Part I

Discovering Amazon Web Services

1 Learning About the Amazon Web Services

2 Using Amazon Web Services to Your Advantage

3 Working with Web Service Data

Chapter 1

Understanding Web Services | Working with Amazon Web Services | Using Amazon Web Services to Your Advantage

Learning about Amazon Web Services

Getting the Amazon Web Services Kit | Installing the Kit on a Machine | Considering the Setup of the Development System

You might have already heard quite a bit about Amazon Web Services without really understanding what the phrase means. Amazon is an amazing company. Unlike many companies that created an online presence and are no longer in business, Amazon is still around today and thriving. In fact, the company's owner, Jeff Bezos, was recently quoted as saying the Amazon Services subsidiary (the one that provides your online experience among other things) might eventually become their main business (see the InfoWorld article at `http://www.infoworld.com/article/03/06/10/HNamazondives_1.html?business` for details). Given all that Amazon is doing, it's easy to understand how all of these new services could become confusing even to seasoned professionals.

This chapter helps you understand what Amazon Web Services can do for you and how it fits within your business or research plans. It also discusses what you need to know in order to best use Amazon Web Services for your particular needs. Although you might think that Amazon Web Services is only good for people who want to act as associates and sell goods through their Web site, Amazon Web Services is capable of performing myriad other interesting tasks. For example, many individuals use Amazon Web Services as a means for performing research without ever engaging in buying or selling products.

Once the chapter gets past the basic concepts behind the Amazon Web Service, it demonstrates how to obtain the required resources and set them up on your machine. For example, you need both the Amazon Web Services Kit and a developer token to perform basic tasks. In fact, the chapter describes setup scenarios that you should consider to make your machine better reflect what the user of the Web site, document, or application you build will see. Some situations might require you to set up a system that also includes alternative devices such as a Personal Digital Assistant (PDA). In short, when you complete this chapter, you'll have a copy of the Amazon Web Services Kit installed on a machine configured to meet specific requirements.

Understanding Amazon Web Service

Whenever a new technology appears on the scene, it's important to compare it with other technologies you might have used in the past. The comparison process often helps you decide how this new technology differs from what you used in the past and reduces problems caused by hype. The media might try to convince you that a new product or service is something completely different, when in fact it's merely an update of an existing technology. Currently, there's a lot of hype about Web services that makes them sound like something new and very complex. This section of the chapter defines Web services generally, examines Amazon Web Services specifically, and compares this technology to older technologies. What you'll find might surprise you because Web services are really a new implementation of an old technique.

What Is a Web Service?

You can look at a Web service from a number of perspectives. The easiest way to view a Web service is as a means of obtaining access to information. Essentially, you ask the server for information and the server returns that information in some form. The request and the returned information normally appear in eXtensible Markup Language (XML) form. Using XML preserves the meaning behind the information, regardless of the diversity of the platforms involved, so that you receive not only the information, but understand the context in which the information is used. The "Understanding XML Basics" section of Chapter 3 tells you more about XML. All you need to know now is that you receive information in XML format.

▶ NOTE

Don't confuse *new* with *useful*. Web services are very useful because they add new functionality to an existing idea that has worked for a long time. They're also new in that they use a different process from other technologies. However, the technology itself builds on other techniques that you have already used in some way. In sum, the implementation is new, the process is useful, but the technique is the same one you've used in the past.

From an Amazon Web Services perspective, you request information on one or more products, vendors, or other search criteria. The request defines the kind of information you want to know and how detailed that information will be. Amazon Web Services returns the information you request (when available) in a standardized format.

A Web service also performs some type of useful work. The useful work might be something as simple as interpreting your request, calculating the answer, and sending the result back. In the case of the Amazon Web Service, the Web service accepts your request (normally some search criteria), interacts with the database through a search engine to obtain the information you requested, and sends the information back to you. You can also perform other

tasks using the Amazon Web Service, such as making a purchase or selling goods to others. The rest of the book shows how to perform all of these tasks.

The final consideration for a Web service (at least from the Web service user perspective) is that it executes on the remote machine, not on your machine. In short, this means you're using resources on that other machine with the permission of the machine's owner. The remote machine can set requirements for using the Web service, as well as require you to perform specific setup and security checks as part of your request. In the case of the Amazon Web Service, you need to obtain this permission by requesting it as part of the Amazon Web Services Kit download process. The "Getting a Developer Token" section of this chapter tells how to obtain the required permission and what this permission means to you.

How Do Web Services Work?

Many people fear new technology because they don't understand how it works, and many of those who do know how it works enjoy the mystique of knowledge too much to share it with anyone else. Web services are actually quite easy to understand if you look at them in a way that relates the task to everyday occurrences. For example, you might compare the operation of a Web service to making a withdrawal at the bank—the process really is the same. The one thing to remember is that the process a Web service uses to perform a task is always the same. No matter what technology you use to make a request or receive a response, the steps are still the same. Here are the steps that most Web services, including the Amazon Web Service, use to complete a transaction.

