine that don't persist (and session IDs by default expire when the browser shuts down). In essence, session IDs are like temporary cookies (although, as you'll see shortly, there are options to control this).

FIGURE 6-33. How session IDs are used

Although technically speaking, session IDs are just a form of cookie without an expiration date, it is possible to set session IDs with expiration dates similar to cookies (going out decades). In this sense, they are virtually identical to cookies. Session IDs do come with an important caveat, though: they are frequently passed in the URL string, which can create serious problems for search engines (as every request produces a unique URL with duplicate content). A simple fix is to use the `canonical` tag (which we'll discuss in "Content Delivery and Search Spider Control" on page 245) to tell the search engines that you want them to ignore the session IDs.

NOTE

Any user has the ability to turn off cookies in his browser settings. This often makes web browsing considerably more difficult, though, and many sites will actually display a page saying that cookies are required to view or interact with their content. Cookies, persistent though they may be, are also deleted by users on a semiregular basis. For example, a 2011 comScore study (*http://www.websitemagazine.com/content/blogs/posts/archive/2011/05/09/the-impact-of-cookie-deletion.aspx*) found that 33% of web users deleted their first-party cookies at least once per month.

How Do Search Engines Interpret Cookies and Session IDs?

They don't. Search engine spiders are not built to maintain or retain cookies or session IDs and act as browsers with this functionality shut off. However, unlike visitors whose browsers won't accept cookies, the crawlers can sometimes reach sequestered content by virtue of webmasters who want to specifically let them through. Many sites have pages that require cookies or sessions to be enabled but have special rules for search engine bots, permitting them to access the content as well. Although this is technically cloaking, there is a form of this known as First Click Free that search engines generally allow (we will discuss this in more detail in "Content Delivery and Search Spider Control" on page 245).

Despite the occasional access engines are granted to cookie/session-restricted pages, the vast majority of cookie and session ID usage creates content, links, and pages that limit access. Web developers can leverage the power of concepts such as First Click Free to build more intelligent sites and pages that function in optimal ways for both humans and engines.

Why Would You Want to Use Cookies or Session IDs to Control Search Engine Access?

There are numerous potential tactics to leverage cookies and session IDs for search engine control. Here are some of the major strategies you can implement with these tools, but there are certainly limitless other possibilities:

Showing multiple navigation paths while controlling the flow of link juice

Visitors to a website often have multiple ways in which they'd like to view or access content. Your site may benefit from offering many paths to reaching content (by date, topic, tag, relationship, ratings, etc.), but doing so expends PageRank or link juice that would be better optimized by focusing on a single, search engine–friendly navigational structure. This is important because these varied sort orders may be seen as duplicate content.

You can require a cookie for users to access the alternative sort order versions of a page, thereby preventing the search engine from indexing multiple pages with the same content. One alternative solution to this is to use the `canonical` tag to tell the search engine that

these alternative sort orders are really just the same content as the original page (we will discuss the `canonical` tag in "Content Delivery and Search Spider Control" on page 245).

Keeping limited pieces of a page's content out of the engines' indexes

Many pages may contain some content that you'd like to show to search engines, and other pieces you'd prefer to appear only for human visitors. These could include ads, login-restricted information, links, or even rich media. Once again, showing noncookied users the plain version and cookie-accepting visitors the extended information can be invaluable. Note that this approach is often used in conjunction with a login, so only registered users can access the full content (such as on sites like Facebook and LinkedIn).

Granting access to pages requiring a login

As with snippets of content, there are often entire pages or sections of a site to which you'd like to restrict search engine access. This can be easy to accomplish with cookies/sessions, and it can even help to bring in search traffic that may convert to "registered-user" status. For example, if you had desirable content that you wished to restrict access to, you could create a page with a short snippet and an offer to continue reading upon registration, which would then allow full access to that work at the same URL. We will discuss this more in "Content Delivery and Search Spider Control" on page 245.

Avoiding duplicate content issues

One of the most promising areas for cookie/session use is to prohibit spiders from reaching multiple versions of the same content, while allowing visitors to get the version they prefer. As an example, at SEOmoz, logged-in users can see full blog entries on the blog home page, but search engines and nonregistered users will see only the snippets. This prevents the content from being listed on multiple pages (the blog home page and the specific post pages), and provides a positive user experience for members.

Content Delivery and Search Spider Control

On occasion, it can be valuable to show search engines one version of content and show humans a different version. This is technically called "cloaking," and the search engines' guidelines have near-universal policies restricting this behavior. In practice, many websites, large and small, appear to use some forms of cloaking without being penalized by the search engines. However, use great care if you implement these techniques, and know the risks that you are taking.

Cloaking and Segmenting Content Delivery

Before we discuss the risks and potential benefits of cloaking-based practices, take a look at Figure 6-34, which shows an illustration of how cloaking works.

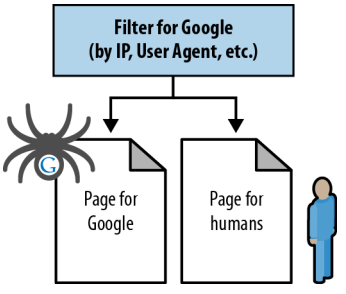

FIGURE 6-34. How cloaking works

Matt Cutts, head of Google's webspam team, has made strong public statements indicating that all forms of cloaking (with the only exception being First Click Free) are subject to penalty. This position was backed up by statements by Google's John Mueller in a May 2009 interview, which you can read at *http://www.stonetemple.com/articles/interview-john-mueller.shtml*, and Cutts confirmed it again in August 2011 in this video on YouTube: *http://www.youtube.com/watch?feature=player_embedded&v=QHtnfOgp65Q*. In the video, Matt Cutts makes the strong statement, "There is no such thing as white-hat cloaking."

Google also makes its policy pretty clear in its "Guidelines on Cloaking" (*http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=66355*):

> Serving up different results based on user agent may cause your site to be perceived as deceptive and removed from the Google index.

There are two critical pieces in the preceding quote: *may* and *user agent*. It is true that if you cloak in the wrong ways, with the wrong intent, Google and the other search engines *may* remove you from their indexes, and if you do it egregiously, they certainly *will*.

A big factor is intent: if the engines feel you are attempting to manipulate their rankings or results through cloaking, they may take adverse action against your site. If, however, the intent of your content delivery doesn't interfere with their goals, you're less likely to be subject to a penalty. Still, there is never no risk of a penalty. Google has taken a strong stand against all forms of cloaking, regardless of intent.

The following are some examples of websites that perform some level of cloaking:

Google.com
: Search for *google toolbar*, *google translate*, *adwords*, or any number of other Google properties, and note how the URL you see in the search results and the one you land on almost never match. What's more, on many of these pages, whether you're logged in or not, you might see some content that is different from what's in the cache.

NYTimes.com

The interstitial ads, the request to log in/create an account after five clicks, and the archive inclusion are all showing different content to engines versus humans.

Wine.com

In addition to some redirection based on your path, there's a state overlay forcing you to select a shipping location prior to seeing any prices (or any pages). That's a form the engines don't have to fill out.

Yelp.com

Geotargeting through cookies based on location is a very popular form of local targeting that hundreds, if not thousands, of sites use.

Trulia.com

Trulia was found to be doing some interesting redirects on partner pages and its own site (*http://www.bramblog.com/trulia-caught-cloaking-red-handed/*).

The message should be clear. Cloaking won't always get you banned, and you can do some pretty smart things with it. The key to all of this is your intent. If you are doing it for reasons that are not deceptive and that provide a positive experience for users and search engines, you might not run into problems. However, there is no guarantee of this, so use these types of techniques with great care, and know that you may still get penalized for it.

When to Show Different Content to Engines and Visitors

There are a few common causes for displaying content differently to different visitors, including search engines. Here are some of the most common ones:

Multivariate and A/B split testing

Testing landing pages for conversions requires that you show different content to different visitors to test performance. In these cases, it is best to display the content using JavaScript/cookies/sessions and give the search engines a single, canonical version of the page that doesn't change with every new spidering (though this won't necessarily hurt you). Google offers software called Google Website Optimizer to perform this function.

Content requiring registration and First Click Free

If you force registration (paid or free) on users to view specific content pieces, it is best to keep the URL the same for both logged-in and non-logged-in users and to show a snippet (one to two paragraphs is usually enough) to non-logged-in users and search engines. If you want to display the full content to search engines, you have the option to provide some rules for content delivery, such as showing the first one or two pages of content to a new visitor without requiring registration, and then requesting registration after that grace period. This keeps your intent more honest, and you can use cookies or sessions to restrict human visitors while showing all the content to the engines.

In this scenario, you might also opt to participate in a specific program from Google called *First Click Free*, wherein websites can expose "premium" or login-restricted content to Google's spiders, as long as users who click from the engine's results are given the ability to view that first article for free. Many prominent web publishers employ this tactic, including the popular site Experts-Exchange.com.

To be specific, to implement First Click Free, the publisher must grant Googlebot (and presumably the other search engine spiders) access to all the content they want indexed, even if users normally have to log in to see the content. The user who visits the site will still need to log in, but the search engine spider will not have to do so. This will lead to the content showing up in the search engine results when applicable. If a user clicks on that search result, you must permit her to view the entire article (including all pages of a given article if it is a multiple-page article). Once the user clicks to look at another article on your site, you can require her to log in. Publishers can also limit the number of free accesses a user gets using this technique to five articles per day.

For more details, visit Google's First Click Free program page at *http://googlewebmastercentral.blogspot.com/2008/10/first-click-free-for-web-search.html* and *http://googlewebmastercentral.blogspot.com/2009/12/changes-in-first-click-free.html*.

Navigation unspiderable by search engines

If your navigation is in Flash, JavaScript, a Java application, or another format that the search engines may not be able to parse, you should consider showing search engines a version of your site that has spiderable, crawlable content in HTML. Many sites do this simply with CSS layers, displaying a human-visible, search-invisible layer and a layer for the engines (and for less-capable browsers, such as mobile browsers). You can also employ the `NoScript` tag for this purpose, although it is generally riskier, as many spammers have applied `NoScript` as a way to hide content. Adobe recently launched a portal on SEO and Flash (*http://www.adobe.com/devnet/seo.html*) and provides best practices that have been cleared by the engines to help make Flash content discoverable. Take care to make sure the content shown in the search-visible layer is substantially the same as it is in the human-visible layer.

Duplicate content

If a significant portion of a page's content is duplicated, you might consider restricting spider access to it by placing it in an iframe that's restricted by *robots.txt*. This ensures that you can show the engines the unique portion of your pages, while protecting against duplicate content problems. We will discuss this in more detail in the next section.

Different content for different users

At times you might target content uniquely to users from different geographies (such as different product offerings that are more popular in their respective areas), with different screen resolutions (to make the content fit their screen size better), or who entered your site from different navigation points. In these instances, it is best to have a "default" version

of content that's shown to users who don't exhibit these traits that you can show to search engines as well.

How to Display Different Content to Search Engines and Visitors

A variety of strategies exist to segment content delivery. The most basic is to serve content that is not meant for the engines in unspiderable formats (e.g., placing text in images, Flash files, plug-ins). You should not use these formats for the purpose of cloaking. You should use them only if they bring a substantial end user benefit (such as an improved user experience). In such cases, you may want to show the search engines the same content in a search spider–readable format. When you're trying to show the engines something you don't want visitors to see, you can use CSS formatting styles (preferably not `display:none`, as the engines may have filters to watch specifically for this), JavaScript, user-agent detection, cookies or session-based delivery, or perhaps IP delivery (showing content based on the visitor's IP address).

Be very wary when employing cloaking such as that we just described. The search engines expressly prohibit these practices in their guidelines, and though there may be some leeway based on intent and user experience (e.g., if your site is using cloaking to improve the quality of the user's experience, not to game the search engines), the engines do take these tactics seriously and may penalize or ban sites that implement them inappropriately or with the intention of manipulation.

The robots.txt file

This file is located on the root level of your domain (e.g., *http://www.yourdomain.com/robots.txt*), and it is a highly versatile tool for controlling what the spiders are permitted to access on your site. You can use *robots.txt* to:

- Prevent crawlers from accessing nonpublic parts of your website
- Block search engines from accessing index scripts, utilities, or other types of code
- Avoid the indexation of duplicate content on a website, such as "print" versions of HTML pages, or various sort orders for product catalogs
- Autodiscover XML Sitemaps

The *robots.txt* file must reside in the root directory, and the filename must be entirely in lowercase (*robots.txt*, not *Robots.txt* or other variations including uppercase letters). Any other name or location will not be seen as valid by the search engines. The file must also be entirely in text format (not in HTML format).

You can use the *robots.txt* file to instruct a search engine robot not to access certain pages on your site. Figure 6-35 illustrates what happens when the search engine robot sees a direction in *robots.txt* telling it not to crawl a web page.

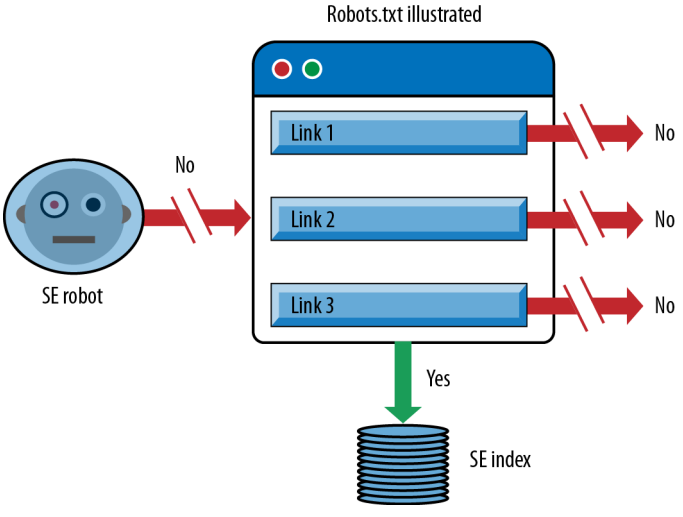

FIGURE 6-35. Impact of robots.txt

In essence, the page will not be crawled. This means links on the page cannot pass link juice to other pages, since the search engine does not see those links. However, the page can still be in the search engine index, if other pages on the Web link to it. Of course, the search engine will not have very much information on the page since it cannot read it, and it will rely mainly on the anchor text and other signals from the pages linking to it to determine what the page may be about. Any resulting search listings end up being pretty sparse when you see them in the Google index, as shown in Figure 6-36.

Figure 6-32 shows the results for the Google query *site:news.yahoo.com/topics/ inurl:page*. This is not a normal query that a user would enter, but you can see what the results look like. Only the URL is listed, and there is no description. This is because the spiders aren't permitted to read the page to get that data. In today's algorithms, these types of pages don't rank very high because their relevance scores tend to be quite low for any normal queries.

Google, Yahoo!, Bing, Ask, and nearly all of the legitimate crawlers on the Web will follow the instructions you set out in the *robots.txt* file. Commands in *robots.txt* are primarily used to prevent spiders from accessing pages and subfolders on a site, though they have other options as well. Note that subdomains require their own *robots.txt* files, as do files that reside on an *https:* server.

Syntax of the robots.txt file. The basic syntax of *robots.txt* is fairly simple. You specify a robot name, such as "googlebot," and then you specify an action. The robot is identified by user agent, and then the actions are specified on the lines that follow. Here are the major actions you can specify:

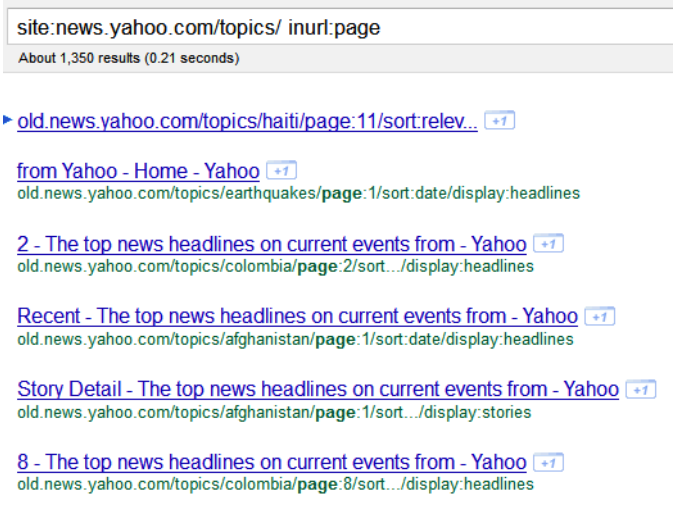

FIGURE 6-36. Search engine results for pages that are listed in robots.txt

`Disallow:`
: Use this for the pages you want to block the bot from accessing (you can include as many disallow lines as needed).

`Noindex:`
: Use this for the pages you want to block a search engine from including in its index (if previously indexed, this instruction tells the search engine to de-index the page).

Some other restrictions apply:

- Each `User-agent`/`Disallow` group should be separated by a blank line; however, no blank lines should exist within a group (between the `User-agent` line and the last `Disallow`).
- The hash symbol (`#`) may be used for comments within a *robots.txt* file, where everything after the `#` on that line will be ignored. This may be used either for whole lines or for the end of a line.
- Directories and filenames are case-sensitive: "private", "Private", and "PRIVATE" are all different to search engines.

Here is an example of a *robots.txt* file:

```
User-agent: Googlebot
Disallow:

User-agent: msnbot
Disallow: /

# Block all robots from tmp and logs directories
```

```
User-agent: *
Disallow: /tmp/
Disallow: /logs     # for directories and files called logs
```

The preceding example will do the following:

- Allow "Googlebot" to go anywhere.
- Prevent "msnbot" from crawling any part of the site.
- Block all robots (other than Googlebot) from visiting the */tmp/* directory or any directories or files called */logs* (e.g., */logs* or *logs.php*).

Notice that the behavior of Googlebot is not affected by instructions such as `Disallow: /tmp/`. Since Googlebot has its own instructions from *robots.txt*, it will ignore directives labeled as being for all robots (i.e., using an asterisk).

One common problem that novice webmasters run into occurs when they have SSL installed so that their pages may be served via HTTP and HTTPS. The search engines will not interpret a *robots.txt* file at *http://www.yourdomain.com/robots.txt* as guiding their crawl behavior on *https://www.yourdomain.com*. You will need to create an additional *robots.txt* file at *https://www.yourdomain.com/robots.txt*. So, if you want to allow crawling of all pages served from your HTTP server and prevent crawling of all pages from your HTTPS server, you would need to implement the following:

For HTTP:

```
User-agent: *
Disallow:
```

For HTTPS:

```
User-agent: *
Disallow: /
```

These are the most basic aspects of *robots.txt* files, but there are more advanced techniques as well. Some of these methods are supported by only some of the engines, as detailed in the list that follows:

Crawl delay

Crawl delay is supported by Yahoo!, Bing, and Ask. It instructs a crawler to wait the specified number of seconds between crawling pages. The goal of this directive is to reduce the load on the publisher's server. You can use it as follows:

```
User-agent: msnbot
Crawl-delay: 5
```

Pattern matching

Pattern matching appears to be supported by Google, Yahoo!, and Bing. The value of pattern matching is considerable. You can do some basic pattern matching using the asterisk wildcard character (*). Here is how you can use pattern matching to block access to all subdirectories that begin with *private* (e.g., */private1/*, */private2/*, */private3/*, etc.):

```
User-agent: Googlebot
Disallow: /private*/
```

You can match the end of the string using the dollar sign (`$`). For example, to block URLs that end with *.asp*:

```
User-agent: Googlebot
Disallow: /*.asp$
```

You may wish to prevent the robots from accessing any URLs that contain parameters. To block access to all URLs that include a question mark (?), simply use the question mark:

```
User-agent: *
Disallow: /*?*
```

The pattern-matching capabilities of *robots.txt* are more limited than those of programming languages such as Perl, so the question mark does not have any special meaning and can be treated like any other character.

`Allow` *directive*

The `Allow` directive appears to be supported only by Google, Yahoo!, and Ask. It works opposite to the `Disallow` directive and provides the ability to specifically call out directories or pages that may be crawled. When this is implemented it can partially override a previous `Disallow` directive. This may be beneficial after large sections of the site have been disallowed, or if the entire site itself has been disallowed.

Here is an example that allows Googlebot into only the *google* directory:

```
User-agent: Googlebot
Disallow: /
Allow: /google/
```

`Noindex` *directive*

This directive works in the same way as the meta `robots noindex` command (which we will discuss shortly) and tells the search engines to explicitly exclude a page from the index. Since a `Disallow` directive prevents crawling but not indexing, this can be a very useful feature to ensure that the pages don't show in search results.

Sitemaps

You can use *robots.txt* to provide an autodiscovery mechanism for the spiders to find your XML Sitemap file (discussed at the beginning of this chapter). The search engines can be told where to find the file with one simple line in the *robots.txt* file:

```
Sitemap: sitemap_location
```

where *`sitemap_location`* is the complete URL to the Sitemap, such as *http://www.yourdomain.com/sitemap.xml*. You can place this anywhere in your file.

For full instructions on how to apply *robots.txt*, see Robots.txt.org. You may also find it valuable to use Dave Naylor's Robots.txt Builder tool to save time and heartache (*http://www.davidnaylor.co.uk/the-robotstxt-builder-a-new-tool.html*).

You should use great care when making changes to *robots.txt*. A simple typing error can, for example, suddenly tell the search engines to no longer crawl any part of your site. After updating your *robots.txt* file, it is always a good idea to check it with the Google Webmaster Tools Test Robots.txt tool. You can find this by logging in to Webmaster Tools and then selecting "Site configuration" followed by "Crawler access."

The rel="NoFollow" attribute

In 2005, the three major search engines (Yahoo!, Google, and Bing) all agreed to support an initiative intended to reduce the effectiveness of automated spam. Unlike the meta `robots` version of `NoFollow`, the new directive could be employed as an attribute within an `<a>` or link tag to indicate that the linking site "does not editorially vouch for the quality of the linked-to page." This enables a content creator to link to a web page without passing on any of the normal search engine benefits that typically accompany a link (things such as trust, anchor text, PageRank, etc.).

Originally, the intent was to enable blogs, forums, and other sites where user-generated links were offered to discourage spammers who built crawlers that automatically created links. However, its use has expanded, as Google in particular recommends use of `NoFollow` on links that are paid for: the search engine's preference is that only those links that are truly editorial and freely provided by publishers (i.e., without their being compensated) should count toward bolstering a site's/page's rankings.

You can implement `NoFollow` using the following format:

```
<a href="http://www.google.com" rel="NoFollow">
```

Note that although you can use `NoFollow` to restrict the passing of link value between web pages, the search engines still crawl through those links (despite the lack of semantic logic) and crawl the pages they link to. The search engines have provided contradictory input on this point. To summarize, `NoFollow` does not expressly forbid indexing or spidering, so if you link to your own pages with it, intending to keep those pages from being indexed or ranked, others may find them and link to them, and your original goal will be thwarted.