1. *The client discovers the Web service*. During the act of discovery, the client might do things like download a file that tells how to interact with the Web service. This step is the same as someone walking into the bank. The person knows the bank exists and the bank teller might have noticed the person. The bank posts the rules for making a withdrawal or the teller might help a first-time banker understand the rules.

▶ TIP

You may find that Amazon Web Services is so indispensable that you'll want to work with Web services from other vendors. For example, Microsoft supports the MapPoint Web Service (`http://www.microsoft.com/mappoint/net/`). In time, standards organizations will set up directories of these Web services that you can access with ease. In the meantime, you can search for companies that offer Web services using the Web Services Finder page at `http://www.15seconds.com/WebService/`. Some people have problems using the Web Services Finder. In some cases, you'll need to use a specialty Web service list such as the one at `http://www.flash-db.com/services/`.

> **NOTE**
>
> Amazon's database *schema* specifies the format of the information. A schema defines the organization of information in a database. You can learn more about the format of the data in the "Previewing the Amazon Database" section of Chapter 3.

2. *The client makes a request based on the rules delivered during the discovery phase*. The rules might specify that the request has to appear in a certain form and the client must provide specific data. This step is the same as the person walking up to the teller's window with a withdrawal request. The request must contain the person's account number, the amount they wish to withdraw, and other identifying information. The bank specifies the format of the request and the information it must contain.
3. *The server might ask the client for credentials depending on the openness of the Web service*. Amazon Web Services is public but still requires that you supply a developer token as identification. This step is the same as the bank teller asking you for a driver's license or other form of identification before honoring your withdrawal request.
4. *The Web service performs the work required to honor your request*. In most cases, the Web service accesses a database for information, it could enter an order, and it might even provide some level of formatting. Amazon Web Services performs a number of tasks depending on the request you make. The easiest request is a product search. This step equates to the bank teller getting the money from the drawer and counting it.
5. *The Web service sends the data to the client*. The content of the information depends on the Web service. Amazon Web Services provides data in a very specific format based on the content of the associated database and the nature of the request. This step equates to the teller handing the person their money. In general, the teller orders the money in a specific way and counts it out to the person, rather than simply handing the money over.
6. *The client logs out of the Web service or the Web service disconnects the client after some period of inactivity*. This step equates to the person leaving the bank, money in hand. If the person doesn't leave the bank (they just hang out in the lobby), you can be sure that someone will ask them to leave.
7. *The client does something with the data it receives*. In many cases, it formats the data and presents it on screen for the user. This step equates to the person spending the money they receive from the bank.

You can add any amount of complexity needed to the individual steps, but these seven steps define the process every Web server follows. When you break a Web service down into these seven steps, the process that used to appear as magic suddenly becomes quite doable. Chapters 5 through 9 are essentially options you can use to perform these seven steps using different technologies. This book explores the seven steps using various languages and platforms—Amazon Web Services makes information available to just about anyone who needs

it. However, it's important to remember that everything comes down to a client making a request and the Web server returning data.

Considering the Usage Requirements

There's no free lunch. Some people would have you believe that the Web service does everything for you and that the client does nothing at all. However, the client interacts with the Web service, which means the client must possess some intelligence to perform the task. To use a Web service, you must understand the usage requirements.

From a client perspective, the type of device you use to access the Web service determines the access speed, as well as what you can do with the data once you receive it. Although a PDA such as the Pocket PC can access Amazon Web Services just fine, most people wouldn't want to use it to place orders. About the best you can hope for is to search for individual products or to perform other kinds of simple research. On the other hand, a desktop or laptop machine has all of the processing power, screen real estate, and functionality to perform any task. Amazon Web Services hasn't changed, but the capability of the client has.

Amazon Web Services also has some usage requirements and these requirements might change the way that you use your client. For example, according to the license agreement (see Appendix B for details) you can't make more than one request per second. In addition, you can't send a file larger than 20KB to Amazon Web Services. Some requests let you supply information to Amazon, so it's possible to exceed the size limitation. These two limitations ensure the Amazon servers won't become overloaded, but they also mean you must provide some type of monitoring in your application to prevent abuse of the licensing terms.

Amazon is interested in protecting its intellectual property. A Web service is essentially a borrowed resource—you're using information supplied by someone else with resources on the remote server. Consequently, you'll find some restrictions on what you can do with the data that Amazon supplies through its Web service. Although you can display the data in any format desired, you can't take credit for the data or modify it (except to shorten it as necessary). You also have to display any

▶ NOTE

This book discusses a number of mobile devices. The Pocket PC provides additional functionality and features that make it a better target for some types of applications than devices such as the Palm. On the other hand, most Palm devices are much easier to carry and cost less than the Pocket PC. This book examines the entire range of mobile devices to ensure you understand the limitations of using a specific device to access Amazon Web Services. I'm not saying one device is better than another—simply that one device works better than the other for a given application.

> **WARNING**
>
> If you violate the licensing terms, Amazon Web Services simply denies your request. In addition, you might receive a message from Amazon requesting that you adhere to the terms of usage for the Web service.

copyright or other notices provided with the data. These provisions ensure that someone viewing the data you obtain from Amazon sees the same information that they'll find on the Amazon site.

Old data is a problem. The sales rank for a book might change or Amazon could change the price of a product. Consequently, when using Amazon Web Services you must guarantee that you'll refresh the data every 24 hours when you don't provide prices or availability, or every hour when you do. The licensing agreement provides an email address you can use to request changes to this policy. Data update requirements become a problem when you choose to store the data locally, rather than obtain it from Amazon each time. Developers will store the data locally to obtain a performance benefit—it takes less time to use the local copy of a product description than to request it from Amazon. Obviously, this particular requirement won't be a problem for most Pocket PC users because local storage is at a premium, but it's something you do need to consider for every other environment.

Discovering Uses for Amazon Web Service

The main reason to use Amazon Web Services is to gain personal, company, or monetary benefit. Many people view Amazon Web Services as a way to earn extra money by becoming an associate. However, there are other ways to use this Web service and many of them don't involve an exchange of money. Although this book will provide a lot of examples on how to leverage Amazon Web Services on a Web site or as part of an application to act as an associate, you'll also see other examples that show other uses. (See the "Developing a Shopping Cart Application" section in Chapter 12 for an in-depth explanation of how to develop a Web application that allows users to buy and sell products from Amazon through your Web site.)

Using Overviews and Details

No matter what you plan to do with Amazon, you need to know how to search for the products and services you want. It might be tempting to say that you could use the search engine interface provided by Amazon to perform this task, but the Amazon display is somewhat limiting in that you need to perform several searches to obtain the information you want in most cases. For example, suppose you want to find a book on working with exotic fish. To ensure the book contains information on tetras, you need to look through the table of contents for each book. Although the search engine provides this information, getting to it online can be cumbersome because you have to look up all of the available books that discuss exotic fish,

select an individual book from the list, locate the table of contents, review the table of contents one page at a time, and hope that you find what you need. If the first book doesn't contain the information you need, you must perform the same process with the second and subsequent books. This search process could take all day.

Using an application to hunt down the information for you is much simpler. A program that relies on Amazon Web Services can download all of the books that discuss exotic fish, including the table of contents for each book. (Not every publisher supplies a table of contents for online viewing, so you might not be able to perform this search on every book.) The program can then do the searching for you. Your program would only display the books that actually mention tetras in the table of contents.

Searching for information on tetras is one example of a far more complex process, but it gives you a good understanding of how you can use custom search programs to your benefit. In this case, the custom program saves substantial time, lets you locate the precise book you need quickly, and still garners a sale for Amazon. It is the perfect example of how Web services can pay off for everyone involved. Amazon could never design a search engine interface that meets such specific needs—the developers at Amazon have to create a search engine interface that works for the majority of people who visit the site. In sum, a generic search engine can serve most needs, but it might not be the most efficient choice for everyone.

Amazon doesn't require you to know specific product names or any specifics for that matter. Searching for information lets you refine the criteria for locating specific products. You can specify searches using any of the criteria in the following list (Amazon might add other criteria in a future update).

Amazon Standard Item Number (ASIN) Amazon assigns a unique number to every product in their stock called an ASIN. You can use the ASIN, if you know it, to obtain information about a specific book. For books, Amazon uses the ISBN number without any punctuation the book publisher might provide. For example, an ISBN of 0-7821-4134-X becomes an ASIN of 078214134X. The correlation isn't always direct, but you can learn more about the ASIN topic essentials in the kit at `\AmazonWebServices\API Guide\search_asin_isbn.htm`.

▶ NOTE

The licensing agreement you see when you sign up for an Amazon Web Services developer token includes a number of other requirements for using the Web service. For example, you must create a link back to the Amazon site for every Amazon product you display on your Web site, even if you derive this information from the Amazon data in some way that isn't apparent. Make sure you understand these requirements before you begin a project. Appendix B can also help you create a checklist for your application that ensures compliance with the requirements.

Browse Identification Number A browse identification number (Browse ID) lets you search by product classification. For example, you might want to find bargain books. In this case, you'd use the Browse ID of 45. Likewise, technical books have a Browse ID of 173507. You'll find a list of common Browse IDs in the kit at `/AmazonWebServices/API Guide/common_modes.htm`.

Keyword The term keyword is somewhat misleading. A keyword can be anything from the ISBN of a book to the author's name. When searching for a CD, you can choose the artist's name. You can also select movies by genre or products by type. Any information that relates back to the product in some way can appear as a keyword. Obviously, some keywords work better than others, so you need to experiment a little to determine which keywords work best. In addition, you might find that you need to filter the information you receive to ensure you display just the products you want. You can learn more about the essentials of keyword searches in the kit at `/AmazonWebServices/API Guide/search_keywords.htm`.