Figure 6-37 shows how a search engine robot interprets a `NoFollow` attribute when it finds one associated with a link (Link 1 in this example).

The specific link with the `NoFollow` attribute was, for a number of years, considered to be disabled from passing link juice, and the notion of sculpting PageRank using `NoFollow` was a popular one. The belief was that when you `NoFollow`ed a particular link, the link juice that would have been passed to that link was preserved, and the search engines would reallocate it to the other links found on the page. As a result, many publishers implemented `NoFollow` links to lower-value pages on their sites (such as the About Us and Contact Us pages, or alternative sort order pages for product catalogs). In fact, data from SEOmoz's Open Site Explorer tool (*http://www.opensiteexplorer.org*) published in March 2009 showed that at that time

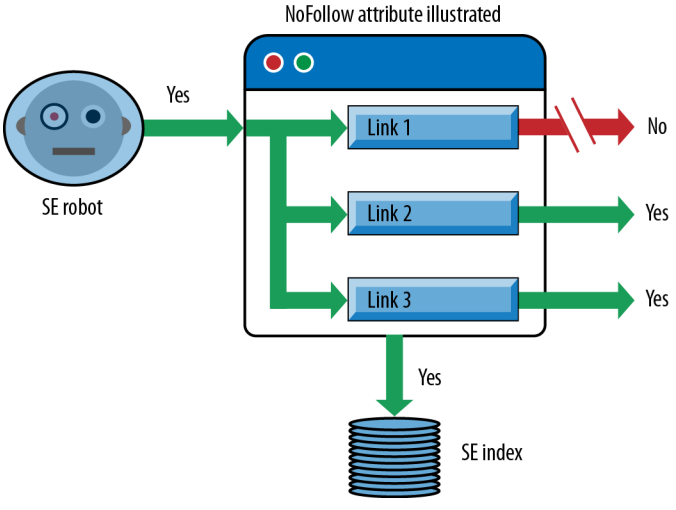

FIGURE 6-37. Impact of NoFollow attribute

about 3% of all links on the Web were `NoFollowed`, and 60% of those `NoFollows` were applied to internal links.

In June 2009, however, Google's Matt Cutts wrote a post that made it clear that the link juice associated with a `NoFollowed` link is discarded rather than reallocated (*http://www.mattcutts.com/blog/pagerank-sculpting/*). In theory, you can still use `NoFollow` however you want, but using it on internal links does not (at the time of this writing, according to Google) bring the type of benefit people have been looking for in the past.

In addition, many SEOs speculate that in some cases some value is being placed on `NoFollowed` links, and we suggest erring on the side of caution when using the `NoFollow` attribute, as its use has been attributed to "flagging" a site as overoptimized, or otherwise aggressive in its SEO tactics.

This is a great illustration of the ever-changing nature of SEO. Something that was a popular, effective tactic is now being viewed as ineffective. Some more aggressive publishers will continue to pursue link juice sculpting by using even more aggressive approaches, such as implementing links in encoded JavaScript or within iframes that have been disallowed in *robots.txt*, so that the search engines don't see them as links. Such aggressive tactics are probably not worth the trouble for most publishers.

The meta robots tag

The meta `robots` tag has three components: `cache`, `index`, and `follow`. The `cache` component instructs the engine about whether it can keep the page in the engine's public index, available via the "cached snapshot" link in the search results (see Figure 6-38).

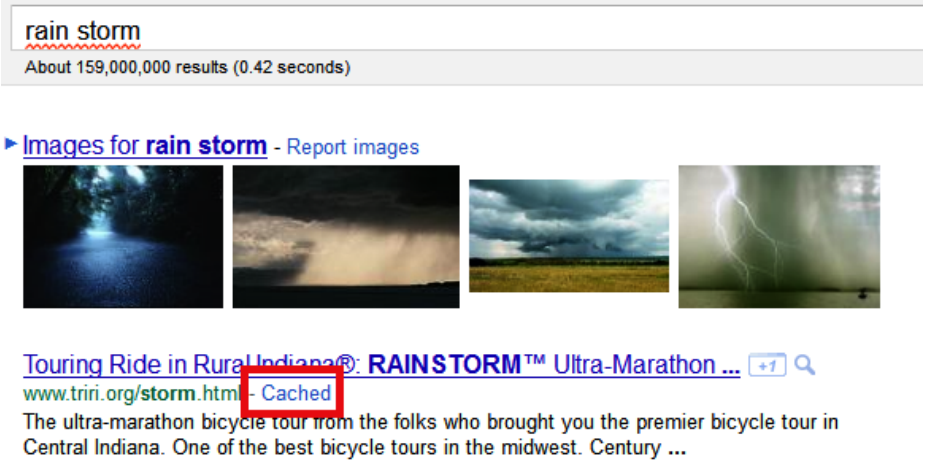

FIGURE 6-38. Snapshot of cached element in the SERPs

The `index` component tells the engine whether the page is allowed to be crawled and stored in any capacity. A page marked `NoIndex` will be excluded entirely by the search engines. By default this value is `index`, telling the search engines, "Yes, please do crawl this page and include it in your index." Thus, it is unnecessary to place this directive on each page. Figure 6-39 shows what a search engine robot does when it sees a `NoIndex` tag on a web page.

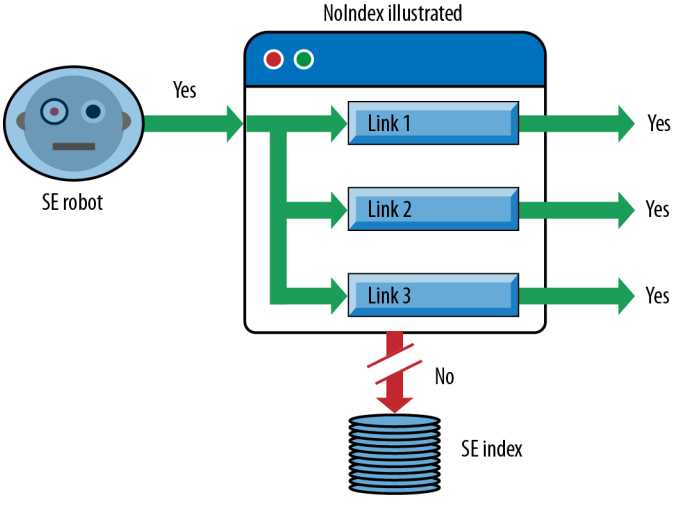

FIGURE 6-39. Impact of NoIndex

The page will still be crawled, and the page can still accumulate and pass link juice to other pages, but it will not appear in search indexes.

The final instruction available through the meta `robots` tag is `follow`. This command, like `index`, defaults to: "Yes, crawl the links on this page and pass link juice through them." Applying `NoFollow` tells the engine that the links on that page should not pass link value or be crawled. By and large, it is unwise to use this directive as a way to prevent links from being crawled. Since human beings will still reach those pages and have the ability to link to them from other sites, `NoFollow` (in the meta `robots` tag) does little to restrict crawling or spider access. Its only application is to prevent link juice from spreading out, and since the 2005 launch of the `rel="NoFollow"` attribute (discussed earlier), which allows this directive to be placed on individual links, its use has diminished.

Figure 6-40 outlines the behavior of a search engine robot when it finds a `NoFollow` meta tag on a web page.

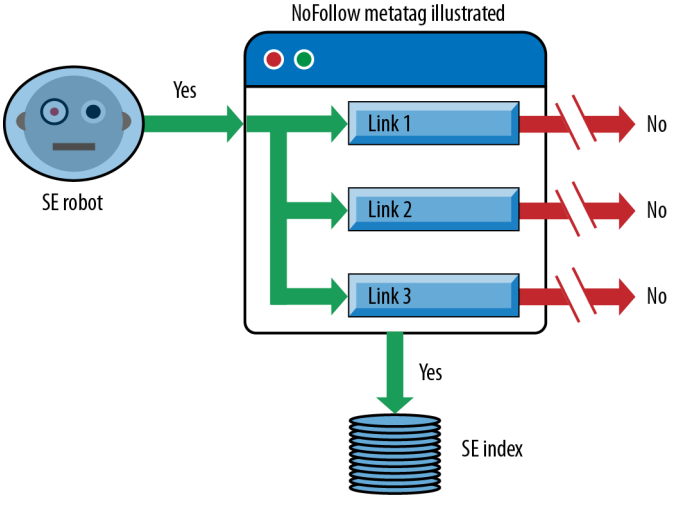

FIGURE 6-40. Impact of NoFollow meta tag

When you use the `NoFollow` meta tag on a page, the search engine will still crawl the page and place it in its index. However, all links (both internal and external) on the page will be disabled from passing link juice to other pages.

One good application for `NoIndex` is to place this tag on HTML sitemap pages. These are pages designed as navigational aids for users and search engine spiders to enable them to efficiently find the content on your site. On some sites these pages are unlikely to rank for anything of importance in the search engines, yet you still want them to pass link juice to the pages they link to. Putting `NoIndex` on these pages keeps these HTML sitemaps out of the index. Make sure

you *do not* apply the `NoFollow` meta tag on the pages or the `NoFollow` attribute on the links on the pages, as these will prevent the pages from passing link juice.

The canonical tag

In February 2009, Google, Yahoo!, and Microsoft announced a new tag known as the `canonical` tag (sometimes referred to as `rel="canonical"`). This tag was a new construct designed explicitly for the purposes of identifying and dealing with duplicate content. Implementation is very simple and looks like this:

```
<link rel="canonical" href="http://www.seomoz.org/blog" />
```

This tag is meant to tell Yahoo!, Bing, and Google that the page in question should be treated as though it were a copy of the URL *http://www.seomoz.org/blog* and that all of the link and content metrics the engines apply should technically flow back to that URL (see Figure 6-41).

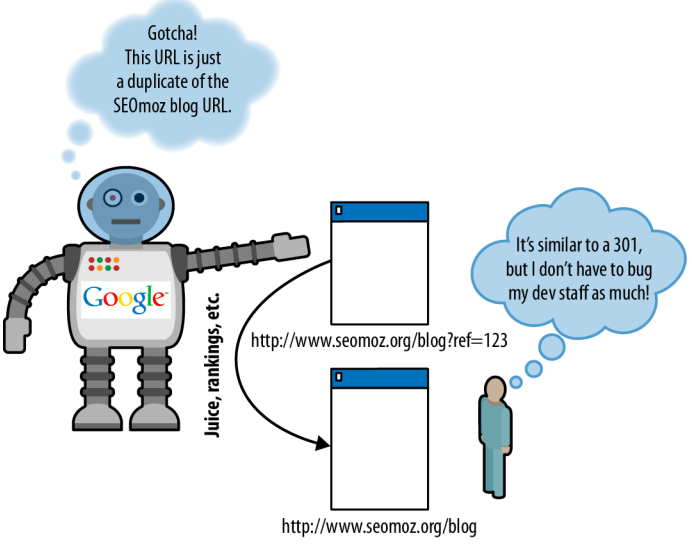

FIGURE 6-41. How search engines look at the canonical tag

The `canonical` URL tag attribute is similar in many ways to a 301 redirect, from an SEO perspective. In essence, you're telling the engines that multiple pages should be considered as one (which a 301 does), without actually redirecting visitors to the new URL (for many publishers this requires less effort than some of the other solutions for their development staff). There are some differences, though:

- Whereas a 301 redirect sends all traffic (bots and human visitors) to the designated location, the `canonical` URL tag is just for engines, meaning you can still separately track visitors to the unique URL versions.

- A 301 is a much stronger signal that multiple pages have a single, canonical source. 301s are considered directives that search engines and browsers are obligated to honor, but the `canonical` tag is treated as a suggestion. Although the engines support this new tag and trust the intent of site owners, there will be limitations. Content analysis and other algorithmic metrics will be applied to ensure that a site owner hasn't mistakenly or manipulatively applied the tag, and you can certainly expect to see mistaken uses of the `canonical` tag, resulting in the engines maintaining those separate URLs in their indexes (meaning site owners would experience the same problems noted earlier in this chapter, in "Duplicate Content Issues" on page 234).

We will discuss some applications for this tag later in this chapter. In general practice, the best solution is to resolve the duplicate content problems at their core, and eliminate them if you can. This is because the `canonical` tag is not guaranteed to work. However, it is not always possible to resolve the issues by other means, and the `canonical` tag provides a very effective backup plan.

You can also include the `canonical` tag directly within the HTTP response header for your page. The code might look something like the following:

```
HTTP/1.1 200 OK
Content-Type: application/pdf
Link: <http://www.example.com/white-paper.html>; rel="canonical"
Content-Length: 785710
(... rest of HTTP response headers...)
```

You can read more about this at *http://googlewebmastercentral.blogspot.com/2011/06/supporting-relcanonical-http-headers.html*.

Blocking and cloaking by IP address range

You can block particular bots from crawling entire IP addresses or ranges through server-side restrictions on IPs. Most of the major engines crawl from a limited number of IP ranges, making it possible to identify them and restrict access. This technique is, ironically, popular with webmasters who mistakenly assume that search engine spiders are spammers attempting to steal their content, and thus block the IP ranges to restrict access and save bandwidth. Use caution when blocking bots, and make sure you're not restricting access to a spider that could bring benefits, either from search traffic or from link attribution.

Blocking and cloaking by user agent

At the server level, it is possible to detect user agents and restrict their access to pages or websites based on their declaration of identity. As an example, if a website detected a rogue bot, you might double-check its identity before allowing access. The search engines all use a similar protocol to verify their user agents via the Web: a reverse DNS lookup followed by a corresponding forward DNS IP lookup. An example for Google would look like this:

```
> host 66.249.66.1
1.66.249.66.in-addr.arpa domain name pointer crawl-66-249-66-1.googlebot.com.

> host crawl-66-249-66-1.googlebot.com
crawl-66-249-66-1.googlebot.com has address 66.249.66.1
```

A reverse DNS lookup by itself may be insufficient, because a spoofer could set up reverse DNS to point to *xyz.googlebot.com* or any other address.

Using iframes

Sometimes there's a certain piece of content on a web page (or a persistent piece of content throughout a site) that you'd prefer search engines didn't see. As we discussed earlier in this chapter, clever use of iframes can come in handy here, as Figure 6-42 illustrates.

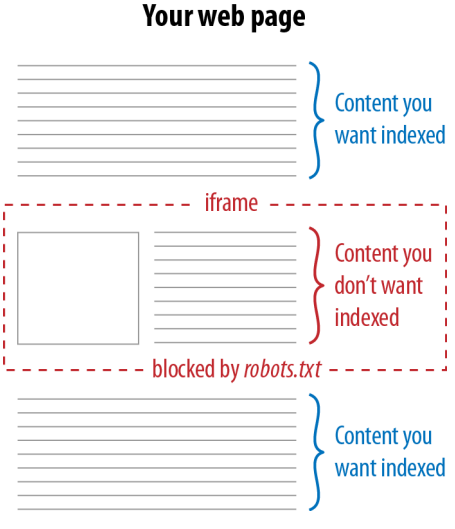

FIGURE 6-42. Using iframes to prevent indexing of content

The concept is simple: by using iframes, you can embed content from another URL onto any page of your choosing. By then blocking spider access to the iframe with *robots.txt*, you ensure that the search engines won't "see" this content on your page. Websites may do this for many reasons, including to avoid duplicate content problems, reduce the page size for search engines, or lower the number of crawlable links on a page (to help control the flow of link juice).

Hiding text in images

As we discussed previously, the major search engines still have very limited capacity to read text in images (and the processing power required makes for a severe barrier). Hiding content inside images isn't generally advisable, though, as it can be impractical for alternative devices (mobile devices, in particular) and inaccessible to others (such as screen readers).

Hiding text in Java applets

As with text in images, the search engines cannot easily parse content inside Java applets. Using them as a tool to hide text would certainly be a strange choice, though.

Forcing form submission

Search engines will not submit HTML forms in an attempt to access the information retrieved from a search or submission. Thus, if you keep content behind a forced-form submission and never link to it externally, your content will remain out of the engines' indexes (as Figure 6-43 demonstrates).

FIGURE 6-43. Content that can only be accessed by submitting a form is unreadable by crawlers

The problem comes when content behind forms earns links outside your control, as when bloggers, journalists, or researchers decide to link to the pages in your archives without your knowledge. Thus, although form submission may keep the engines at bay, you should make sure that anything truly sensitive has additional protection (e.g., through *robots.txt* or meta `robots`).

Using login/password protection

Password protection of any kind will effectively prevent any search engines from accessing content, as will any form of human-verification requirements, such as CAPTCHAs (the boxes that request the copying of letter/number combinations to gain access). The major engines won't try to guess passwords or bypass these systems.

Removing URLs from a search engine's index

A secondary, post-indexing tactic, URL removal is possible at most of the major search engines through verification of your site and the use of the engines' tools. For example, Google allows you to remove URLs through Webmaster Central (*https://www.google.com/webmasters/tools/removals*). Microsoft's Bing search engine may soon carry support for this as well.

Redirects

A *redirect* is used to indicate when content has moved from one location to another. For example, suppose you have some content at *http://www.yourdomain.com/old.html*, and you decide to restructure your site. As a result of this restructuring, your content may move to *http://www.yourdomain.com/critical-keyword.html*.

Once a redirect is implemented, users who go to the old versions of your pages (perhaps via bookmarks they have kept for the pages) will be sent to the new versions of those pages. Without the redirect, the user would get a Page Not Found (404) error. With the redirect, the web server tells the incoming user agent (whether a browser or a spider) to instead fetch the requested content from the new URL.

Why and When to Redirect

Redirects are important not only for users, but also for letting search engines know when you have moved content. After you move a page, the search engines will still have the old URL in their indexes and will continue to return it in their search results until they discover the page is no longer there and discover the new page. You can help speed up this process by implementing a redirect. Here are some scenarios in which you may end up needing to implement redirects:

- You have old content that expires, so you remove it.
- You find that you have broken URLs that have links and traffic.
- You change your hosting company.
- You change your CMS.
- You want to implement a canonical redirect (redirect all pages on *http://yourdomain.com* to *http://www.yourdomain.com*).
- You change the URLs where your existing content can be found for any reason.

Not all of these scenarios require a redirect. For example, you can change hosting companies without impacting any of the URLs used to find content on your site, in which case no redirect is required. However, any scenario in which any of your URLs change is a scenario in which you need to implement redirects.

Good and Bad Redirects

It turns out that there is more than one way to perform a redirect, and they are not all created equal. There are two major types of redirects that can be implemented, tied specifically to the HTTP status code returned by the web server to the browser. These are:

"301 moved permanently"
: This status code tells the browser (or search engine crawler) that the resource has been permanently moved to another location, and there is no intent to ever bring it back.

"302 moved temporarily"
: This status code tells the browser (or search engine crawler) that the resource has been temporarily moved to another location, and that the move should not be treated as permanent.

Both forms of redirect send a human or a search engine crawler to the new location, but the search engines interpret these two HTTP status codes in very different ways. When a crawler sees a 301 HTTP status code, it assumes it should pass the historical link juice (and any other metrics) from the old page to the new one. When a search engine crawler sees a 302 HTTP status code, it assumes it should not pass the historical link juice from the old page to the new one. In addition, the 301 redirect will lead the search engine to remove the old page from its index and replace it with the new one.

The preservation of historical link juice is critical in the world of SEO. For example, imagine you had 1,000 links to *http://www.yourolddomain.com* and you decided to relocate everything to *http://www.yournewdomain.com*. If you used redirects that returned a 302 status code, you would be starting your link-building efforts from scratch again. In addition, the old version of the page might remain in the search engines' indexes and compete with the new version for search rankings.

It should also be noted that there can be redirects that pass no status code, or the wrong status code, such as a 404 error (Page Not Found) or a 200 OK (page loaded successfully). These are also problematic, and should be avoided. You want to be sure to return a 301 HTTP status code when you have performed a redirect whenever you make a permanent change to a page's location.

Methods for URL Redirecting and Rewriting

There are many possible ways to implement redirects. On Apache web servers (normally present on machines running Unix or Linux as the operating system), it is possible to implement redirects quite simply in a standard file called *.htaccess*, using the `Redirect` and `RedirectMatch` directives (you can learn more about this file format at *http://httpd.apache.org/docs/2.2/howto/htaccess.html*). More advanced directives known as *rewrite rules* can be employed as well, using the Apache module known as mod_rewrite, which we will discuss in a moment.

On web servers running Microsoft IIS (*http://www.iis.net*), different methods are provided for implementing redirects. The basic method for doing redirects is through the IIS console (you can read more about this at *http://www.mcanerin.com/EN/articles/301-redirect-IIS.asp*). People with IIS servers can also make use of a text file with directives, provided they use an ISAPI plug-in such as ISAPI_Rewrite (*http://www.isapirewrite.com*). This scripting language offers similar capabilities to Apache's mod_rewrite module.

Many programmers use other techniques for implementing redirects. This can be done directly in programming languages such as Perl, PHP, ASP, and JavaScript. When implementing redirects in this fashion, the key thing that the programmer must do is to make sure the HTTP status code returned by the web server is a 301. You can check the header returned with the Firefox plug-in Live HTTP Headers (*http://livehttpheaders.mozdev.org*).

Another method that you can use to implement a redirect occurs at the page level, via the meta `refresh` tag, which looks something like this:

```
<meta http-equiv="refresh"
      content="5;url=http://www.yourdomain.com/newlocation.htm" />
```

The first parameter in the `content` section in the preceding statement, the number 5, indicates the number of seconds the web server should wait before redirecting the user to the indicated page. This gets used in scenarios where the publisher wants to display a page letting the user know that he is going to get redirected to a different page than the one he requested.

The problem is that most meta refreshes are treated like 302 redirects. The sole exception to this is if you specify a redirect delay of 0 seconds. You will have to give up your helpful page telling the user that you are moving him, but the search engines appear to treat this as though it were a 301 redirect (to be safe, the best practice is simply to use a 301 redirect if at all possible).

mod_rewrite and ISAPI_Rewrite for URL rewriting and redirecting

There is much more to this topic than we can reasonably address in this book. The following description is intended only as an introduction to help orient more technical readers, including web developers and site webmasters, as to how rewrites and redirects function. If you'd prefer to skip this technical discussion, proceed to "Redirecting a Home Page Index File Without Looping" on page 269.

mod_rewrite for Apache and ISAPI_Rewrite for Microsoft IIS Server offer very powerful ways to rewrite your URLs. Here are some reasons why you might want to use these powerful tools:

- You have changed the URL structure on your site so that content has moved from one location to another. This can happen when you change your CMS, or when you change your site organization for any reason.
- You want to map your search engine–unfriendly URLs into friendlier ones.

If you are running Apache as your web server, you would place directives known as *rewrite rules* within your *.htaccess* file or your Apache configuration file (e.g., *httpd.conf* or the

site-specific config file in the *sites_conf* directory). Similarly, if you are running IIS Server, you'd use an ISAPI plug-in such as ISAPI_Rewrite and place rules in an *httpd.ini* config file.