Listmania® ID Amazon is very much customer driven. This way of doing business appears in every aspect of the online experience from the customer reviews to the use of the Listmania feature. Listmania is a collection of customer classifications. For example, you might find a category called best DVD or most thrilling adventure. You must actually search online to discover the Listmania ID numbers because Amazon doesn't provide a list of them in the kit. The number appears as part of the URL for the Listmania search. Figure 1.1 shows a typical example of a Listmania entry. Notice that the URL includes the number 19J6Y001ZYYD3. This is the Listmania ID. You can learn more about this search type in the kit at `/AmazonWebServices/API Guide/search_listmania.htm`.

Wish List A wish list search works very much the same as a Listmania ID search. The big difference is that someone registers wish items in a registry. Knowing the Wishlist ID lets you access this list and buy products for that person based on the entries. This is a great option to make birthdays or other special events easier to handle. You can learn more about wish list searches by looking in the kit at `/AmazonWebServices/API Guide/search_wishlist.htm`.

▶ TIP

Some people think that the only product that Amazon sells is books. However, you can buy many products through Amazon, including products from other vendors such as Toys"R"Us. For more information on the Toys"R"Us connection, check the BusinessWeek story at `http://www.businessweek.com/2000/00_43/b3704050.htm`. Another good place to look for information is the World Trade magazine article at `http://www.worldtrademag.com/CDA/ArticleInformation/coverstory/BNPCoverStoryItem/0,3481,89675,00.html`. (Note that you might have to sign up as a subscriber to access the World Trade magazine article.) The bottom line is that you should become familiar with all of the products you can obtain through Amazon to ensure you make maximum benefit of its Web service offering.

FIGURE 1.1:
Use Listmania ID numbers to access specific product collections.

Now that you have a better idea of why you would want to build a custom search engine, it's time to discuss specific kinds of searches. The following sections describe three kinds of searches that you might want to perform. This list simply contains suggestions. As you become more knowledgeable about the Amazon database structure, you'll discover other kinds of searches that meet your specific needs.

Locating Books

Many people will begin their Amazon Web Services experience by looking for books because they're most familiar with that part of Amazon. You can perform searches using any of the four techniques listed at the introduction to this section. In addition to these standard techniques, you can also use a number of techniques specifically suited to meet the needs of book searches. The following list describes each of these searches.

Author Many people like to search by author name because they have read other works by that author and wish to find something new. When using the browser interface to the Amazon search engine, you can specify the author name as a keyword. However, the advanced search shown in Figure 1.2 lets you enter the author name as a specific item,

which usually produces better results because the search engine understands the search type better. (You can find the advanced search page at `http://www.amazon.com/exec/obidos/ats-query-page/`.) Fortunately, the Amazon Web Services interface lets you specify author as a separate item as well as long as you specify the search type as a book. You can learn more about this kind of search by looking in the kit at `/AmazonWebServices/API Guide/search_author.htm`.

International Standard Book Number (ISBN) Like the author search, you can specify an ISBN (including the punctuation) as part of a keyword search. When you want to locate a specific book, use the ASIN search method described earlier. However, you can actually obtain better results for all books that include this ISBN as part of the descriptive data if you use a keyword search. For example, you might find that you need to perform a keyword search when you want to locate a set of books, rather than a single book. You can learn more about the ISBN option by looking in the kit at `/AmazonWebServices/API Guide/search_asin_isbn.htm`.

Power Search You might have noticed that Figure 1.2 includes more than just the author name and ISBN. Amazon Web Services lets you supply other input values for books using the power search technique. This technique only works with books and not with any other product type (including third party products). A power search can include the following keywords: Title, Subject, Author, Keyword, ISBN, Publisher, Language, and Publication. You can learn more about power searches by looking in the kit at `/AmazonWebServices/API Guide/search_power.htm`.