The following discussion focuses on mod_rewrite; note that the rules can differ slightly in ISAPI_Rewrite. Your *.htaccess* file would start with:

```
RewriteEngine on
RewriteBase /
```

You should omit the second line if you're adding the rewrites to your server config file, since `RewriteBase` is supported only in *.htaccess*. We're using `RewriteBase` here so that you won't have to have `^/` at the beginning of all the rules, just `^` (we will discuss regular expressions in a moment).

After this step, the rewrite rules are implemented. Perhaps you want to have requests for product page URLs of the format *http://www.yourdomain.com/products/123* display the content found at *http://www.yourdomain.com/get_product.php?id=123*, without the URL changing in the location bar of the user's browser and without you having to recode the *get_product.php* script. Of course, this doesn't replace all occurrences of dynamic URLs within the links contained on all the site pages; that's a separate issue. You can accomplish the first part with a single rewrite rule, like so:

```
RewriteRule ^products/([0-9]+)/?$ /get_product.php?id=$1 [L]
```

This tells the web server that all requests that come into the */product/* directory should be mapped into requests to */get_product.php*, while using the subfolder to */product/* as a parameter for the PHP script.

The `^` signifies the start of the URL following the domain, `$` signifies the end of the URL, `[0-9]` signifies a numerical digit, and the `+` immediately following it means one or more occurrences of a digit. Similarly, the `?` immediately following the `/` means zero or one occurrence of a slash character. The `()` puts whatever is wrapped within it into memory. You can then access what's been stored in memory with `$1` (i.e., what is in the first set of parentheses). Not surprisingly, if you included a second set of parentheses in the rule, you'd access that with `$2`, and so on. The `[L]` flag saves on server processing by telling the rewrite engine to stop if it matches on that rule. Otherwise, all the remaining rules will be run as well.

Here's a slightly more complex example, which indicates that URLs of the format *http://www.yourdomain.com/webapp/wcs/stores/servlet/ProductDisplay?storeId=10001&catalogId=10001&langId=-1&categoryID=4&productID=123* should be rewritten to *http://www.yourdomain.com/4/123.htm*:

```
RewriteRule ^([^/]+)/([^/]+)\.htm$
/webapp/wcs/stores/servlet/ProductDisplay?storeId=10001&catalogId=10001&
langId=-1&categoryID=$1&productID=$2 [QSA,L]
```

The `[^/]` signifies any character other than a slash. That's because, within square brackets, `^` is interpreted as *not*. The `[QSA]` flag is for when you don't want the query string dropped (like when you want a tracking parameter preserved).

To write good rewrite rules, you will need to become a master of *pattern matching* (which is simply another way to describe the use of regular expressions). Here are some of the most important special characters and how the rewrite engine interprets them:

- * means 0 or more occurrences of the immediately preceding character.
- + means 1 or more occurrences of the immediately preceding character.
- ? means 0 or 1 occurrence of the immediately preceding character.
- ^ means the beginning of the string.
- $ means the end of the string.
- . means any character (i.e., it acts as a wildcard).
- \ "escapes" the character that follows; for example, \. means the dot is not meant to be a wildcard, but an actual character.
- ^ inside square brackets ([]) means *not*; for example, [^/] means *not slash*.

It is incredibly easy to make errors in regular expressions. Some of the common gotchas that lead to unintentional substring matches include:

- Using .* when you should be using .+, since .* can match on nothing.
- Not "escaping" with a backslash a special character that you don't want interpreted, as when you specify . instead of \. and you really meant the dot character rather than any character (thus, *default.htm* would match on *defaulthtm*, and *default\.htm* would match only on *default.htm*).
- Omitting ^ or $ on the assumption that the start or end is implied (thus, *default\.htm* would match on *mydefault.html*, whereas *^default\.htm$* would match only on *default.htm*).
- Using "greedy" expressions that will match on all occurrences rather than stopping at the first occurrence.

The easiest way to illustrate what we mean by greedy is to provide an example:

```
RewriteRule ^(.*)/?index\.html$ /$1/ [L,R=301]
```

This will redirect requests for *http://www.yourdomain.com/blah/index.html* to *http://www.yourdomain.com/blah//*. This is probably not what was intended. Why did this happen? Because .* will capture the slash character within it before the /? gets to see it. Thankfully, there's an easy fix: simply use [^] or .*? instead of .* to do your matching. For example, use ^(.*?)/? instead of ^(.*)/? or [^/]+/[^/] instead of .*/.*.

So, to correct the preceding rule, you could use the following:

```
RewriteRule ^(.*?)/?index\.html$ /$1/ [L,R=301]
```

When wouldn't you use the following?

```
RewriteRule ^([^/]*)/?index\.html$ /$1/ [L,R=301]
```

This is more limited because it will match only on URLs with one directory. URLs containing more than one subdirectory, such as *http://www.yourdomain.com/store/cheese/swiss/wheel/index.html*, would not match.

> NOTE
> **The `[R=301]` flag in the last few examples—as you might guess—tells the rewrite engine to do a 301 redirect instead of a standard rewrite.**

As you might imagine, testing/debugging is a big part of URL rewriting. When you're debugging, the `RewriteLog` and `RewriteLogLevel` directives are your friends! Set the `RewriteLogLevel` to `4` or more to start seeing what the rewrite engine is up to when it interprets your rules.

Another handy directive to use in conjunction with `RewriteRule` is `RewriteCond`. You would use `RewriteCond` if you were trying to match on something in the query string, the domain name, or anything else not present between the domain name and the question mark in the URL (which is what `RewriteRule` looks at).

Note that neither `RewriteRule` nor `RewriteCond` can access what is in the anchor part of a URL—that is, whatever follows a `#`—because that is used internally by the browser and is not sent to the server as part of the request. The following `RewriteCond` example looks for a positive match on the hostname before it allows the rewrite rule that follows to be executed:

```
RewriteCond %{HTTP_HOST} !^www\.yourdomain\.com$ [NC]
RewriteRule ^(.*)$ http://www.yourdomain.com/$1 [L,R=301]
```

Note the exclamation point (`!`) at the beginning of the regular expression. The rewrite engine interprets that as *not*.

For any hostname other than *http://www.yourdomain.com*, a 301 redirect is issued to the equivalent canonical URL on the *www* subdomain. The `[NC]` flag makes the rewrite condition case-insensitive. Where is the `[QSA]` flag so that the query string is preserved, you might ask? It is not needed when redirecting; it is implied.

If you don't want a query string retained on a rewrite rule with a redirect, put a question mark at the end of the destination URL in the rule, like so:

```
RewriteCond %{HTTP_HOST} !^www\.yourdomain\.com$ [NC]
RewriteRule ^(.*)$ http://www.yourdomain.com/$1? [L,R=301]
```

Why not use `^yourdomain\.com$` instead? Consider:

```
RewriteCond %{HTTP_HOST} ^yourdomain\.com$ [NC]
RewriteRule ^(.*)$ http://www.yourdomain.com/$1? [L,R=301]
```

That would not have matched on typo domains, such as "yourdoamin.com", that the DNS server and virtual host would be set to respond to (assuming that misspelling was a domain you registered and owned).

Under what circumstances might you want to omit the query string from the redirected URL, as we did in the preceding two examples? When a session ID or a tracking parameter (such as `source=banner_ad1`) needs to be dropped. Retaining a tracking parameter after the redirect is not only unnecessary (because the original URL with the source code appended would have been recorded in your access logfiles as it was being accessed); it is also undesirable from a canonicalization standpoint. What if you wanted to drop the tracking parameter from the redirected URL, but retain the other parameters in the query string? Here's how you'd do it for static URLs:

```
RewriteCond %{QUERY_STRING} ^source=[a-z0-9]*$
RewriteRule ^(.*)$ /$1? [L,R=301]
```

And for dynamic URLs:

```
RewriteCond %{QUERY_STRING} ^(.+)&source=[a-z0-9]+(&?.*)$
RewriteRule ^(.*)$ /$1?%1%2 [L,R=301]
```

Need to do some fancy stuff with cookies before redirecting the user? Invoke a script that cookies the user and then 301s her to the canonical URL:

```
RewriteCond %{QUERY_STRING} ^source=([a-z0-9]*)$
RewriteRule ^(.*)$ /cookiefirst.php?source=%1&dest=$1 [L]
```

Note the lack of an `[R=301]` flag in the preceding code. That's on purpose. There's no need to expose this script to the user. Use a rewrite and let the script itself send the 301 after it has done its work.

Other canonicalization issues worth correcting with rewrite rules and the `[R=301]` flag include when the engines index online catalog pages under HTTPS URLs, and URLs missing a trailing slash that should be there. First, the HTTPS fix:

```
# redirect online catalog pages in the /catalog/ directory if HTTPS
RewriteCond %{HTTPS} on
RewriteRule ^catalog/(.*) http://www.yourdomain.com/catalog/$1 [L,R=301]
```

Note that if your secure server is separate from your main server, you can skip the `RewriteCond` line.

Now to append the trailing slash:

```
RewriteRule ^(.*[^/])$ /$1/ [L,R=301]
```

After completing a URL rewriting project to migrate from dynamic to static URLs, you'll want to phase out the dynamic URLs not just by replacing all occurrences of the legacy URLs on your site, but also by 301-redirecting the legacy dynamic URLs to their static equivalents. That way, any inbound links pointing to the retired URLs will end up leading both spiders and humans to the correct new URLs—thus ensuring that the new URLs are the ones that are indexed, blogged about, linked to, and bookmarked, and that the old URLs are removed from the search engine's indexes. Generally, here's how you'd accomplish that:

```
RewriteCond %{QUERY_STRING} id=([0-9]+)
RewriteRule ^get_product\.php$ /products/%1.html? [L,R=301]
```

However, you'll get an infinite loop of recursive redirects if you're not careful. One quick and dirty way to avoid that situation is to add a nonsense parameter to the destination URL for the rewrite and ensure that this nonsense parameter isn't present before doing the redirect. Specifically:

```
RewriteCond %{QUERY_STRING} id=([0-9]+)
RewriteCond %{QUERY_STRING} !blah=blah
RewriteRule ^get_product\.php$ /products/%1.html? [L,R=301]
RewriteRule ^products/([0-9]+)/?$ /get_product.php?id=$1&blah=blah [L]
```

Notice that the example used two `RewriteCond` lines, stacked on top of each other. All redirect conditions listed together in the same block will be "ANDed" together. If you wanted the conditions to be "ORed," you will have to use the `[OR]` flag.

Redirecting a Home Page Index File Without Looping

Many websites link to their own home page in a form similar to *http://www.yourdomain.com/index.html*. The problem with that is that most incoming links to the site's home page specify *http://www.yourdomain.com*, thus dividing the link juice into the site. Once publishers realize this, they will typically want to fix their internal links and then 301-redirect *http://www.yourdomain.com/index.html* to *http://www.yourdomain.com/*. However, problems with recursive redirects can develop if this is not done correctly.

When someone comes to your website by typing in *http://www.yourdomain.com*, the DNS system of the Internet helps the browser locate the web server for your website. How, then, does the web server decide what to send to the browser? It turns out that it does this by loading a file from the hard drive of the web server for your website.

When no file is specified (i.e., if, as in the preceding example, only the domain name is specified), the web server loads a file that is known as the *default file*. This is often a file with a name such as *index.html*, *index.htm*, *index.shtml*, *index.php*, or *default.asp*.

The filename can actually be anything, but most web servers default to one specific filename or another. The problem is that many CMSs will expose both forms of your home page—that is, both *http://www.yourdomain.com* and *http://www.yourdomain.com/index.php*.