Locating Other Products

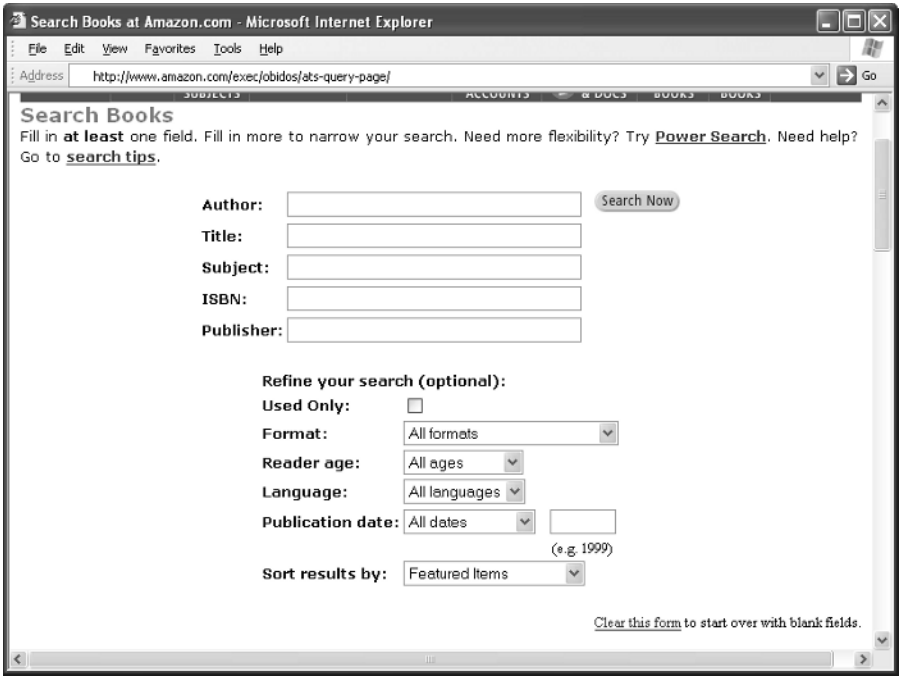
Amazon is directly involved in providing access for third parties, both large and small. For example, you can locate and purchase toys from Toys"R"Us and music from CDNOW through Amazon. (In fact, the `http://www.cdnow.com` URL redirects your browser to `http://www.amazon.com/exec/obidos/tg/browse/-/3023481/`.) You can even build a Web service application to handle the task. Likewise, you can find movies, music, dishes, apparel, and a long list of other items. Look at `http://www.amazon.com/exec/obidos/subst/home/all-stores.html/` for a complete list of stores that Amazon currently works with.

▶ TIP

This list is by no means complete. I often experiment with the online engine to see what search methods are practical and what results they produce. For example, the keyword category covers a lot of ground because you can use practically anything for a keyword. However, not all keywords work equally well. For example, entering an author's full name in the search box might not yield all of the books that are available for that author. (Some authors only use their first and last name.) On the other hand, entering just the author's last name may yield too many results. I find that specifying just the first and last name provides satisfactory results in most cases.

FIGURE 1.2:

The advanced search lets you locate books based on specific input.



Although you can't find these items using a power search, Amazon Web Services does make them available through other means. For example, you can search for movies by the actor, actress, or director. The following list describes some of the searches you can perform.

Actor/Actress Generally, you'll use this kind of search to locate a favorite movie. The search doesn't work with non-movie searches because the Web service assumes you'll use a keyword search for other types of actor/actress searches (such as locating a book). You can learn more about actor/actress searches by looking in the kit at `/AmazonWebServices/API Guide/search_actor.htm`.

Artist/Musician An artist/musician search helps you locate music. You can't use this search to perform tasks such as locating a famous painting because the Web service doesn't view an artist in that way. If you need to

▶ NOTE

A few of the searches in the "Locating Other Products" section also apply to books. For example, you can perform similarities and UPC searches on books. As Amazon improves their Web service, you'll find additional crossover between product categories. When you have doubts about the usefulness of a search, try it and see what results you get.

find an artist (the kind that paints or creates some other form of physical art), then you need to perform a keyword search. You can learn more about artist/musician searches by looking in the kit at `/AmazonWebServices/API Guide/search_artist.htm`.

Director As with the actor/actress search, this search type relates to movies. You can learn more about director searches by looking in the kit at `/AmazonWebServices/API Guide/search_director.htm`.

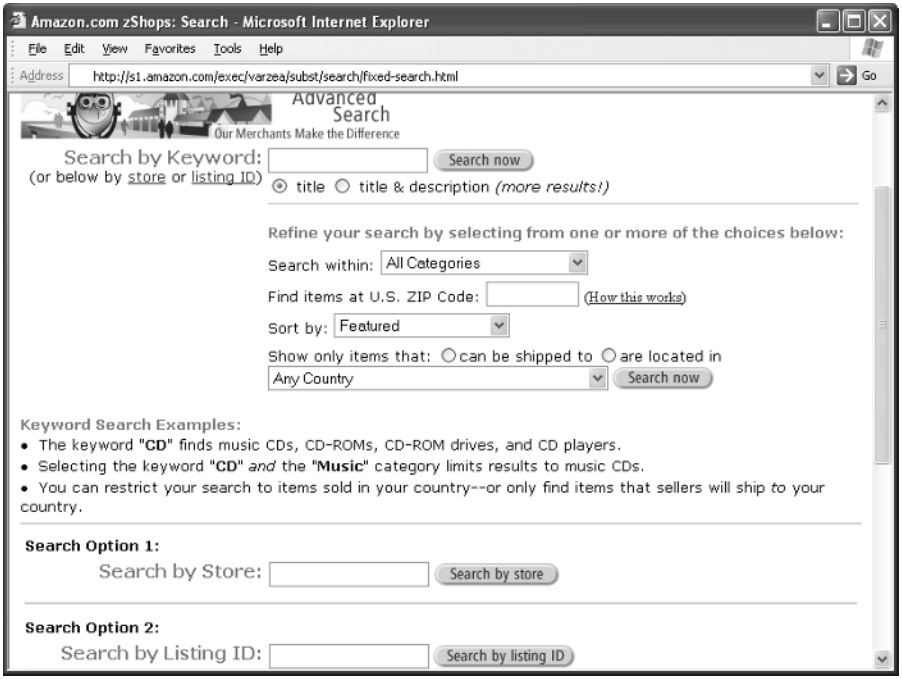
Exchange Every third party product offered by Amazon has an exchange identifier. This identifier works much like the Browse ID and Listmania ID mentioned earlier in the chapter. Amazon doesn't provide an Exchange ID list, so you need to perform a search to find the one you need. The Exchange ID appears as part of the URL when you see the word "exchange" in the URL as well. For example, in the URL: `http://s1.amazon.com/exec/varzea/ts/exchange-glance/Y01Y4980982Y9866305/`, Y01Y4980982Y9866305 is the Exchange ID. You can learn more about Exchange ID searches by looking in the kit at `/AmazonWebServices/API Guide/search_exchanges.htm`.