All the pages on your site may link only to *http://www.yourdomain.com/index.php*, but given human nature, most of the links to your home page that third parties give you will probably point to *http://www.yourdomain.com/*. This can create a duplicate content problem if the search engine sees two versions of your home page and thinks they are separate, but duplicate, documents. Google is pretty smart at figuring out this particular issue, but it is best to not rely on that.

Since you've learned how to do 301 redirects, you might conclude that the solution is to 301-redirect *http://www.yourdomain.com/index.php* to *http://www.yourdomain.com/*. Sounds good, right? Unfortunately, there is a big problem with this.

What happens is the server sees the request for *http://www.yourdomain.com/index.php* and then sees that it is supposed to 301-redirect that to *http://www.yourdomain.com/*, which it dutifully does. But when it loads *http://www.yourdomain.com/*, it retrieves the default filename (*index.php*) and proceeds to load *http://www.yourdomain.com/index.php*. Then it sees that you want to redirect that to *http://www.yourdomain.com/*, and it creates an infinite loop.

The default document redirect solution

The solution that follows is specific to the preceding *index.php* example. You will need to plug in the appropriate default filename for your own web server:

1. Copy the contents of *index.php* to another file. For this example, we'll be using *sitehome.php*.
2. Create an Apache `DirectoryIndex` directive for your document root. Set it to *sitehome.php*. Do not set the directive on a server-wide level; otherwise, it may cause problems with other folders that still need to use *index.php* as a directory index.
3. Put this in an *.htaccess* file in your document root: *DirectoryIndexsitehome.php*. Or, if you aren't using per-directory context files, put this in your *httpd.conf*:

```
<Directory /your/document/root/examplesite.com/>
 DirectoryIndex sitehome.php
</Directory>
```

4. Clear out the contents of your original *index.php* file and insert this line of code:

```
<? header("Location: http://www.example.com"); ?>
```

This sets it up so that *index.php* is not a directory index file (i.e., the default filename). It forces *sitehome.php* to be read when someone types in the canonical URL (*http://www.yourdomain.com*). Any requests to *index.php* from old links can now be 301-redirected while avoiding an infinite loop.

If you are using a CMS, you also need to make sure when you are done with this that all the internal links now go to the canonical URL, *http://www.yourdomain.com*. If for any reason the CMS started to point to *http://www.yourdomain.com/sitehome.php*, the loop problem would return, forcing you to go through this entire process again.

Content Management System (CMS) Issues

When looking to publish a new site, many publishers may wonder whether they need to use a CMS, and, if so, how to ensure that it is SEO-friendly.

It is essential to determine whether a CMS is necessary before embarking on a web development project. You can use the flowchart in Figure 6-44 to help guide you through the process.

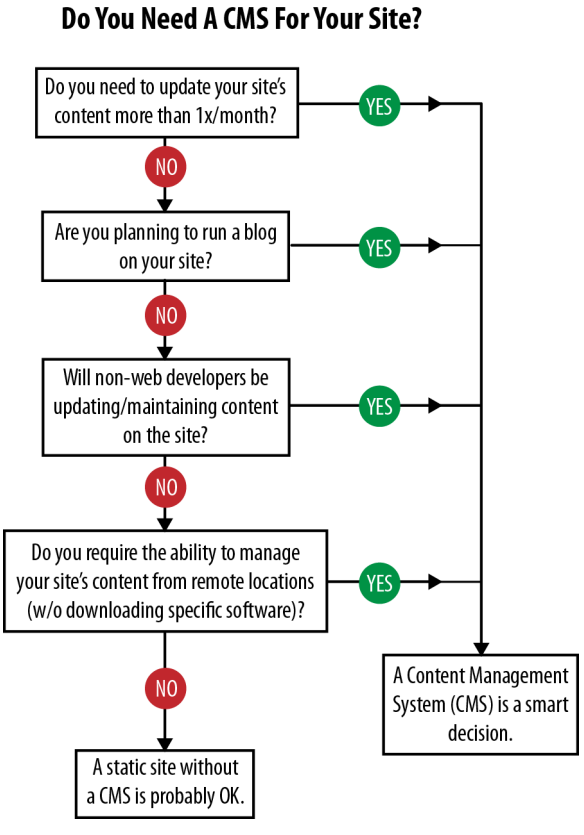

FIGURE 6-44. Flowchart to determine whether you need a CMS

Due to the inexpensiveness of customizable, free platforms such as Drupal, Joomla, WordPress, and Mambo, it is increasingly rare for a publisher to develop a static site, even when a CMS isn't required.

The next step involves understanding how to ensure that a CMS will be search engine–friendly. Here is a list of basic SEO issues that you should be aware of when dealing with CMSs (both prebuilt and custom-made). By dealing with these, you will ensure a relatively smooth platform for content delivery:

Title tag customization and rules

A search engine–friendly CMS must not only allow for title tags to be customized on a page-specific level, but also enable rules for particular sections of a website. For example, if your CMS requires that the title tag always has to start with your site name followed by a colon followed by your article title, you're sunk—at least as far as your SEO is concerned. You should be able to revise the formulas used to generate the title tags across your site to make them more search-optimal.

Static, keyword-rich URLs

URLs have historically been the most problematic SEO issue for CMS platforms. Nowadays, search-friendly CMSs should feature custom URL creation. In WordPress, a custom URL is referred to as a "post slug."

Figure 6-45 is an example from SEOmoz's custom-built CMS. Notice how the first line allows you to create the title of the post, and the second enables manual sculpting of the URL structure (and an automatic Generate button if you prefer to simply use the post title).

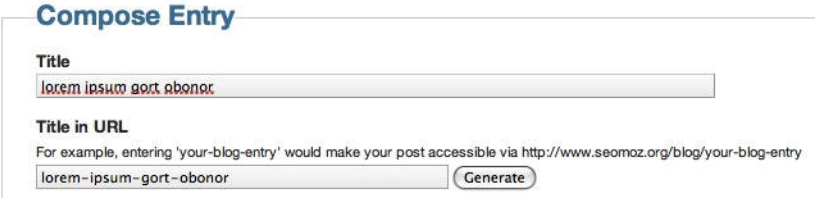

FIGURE 6-45. Example of custom URL creation

Meta tag customization

Being able to implement custom meta descriptions and meta `robots` tags is critical. Enabling editorial control is essential for a good CMS.

Enabling custom HTML tags

A good CMS has to offer extra functionality on HTML tags for things such as `NoFollow` on links, or `<Hx>` tags for headings and subheadings. These can be built-in features accessible through menu options, or the CMS can simply allow manual editing of HTML in the text editor window when required. Having no `<h1>` tags on a given page is not desirable, but neither is having too many `<h1>` tags. The best content to have wrapped in an `<h1>` is the article or page title; having low-value content (such as the publication date) marked up as an `<h1>` is not desirable.

Internal anchor text flexibility

For a site to be "optimized" rather than simply search-friendly, customizing the anchor text on internal links is critical. Rather than simply using the page titles for all links in a site's architecture, a great CMS should be flexible enough to handle custom input from the administrators as to the anchor text of category-level or global navigation links.

Intelligent categorization structure

Another problem is poor category structure. When designing an information architecture for a website, you should not place limits on how pages are accessible due to the CMS's inflexibility. CMSs that offer customizable navigation panels will be the most successful in this respect.

Pagination controls

Pagination can be the bane of a website's search rankings, so controlling it through inclusion of more items per page, more contextually relevant anchor text (e.g., not "next,"

"prev," and page numbers), and careful use of meta `NoIndex` tags will make your important content get more link juice and crawl attention.

301-redirect functionality

Many CMSs sadly lack this critical feature, disallowing the proper redirection of content when necessary; 301s are valuable for expired content, for pages that have a newer version, and for dodging keyword cannibalization issues similar to those we discussed earlier in this chapter.

XML/RSS pinging

Although it is primarily useful for blogs, any content, from articles to product info to press releases, can be issued in a feed, and by utilizing quick, accurate pinging of the major feed services, you limit some of your exposure to duplicate content spammers who pick up your feeds and ping the major services quickly in the hopes of beating you to the punch.

Image-handling and `alt` *attributes*

`alt` attributes are a clear must-have from an SEO perspective, serving as the "anchor text" when an image is used as a link (note that text links are much better than images with `alt` attributes for links, but if you must use image links you do want to have the `alt` attribute implemented) and providing relevant, indexable content for the search engines.

Images in a CMS's navigational elements should preferably use CSS image replacement rather than mere `alt` attributes.

CSS exceptions

The application of CSS styles in a proper CMS should allow for manual exceptions so that a user can modify how a strong headline or list element appears visually. If the CMS does not offer this, writers may opt out of using proper semantic markup for presentation purposes, which would not be a good thing.

Static caching options

Many CMSs currently offer caching options, which are a particular boon if a page is receiving a high level of traffic from social media portals or news sites. A bulky CMS often makes dozens of extraneous database connections, which can overwhelm a server if caching is not in place, killing potential inbound links and media attention.

URLs free of tracking parameters and session IDs

Sticking session or tracking information such as the user's click path into the URL is deadly for SEO. It usually leads to incomplete indexation and duplicate content issues.

Customizable URL structure

If the default URL structure of the CMS doesn't suit your needs, you should be able to change it. For example, if you don't want */archives/* in the URLs of all your archived articles, you should be able to remove it; if you want to reference the article name instead of the article's database ID in the URL, you should be able to do that too.

301 redirects to a canonical URL

Duplicate content is the bane of many a dynamic website owner. Automatic handling of this by the CMS through the use of 301 redirects is a must.

Static-looking URLs

The most palatable URLs to spiders are the ones that look like they lead to static pages—so make sure that your CMS will place no query strings in the URL.

Keywords in URLs

Keywords in your URLs can help your rankings. Check that your CMS allows you to do this, as some CMS platforms don't allow you to do this.

RSS feeds

The CMS should auto-create RSS feeds to help your site rank in Google Blog Search and other feed engines.

Multilevel categorization structure

It is awfully limiting to your site structure and internal hierarchical linking structure to have a CMS that doesn't allow you to nest subcategories into categories, sub-subcategories into subcategories, and so on.

Paraphrasable excerpts

Duplicate content issues are exacerbated on dynamic sites such as blogs when the same content is displayed on permalink pages, category pages, archives-by-date pages, tag pages, and the home page. If your CMS offers the capability, crafting unique content for the excerpt and having that content display on all locations except for the permalink page will help strengthen your permalink page as unique content.

Breadcrumb navigation

Verify that your CMS allows you to implement breadcrumb (drill-down) navigation. This is great for SEO because it reinforces your internal hierarchical linking structure with keyword-rich text links.

Meta `NoIndex` *tags for low-value pages*

Even if you use `NoFollow` attributes in links to these pages, other people may still link to them, so there is a risk of those pages ranking above some of your more valuable content. Check if your CMS allows you to `NoIndex` those pages instead, as that is a better way to handle low-value pages.

Keyword-rich intro copy on category-level pages

Some CMS systems do not allow you to write custom keyword-rich introductory text for your category pages. This is unfortunate, as this type of content helps set a stable keyword theme for the page, rather than relying on the latest article or blog post to be the most prominent text on the page.

`NoFollow` *links in comments*

If you allow visitors to post comments and do not `NoFollow` the links, your site will be a spam magnet. Heck, you'll probably be a spam magnet anyway, but you won't risk losing PageRank to spammers if you use `NoFollow` attributes.

Customizable anchor text on navigational links

"Contact," "About Us," "Read More," "Full Article," and so on make for lousy anchor text—at least from an SEO standpoint. Hopefully, your CMS allows you to improve such links to make the anchor text more keyword-rich.