Manufacturer Manufacturer searches help you find a particular product vendor. For example, you might only want to learn about products from Sony. The problem with the manufacturer search is that it doesn't rely on an identifier—it uses a string that contains the manufacturer name. This type of search requires that you know the manufacturer name as Amazon recorded it in their database and assumes Amazon recorded the manufacturer name the same for every product. In many cases, popular manufacturers will require you to perform several searches using the variants of the vendor name. Of course, using a Web service application lets you automate the process instead of entering the information manually in a search form online. You can learn more about manufacturer searches by looking in the kit at `/AmazonWebServices/API Guide/search_manufacturer.htm`.

Marketplace A marketplace search is on the same order as a power search—except that it concentrates on products, instead of books. Like a power search, a marketplace search lets you perform very exacting research on Amazon. Figure 1.3 shows the online version of the marketplace search (`http://s1.amazon.com/exec/varzea/subst/search/fixed-search.html`). As a developer, you can use any of the following parameters as your search criteria: Keyword-Search, Browse-ID, Zipcode, Area-ID, Geo, Rank, and Index. The Area ID parameter identifies the country. Appendix C contains a complete list of Area ID values. You combine the Area ID with a Geo value to specify the location of the shipper or the recipient. The Rank property defines the sort order of the returned values. The Index property determines whether you receive marketplace or ZShop entries. You can learn more about marketplace searches by looking in the kit at `/AmazonWebServices/API Guide/search_marketplace.htm`.

FIGURE 1.3:

Use a marketplace search to provide exacting product search criteria.



Seller Profile You don't have to accept that a vendor has a spotless record. When you find a product you like online, you can use the Seller ID returned by that search to locate seller profile information. This information includes a cumulative rating and seller statistics such as nickname. The profile also includes individual comments and the number of sales the seller lost. Think of the seller profile as a sort of online Better Business Bureau (BBB). You can learn more about seller profile searches by looking in the kit at `/AmazonWebServices/API Guide/search_seller_profiles.htm`.

Similarities Many of the products you see on Amazon have a "Customers who bought this item also bought" or "Customers who shopped for this item also shopped for these items" entry. These entries tell you about similarities between the current product and the list of products that follow the entry. After you find a primary product that interests you, you can use the ASIN from that product to perform a similarities search. Unfortunately, this isn't a smart search. While two books might contain XML in the title, they might not be true replacements. This is one case where Amazon can make recommendations, but you have to exercise caution in make the selection. You can learn more about similarities searches by looking in the kit at `/AmazonWebServices/API Guide/search_similarities.htm`.

Third Party Every third party vendor that works with Amazon has a Seller ID. You obtain this ID by looking for a product sold by that person. The Seller ID appears as part of the URL in the same way that the Exchange ID (explained earlier in this section) does. You can learn more about third party searches by looking in the kit at `/AmazonWebServices/API Guide/search_third_party.htm`.

Universal Product Code (UPC) Just about every product imaginable includes a UPC symbol today. Scanning the symbol enters the UPC in a computer or special scanning device. You can use the UPC to look for items on Amazon. According to the documentation, this option currently works only for music CDs, but testing shows that it does work for other items (albeit, inconsistently). Look for Amazon to allow UPC searches for other product categories in the near future. You can learn more about UPC searches by looking in the kit at `/AmazonWebServices/API Guide/search_upc.htm`.

Locating Services

Amazon provides access to a number of services such as the associates program and its own Amazon VISA card. You can make keyword requests for services that you need using the same techniques you use for finding books or other products. The list of Amazon service categories appears at `http://www.amazon.com/exec/obidos/subst/home/all-stores.html/`. Don't get the idea that services just include the mundane. You can also perform tasks such as looking up your favorite restaurant, reading reviews about it, and downloading the current menu. I find the restaurant service lets me find new places to try with minimal risk.