XML Sitemap generator

Having your CMS generate your XML Sitemap can save a lot of hassle, as opposed to trying to generate one with a third-party tool.

HTML4, HTML5, or XHTML validation

Although HTML validation is not a ranking signal, it is desirable to have the CMS automatically check for malformed HTML, as search engines may end up seeing a page differently from how it renders on the screen and accidentally consider navigation to be part of the content, or vice versa.

Pingbacks, trackbacks, comments, and antispam mechanisms

The problem with comments/trackbacks/pingbacks is that they are vectors for spam, so if you have one or more of these features enabled, you will be spammed. Therefore, effective spam prevention in the form of Akismet, Mollom, or Defensio is a must.

If you want more information on picking a quality CMS, there are some great web resources out there (among them *http://php.opensourcecms.com* and *http://www.cmsmatrix.org*) to help you manage this task.

Selecting a CMS

There are many factors to consider when choosing an existing CMS. Many CMSs are free, but some of them are proprietary, with a license cost per site. The majority of CMSs were not designed with security, stability, search friendliness, and scalability in mind, though in recent years a few vendors have developed excellent CMSs that have search friendliness as their primary focus. Many were developed to fit a certain market niche, but can be expanded to fit other purposes. Some are no longer maintained. Many are supported and developed primarily by hobbyists who don't particularly care if you're having trouble getting them installed and configured. Some are even intentionally made to be difficult to install and configure so that you'll be encouraged to pay the developers a consulting fee to do it all for you.

Popular CMS solutions that the authors have experience with include Joomla, Drupal, Pixelsilk, and WordPress. Each of these has strong support for SEO, but each of them requires some configuration for optimal results. Make sure you get that help up front to get the SEO for your site off to a strong start.

Selecting a CMS is an important process. If you make the wrong choice, you will doom your site to failure. Like most software, CMSs are a moving target—what's missing today may be a new feature tomorrow. In addition, just because a feature exists doesn't mean it is the default option, so in many instances the desired functionality will need to be enabled and possibly customized to work to your specifications.

Third-Party CMS Add-ons

Many CMS platforms offer third-party plug-ins or add-ons that extend the core functionality of the CMS. In the WordPress plug-in directory alone there are over 15,000 plug-ins. Plug-ins provide a simple way to add new SEO features and functionality, making the CMS much more flexible and future-proof. It is particularly helpful when there is an active community developing plug-ins. An active community also comes in very handy in providing free technical support when things go wrong; when bugs and security vulnerabilities crop up, it is important to have an active developer base to solve those issues quickly.

Many CMS add-ons—for example, discussion forums, customer reviews, and user polls—may come in the form of either independent software installed on your web server, or hosted services. Discussion forums, for example, come in two forms: bbPress is installed software and is optimized for search; vbulletin is a hosted solution and therefore is more difficult to optimize for search.

The problem with hosted solutions is that you are helping to build the service providers' link authority and not your own, and you have much less control over optimizing the content. Some hosted solutions can pose even bigger problems if the content and functionality are embedded into your site with JavaScript. Examples of this include leading customer review solutions such as BazaarVoice and PowerReviews.

Google announced in November 2011 that it is continuing to expand its ability to execute JavaScript, and it is known that Google can now index Facebook Comments. It may be able to read reviews implemented with BazaarVoice or PowerReviews in the near future, but what it can and cannot execute is not fully known. To be safe, one novel solution to the JavaScript problem is to execute the JavaScript, extract the content from its encrypted form, and present it in plain-text format so that the search engines can see it.

Flash

As referenced several times earlier in this chapter, Flash is popular on the Web, but it presents challenges to the search engines in terms of indexing the related content. This creates a gap between the user experience with a site and what the search engines can find on that site.

It used to be that search engines did not index Flash content at all. In June 2008, Google announced that it was offering improved indexing of this content (*http://googlewebmastercentral.blogspot.com/2008/06/improved-flash-indexing.html*). This announcement indicates that Google

can index text content and find and follow links within Flash files. However, Google still cannot tell what is contained in images within a Flash file. Here are some reasons why Flash is still not fully SEO-friendly:

Different content is not on different URLs
: This is the same problem you encounter with AJAX-based pages. You could have unique frames, movies within movies, and so on that appear to be completely unique portions of the Flash site, yet there's often no way to link to these individual elements.

The breakdown of text is not clean
: Google can index the output files in the *.swf* file to see words and phrases, but in Flash, a lot of your text is not inside clean `<h1>` or `<p>` tags; it is jumbled up into half-phrases for graphical effects and will often be output in the incorrect order. Worse still are text effects that often require "breaking" words apart into individual letters to animate them.

Flash gets embedded
: A lot of Flash content is only linked to by other Flash content wrapped inside shell Flash pages. This line of links, where no other internal or external URLs are referencing the interior content, leads to documents with very low PageRank/link juice. Even if they manage to stay in the main index, they probably won't rank for anything.

Flash doesn't earn external links like HTML
: An all-Flash site might get a large number of links to the home page, but interior pages almost always suffer. When people implement links to embeddable Flash they normally point to the HTML host page, rather than any of the interior pages within the Flash.

SEO basics are often missing
: Anchor text, headlines, bold/strong text, image `alt` attributes, and even title tags are not simple elements to properly include in Flash. Developing Flash with SEO in mind is just more difficult than doing it in HTML. In addition, it is not part of the cultural lexicon of the Flash development world.

A lot of Flash isn't even crawlable
: Google has indicated that it doesn't execute external JavaScript calls (which many Flash-based sites use) or index the content from external files called by Flash (which, again, a lot of Flash sites rely on). These limitations could severely impact what a visitor can see versus what Googlebot can index.

Note that it used to be that you could not test the crawlability of Flash, but the Adobe Search Engine SDK does allow you to get an idea as to how the search engines will see your Flash file.

Flash Coding Best Practices

If Flash is a requirement for whatever reason, there are best practices you can implement to make your site more accessible to search engine spiders. The following are some guidelines on how to obtain the best possible results.

Flash meta tags

Beginning with Adobe/Macromedia Flash version 8, there has been support for the addition of title and description meta tags to any *.swf* file. Not all search engines are able to read these tags yet, but it is likely that they will soon. Get into the habit of adding accurate, keyword-rich title tags and meta tags to files now so that as search engines begin accessing them, your existing *.swf* files will already have them in place.

Adobe Flash Search Engine SDK

Flash developers may find the SDK useful for server-based text and link extraction and conversion purposes, or for client-side testing of their Flash content against the basic Adobe (formerly Macromedia) Flash Search Engine SDK code.

Tests have shown that Google and other major search engines now extract some textual content from Flash *.swf* files. It is unknown whether Google and others have implemented Adobe's specific Search Engine SDK technology into their spiders, or whether they are using some other code to extract the textual content. Again, tests suggest that what Google is parsing from a given *.swf* file is very close to what can be extracted manually using the Search Engine SDK.

The primary application of Adobe's Search Engine SDK is in the desktop testing of *.swf* files to see what search engines are extracting from a given file. The program cannot extract files directly from the Web; the *.swf* file must be saved to a local hard drive. The program is DOS-based and must be run in the DOS command prompt using DOS commands.

Running a *.swf* file through the Flash SDK *swf2html* program during development gives you a chance to edit or augment the textual assets of the file to address the best possible SEO practices, homing in primarily on keywords and phrases along with high-quality links. Because of the nature of the Flash program and the way in which it deals with both text and animation, it is challenging to get exact, quality SEO results. The goal is to create the best possible SEO results within the limitations of the Flash program and the individual Flash animation, rather than to attempt the creation of an all-encompassing SEO campaign. Extracted content from Flash should be seen as one tool among many in a larger SEO campaign.

Internal Flash coding

There are several things to keep in mind when preparing Flash files for SEO:

- Search engines currently do not read traced text (using the `trace()` function) or text that has been transformed into a shape in Flash (as opposed to actual characters). Only character-based text that is active in the Flash stage will be read (see Figure 6-46).

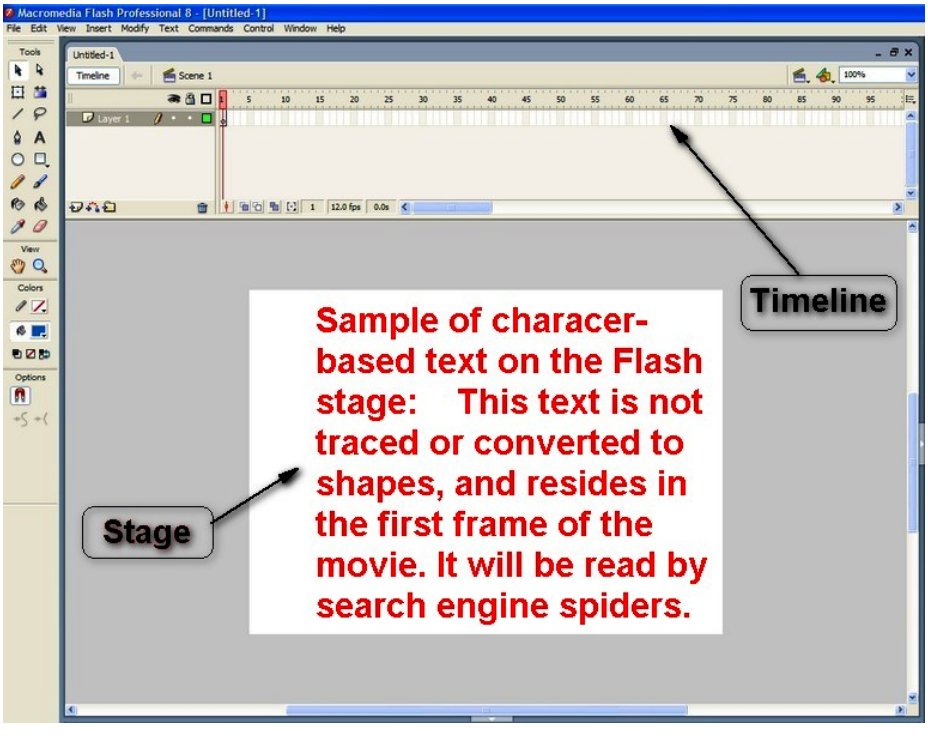

FIGURE 6-46. Example of spider-readable text inside a Flash program

- Animated or affected text often creates duplicate content. Static text in Flash movies is not read as the duplicate instances that "tweening" and other effects can create. Use static text, especially with important content, so that search engines do not perceive the output as spam (see Figure 6-47).
- Search engine spiders do not see dynamically loaded content (text added from an external source, such as an XML file).
- The font size of text does not affect search engines; they read any size font.
- Special characters such as <, >, &, and " are converted to HTML character references (*<*, *>*, *&*, and *"*) and should be avoided.
- Search engines find and extract all URLs stored within the `getURL()` command.
- Search engines have the ability to follow links in Flash, though it is an iffy proposition at best. They will not, however, follow links to other Flash *.swf* files. (This is different from loading child *.swf* files into a parent *.swf* file.) Therefore, links in Flash should always point to HTML pages, not other *.swf* files.

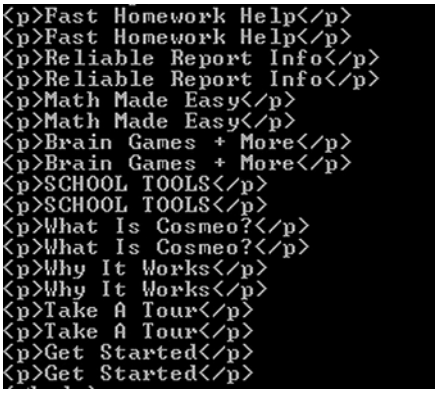

FIGURE 6-47. Animated text results in Flash source; can be seen as duplicate content

SWFObject and NoScript tags

Because alternative content workarounds for SEO of Flash files have been historically abused by spammers, we cannot recommend using these tactics to optimize your Flash files without providing a critical disclaimer.