Acting as an Associate

One of the first uses of the Amazon Web Service, outside of searching for things, is to act as an associate. Today's economy actually makes being an associate a great idea because you can usually get a few dollars each month for very little work. An associate simply directs potential buyers to the appropriate Amazon Web site. When a potential buyer makes a purchase, the associate receives a commission. You can learn more about the associate program at `http://www.amazon.com/gp/browse.html/ref=sd_str_as_dir/?node=3435371`.

Most associates add product entries to their Web site. The Web site likely contains a description of the product and the potential buyer clicks on a link to go to the Amazon Web site to make the purchase. However, associates can use any number of application types to perform the same task. For example, you could create an intranet site for a group or even create a desktop application with an HTML interface. Amazon doesn't require you to provide the links in a specific way.

Depending on your application setup, you must refresh the content periodically to maintain your licensing agreement with Amazon. Normally, this means refreshing the content

every 24 hours if you display just the content. You must refresh the content every hour if you display prices.

Providing Enterprise Services

Maintaining a corporate library, creating purchase lists, or researching product ideas all take considerable time if you rely on catalogs. Even searching for this kind of information online can take a long time. The process is also error prone because you normally have to read someone else's list and use the information to perform the search in some way.

A Web application that relies on Amazon Web Services could automate this process to an extent. Instead of having someone manually retrieve the product information and associated vendor information, the requestor could simply submit an electronic list that the Web application can process automatically. This technique greatly reduces the time requirements for finding products and reduces the number of errors made by the procurement staff.

Performing Research

Only your imagination limits the number of ways to research information using the Amazon Web Service. The "Using Overviews and Details" describes a number of search techniques you can use to find specific products. However, once you find these products, you can perform additional research to discover ways to save money, increase productivity, or reduce risk. For example, you can use a similarities search to locate products that do the same thing

Using Blended Searches to Your Benefit

A blended search lets you look across product boundaries. You perform a blended search by providing search parameters without any mode (product category such as books) or page (the page number of the result set) information. Generally, Amazon discourages blended searches because they return too much information—most of which is incorrect. However, blended searches are valuable in some situations.

One example of a good use for a blended search is a site that offers products in a specific category, rather than a specific type. For example, you might have a Pink Floyd site where you sell everything to do with the music group. A blended search is more likely to provide a complete list of items that you can sell from your site.

Another example of a practical blended search is when you need to perform research and don't know very much about the product. For example, you might be the proud recipient of an Aebelskiver pan. Aside from problems spelling the name if you don't speak Danish, you might have a hard time figuring out uses for the pan. A book search will prove disappointing, but a blended search will show results in both the Home & Garden and the Kitchen & Housewares categories.

as a current product, but for less money. In fact, you might find a used product that someone else doesn't need any longer. Performing a seller profile search lets you reduce risk by ensuring the seller is reputable. Finally, using a Web service to help in the purchasing process will result in a productivity gain.

Research can also include product comparisons. For example, you can create a hot word list and use it to perform comparisons against a list of products downloaded from Amazon. You might rate products based on the number of hot words contained in the list and add or remove points depending on how the vendor uses the hot word. Product research could include a look through the buyer comments too. The appearance of a hot word here could mean that the product provides superior support for that feature or might not include it at all.

Adding a Competitive Advantage

Sellers can also use research to their advantage. For example, you could build a Web service that automatically retrieves the sales rank information for your product and tracks that information over some time frame. The resulting statistics can help you understand how well your product is selling—at least on Amazon. (As Amazon gains popularity, the sales rank number begins to reflect the sales environment for the affected products as a whole.)

Amazon also encourages comment—all kinds of comments. Because the buyer is under no obligation to provide any particular input, you can receive and analyze all kinds of comments. In general, positive comments are what they seem—they tell you that your product satisfied some need better than a competitor's. Many negative comments also include some level of truth; although, you have to consider how much value the comment includes for your particular product.

As with buyers, you can perform keyword searches against your competitor's product to determine how their product compares with yours. The comparison can also reveal general buyer approval of the product and the strength of the competitor's marketing campaign.

Accommodating Personal Needs

Many people recognize the versatility of the Amazon Web site and realize that they can use the information it provides for more than just purchases. The problem is that the Web site often returns more information than they really need or the information isn't in the right format for their needs. For example, one person I recently read about uses the Amazon search site to create a shopping list of items needed at the store. (In this case, the person used the information at a department store when buying gifts.) Unfortunately, the search site doesn't provide information such as size directly, so creating a shopping list is error prone and not very efficient. This person eventually created a Web service that loads the shopping

information into their cellular telephone. With telephone in hand, this person has a new way to go shopping where all of the details for each item are easily accessible.