Both the `SWFObject` and `NoScript` methods were originally designed to be legitimate, graceful degradation tactics that would be acceptable by the search engines as a way to accommodate older browsers or people with special needs. But many unscrupulous sites have used this code to trick search engine spiders. In other words, these methods are used in such a way that browsers display one thing to users, but display something completely different to search engine spiders. As discussed earlier in this chapter, all of the major search engines disapprove of such tactics.

Websites using such methods today are often penalized or removed from search engine indexes altogether. This makes graceful degradation somewhat risky, although if the methods are used clearly within the boundaries for which they were intended, getting penalized or banned is highly unlikely.

Intent is an essential element search engines take into consideration. If your intent is to provide *all* users with a positive experience while visiting your site, you should be fine. If your intent is to game the search engines, all it takes is one online rival to report your site for spam to incur the wrath of the search engines.

Google and other search engines do not algorithmically ban sites for using `SWFObject` and `NoScript` tags; it usually requires human intervention to evoke a penalty or outright ban.

SWFObject. With regard to Flash optimization, `SWFObject` is the better of the two options because it is JavaScript code designed specifically for Flash *.swf* purposes, and it has been abused to a lesser extent than the `NoScript` tag option.

SWFObject is Flash detection code written in JavaScript that checks whether a browser has the Flash plug-in. If the browser does have the Flash plug-in, the *.swf* file is displayed secondary to that detection. If the browser does not have the Flash plug-in or the JavaScript to detect it, the primary, alternative content contained within tags is displayed instead. The key here is that search engine spiders do not render the JavaScript: they read the primary content in the tags.

The opportunity for abuse is obvious upon viewing the code. This small piece of code is placed within the <head> tags:

```
<script type="text/javascript" src="swfobject.js"></script>
```

In the body of the text, the code looks something like Figure 6-48.

```
<script type="text/javascript" src="swfobject.js"></script>

<div id="flashcontent">
      Text, links, and graphics placed here are replaced by the Flash movie. Search
      engine spiders will read this information, but the browser with an active Flash
      plugin will show the Flash movie instead.
</div>

<script type="text/javascript">
      var so = new SWFObject("whatever.swf", "themovie", "200", "100", "7", #336699");
      so.write("flashcontent");
</script>
```

FIGURE 6-48. Information between the HTML tags is read by search engine spiders

Search engine spiders will read text, links, and even alt attributes within the tags, but the browser will not display them unless the Flash plug-in isn't installed (about 95% of browsers now have the plug-in) or JavaScript isn't available.

Once again, the key to successfully implementing SWFObject is to use it to the letter of the law; leverage it to mirror the content of your Flash *.swf* file *exactly*. Do not use it to add content, keywords, graphics, or links that are not contained in the file. Remember, a human being will be making the call as to whether your use of SWFObject is proper and in accordance with that search engine's guidelines. If you design the outcome to provide the best possible user experience, and your intent is *not* to game the search engines, you are probably OK.

You can download the SWFObject JavaScript free of charge at *http://code.google.com/p/swfobject/*. Included in this download is the *flashobject.js* file, which is placed in the same directory as the web pages upon which the corresponding calling code resides.

NoScript. The NoScript tag has been abused in "black hat" SEO attempts so frequently that caution should be taken when using it. Just as SWFObject and tags can be misused for link and keyword stuffing, so too can the NoScript tag. Certain companies have promoted the misuse of the NoScript tag widely; consequently, there have been many more problems with its use.

That being said, conservative and proper use of NoScript tags specifically with Flash *.swf* files can be an acceptable and good way to get content mirrored to a Flash file read by search engine

spiders. As with `SWFObject` and corresponding `<div>` tags, the content must echo that of the Flash *.swf* movie exactly. Do not use these tags to add content, keywords, graphics, or links that are not in the movie. Again, it is a human call as to whether a site or individual page is banned for the use or misuse of `NoScript` tags.

You use `NoScript` tags with Flash *.swf* files in the following manner:

```
<script type="text/javascript" src="YourFlashFile.swf"></script>
```

followed at some point afterward by:

```
<noscript>
<H1>Mirror content in Flash file here.</H1>
<p>Any content within the NoScript tags will be read by the search engine
spiders, including links
http://www.mirroredlink.com, graphics, and corresponding alt attributes.
</noscript>
```

For browsers that do not have JavaScript installed or functioning, content alternatives to JavaScript-required entities are displayed. So, for use with Flash *.swf* files, if a browser does not have JavaScript and therefore cannot display Flash, it displays instead the content inside the `NoScript` tags. This is a legitimate, graceful degradation design. For SEO purposes, as with `SWFObject`, the search engine spiders do not render the JavaScript and do read the content contained in the HTML. Here, it is the content in the `NoScript` tags.

Scalable Inman Flash Replacement (sIFR)

sIFR is a technique that uses JavaScript to read in HTML text and render it in Flash. The essential fact to focus on here is that the method guarantees that the HTML content and the Flash content are identical. One great use for this is to render headline text in an antialiased font (this is the purpose for which sIFR was designed). This can provide a great improvement in the presentation of your site.

At a Search Engine Marketing New England (SEMNE) event in July 2007, Dan Crow, head of Google's Crawl team, said that as long as this technique is used in moderation, it is OK. However, extensive use of sIFR could be interpreted as a poor site quality signal. Since sIFR was not really designed for large-scale use, such extensive use would not be wise in any event.

It is worth noting that there are similar technologies available to web designers for improved type presentation, which provide similar search engine friendliness. FLIR (FaceLift Image Replacement) is an image replacement script similar to sIFR in its use of JavaScript, but without the Flash element, and there is a handy WordPress plug-in for implementation on WordPress-based websites (*http://wordpress.org/extend/plugins/facelift-image-replacement/*).

Best Practices for Multilanguage/Country Targeting

Many businesses target multiple countries with their websites, and for such businesses, various questions arise. Do you put the information for your products or services all on the same

domain? Do you obtain multiple domains? Where do you host the site(s)? It turns out that there are SEO factors, as well as basic marketing questions, that affect the answers. Of course, there are also non-SEO factors, such as the tax implications of what you do; further, for some TLDs you can only register a domain if you have a local physical presence (e.g., France requires this to get a *.fr* domain).

Targeting a Specific Country

Starting with the basics of international targeting, it is important to let the search engines know where your business is based in as many ways as possible. These might include:

- Using a country-specific TLD (ccTLD) for your domain (e.g., *.co.uk*)
- Hosting your site locally, not abroad
- Including the physical local address in plain text on every page of your site
- Setting Google Webmaster Central geotargeting to your country of interest
- Verifying your address with Google Maps
- Getting links from in-country websites
- Using the local language on the website

If you are starting from scratch, getting all these factors lined up will give you the best possible chance of ranking in the local country you are targeting.

Problems with Using Your Existing Domain

If you're expanding into a new country, you may be wondering why you cannot leverage your current domain's weight to target the new territory rather than starting from scratch—in other words, why can't you create multiple versions of your site and determine where the user is in the world before either delivering the appropriate content or redirecting that user to the appropriate place in the site (or even to a subdomain hosted in the target country)?

The problem with this approach, from an SEO perspective, is that the major search engines spider from the United States, so their IP addresses will be in the United States in your lookup, and they will therefore be delivered only US content. This problem is exacerbated if you are going even further and geodelivering content in different languages, as only your English language content will be spidered unless you cloak for the search engine bots.

This kind of IP delivery is therefore a bad idea. You should make sure you do not blindly geodeliver content based on IP address, as you will ignore many of your markets in the search engines' eyes.

The Two Major Approaches

The best practice remains one of two approaches, depending on the size and scale of your operations in the new countries and how powerful and established your *.com* domain is.

If you have strong local teams and/or (relatively speaking) less power in your main domain, launching independent local websites geotargeted as described earlier (hosted locally, etc.) is a smart move in the long run.

If, on the other hand, you have only centralized marketing and PR and/or a strong main domain, you may want to create localized versions of your content either on country-specific subdomains (*http://uk.yourdomain.com*, *http://au.yourdomain.com*, etc.) or in subfolders (*/uk/*, */au/*, etc.), with the preference being for the use of subdomains so that you can set up local hosting.

Both the subdomains and subfolders approaches allow you to set your geotargeting option in Google Webmaster Central, and with either method you have to be equally careful of duplicate content across regions. In the subdomain example, you can host the subdomain locally, while in the subfolder case, more of the power of the domain filters down.

Unfortunately, the Webmaster Tools geotargeting option doesn't work nearly as well as you might hope to geotarget subfolders. The engines will consider hosting and ccTLDs, along with the geographic location of your external link sources, to be stronger signals than the manual country targeting in the tools. In addition, people in other countries (e.g., France) don't like to click on *.com* or *.org* TLDs: they prefer *.fr*. This extends to branding and conversion rates too—web users in France like to buy from websites in France that end in *.fr*.

Multiple-Language Issues

An entire treatise could be written on handling multilanguage content, but the search engines themselves are rapidly evolving in this field, and tactics are likely to change dramatically in the near future. Therefore, this section will focus on providing you with the fundamental components of successful multilanguage content management.

Here are best practices for targeting the search engines as of this writing, using Spanish and English content examples:

- If you have content in Spanish and English serving the same country:
  - — Create a single website with language options that change the URL by folder structure; for example, *http://www.yourdomain.com* versus *http://www.yourdomain.com/esp/*.
  - — Build links from Spanish- and English-language sites to the respective content areas on the site.
  - — Host the site in the country being served.

— Register the appropriate country domain name (for the United States, *.com*, *.net*, and *.org* are appropriate, whereas in Canada using *.ca* or in the United Kingdom using *.co.uk* is preferable).

- If you have content in Spanish and English targeting multiple countries:
  - — Create two separate websites, one in English targeting the United States (or the relevant country) and one in Spanish targeting a relevant Spanish-speaking country.
  - — Host one site in the United States (for English) and the other in the relevant country for the Spanish version.
  - — Register different domains, one using US-targeted domain extensions and one using the Spanish-speaking country's extension.
  - — Acquire links from the United States to the English site and links from the Spanish-speaking country to that site.
- If you have content in Spanish targeting multiple countries:
  - — Create multiple websites (as mentioned earlier) targeting each specific country.
  - — Register domains in each country, using the appropriate country TLD and hosting them locally.
  - — When possible, have native speakers fluent in that region's dialect write the site content for each specific country.
  - — Obtain in-country links to your domains.

Although some of these approaches may seem counterintuitive, you must take into account the joint issues of search engines preferring to show content on country-specific domains hosted in those countries and duplicate content problems. Creating multiple websites is never ideal due to the splitting of link equity, but in the case of international targeting it is often difficult (or even impossible) to rank a US-hosted *.com* address for foreign-language content in another country.

Conclusion

By now you should be aware that a search engine–friendly website is the first step toward SEO success. In the next chapter we will demonstrate how links are also a critical piece of the SEO puzzle—particularly when targeting highly competitive terms. However, if you have not made your site crawler-friendly and optimized, all of your other efforts—whether they be link development, social media promotion, or other tactics to improve search visibility and increase search traffic—will be wasted. A website built from the ground up to optimal specifications for crawler accessibility and top rankings is the foundation from which you will build all SEO initiatives. From this solid foundation, even the loftiest of goals are within reach.