Obviously, Amazon is hoping that each search of their Web site will bring a sale, but they're also realistic enough to understand that it takes multiple searches to garner the transaction. Consider the shopping list on the cellular telephone again. Suppose this person goes to the store and either can't find the item or learns that the store charges too much for it. Amazon is now in a position to get the sale even though the person originally had no intent of buying the item online. In short, supplying information can result in a sale.

Downloading and Installing the Kit

Before you begin working with the Amazon Web Service, you need to obtain the kit and a developer token. Once you have the kit, you need to install it and become familiar with its content. The following sections describe the kit-related tasks you need to perform.

Performing the Download

Downloading the kit is easy. You'll find the main Web services page at `http://www.amazon.com/webservices/`. Figure 1.4 shows that this page contains some information, along with two important links. Although the steps shown in the figure are numbered, you can perform the first two steps in any order. This chapter assumes that you want to download the Amazon Web Services Kit first.

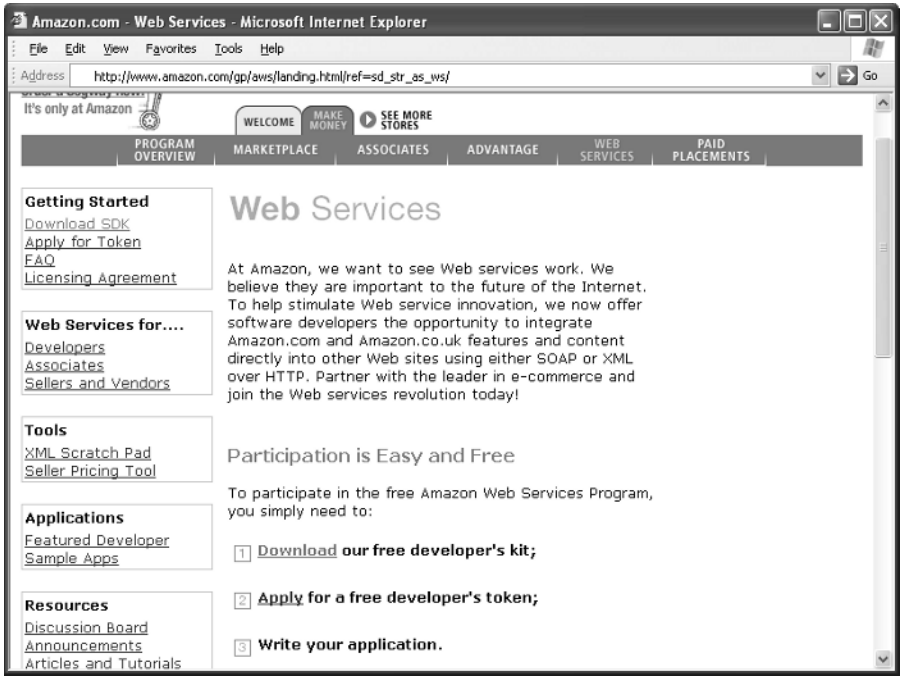
Click the Download Our Free Developer's Kit link and you'll see a page that describes the kit. Click the Download Now link and you'll see a Save File dialog of some type (the precise presentation depends on the browser you use). It always pays to save this file to disk (in an easily located folder), rather than open it immediately, so that you have a permanent copy. In addition, you might want to name the kit something more descriptive than `Kit.ZIP`. I used `Amazon Web Services Kit.ZIP` for my kit and this name might appear in other areas of the book. The file is incredibly small for a developer product—a mere 511KB.

Getting a Developer Token

Once you complete the download process, click the Apply for Amazon Developer's Token link. You'll see a Web page that requests an email address and password. You'll see a complete copy of the licensing agreement below these fields. Make sure you read the licensing agreement because it contains important usage information in addition to the actual terms of use. Don't worry about copying the licensing agreement to disk—the kit includes a copy of the licensing agreement you can use for reference purposes later.

FIGURE 1.4:

The Web Services site helps you obtain the kit and a developer token.



After you read the licensing agreement, check "I have read and accepted the Amazon Web Services terms and conditions." at the bottom of the page. Click Accept Terms and Conditions. (If you aren't an Amazon customer, you'll see additional Web pages requesting personal information. Fill out these Web pages so that Amazon knows who you are.) At this point, you'll see a message stating that Amazon will send your developer token to your email. I received mine in about an hour—you might need to wait more or less time depending on how busy Amazon is at the time. Make sure you save the email message containing the developer token because you'll need it for every transaction later.

Installing the Kit

The kit is actually a Zip file containing examples and documentation. If you're running Windows XP, the operating system provides a program to unpack the file for you. Otherwise, you'll need a special program that reads the compressed file and unpacks it for you such as WinZip (`http://www.winzip.com`).

You won't find any actual developer tools in the Zip file. The file does include complete path information, so you can unpack it in the root folder of your hard drive if you like. I used the D drive on my system, so the Amazon Web Services Kit appears in the `D:\AmazonWebServices` folder.

At this point, the kit is ready for use. However, before you go any further, you need to know about three files in the \AmazonWebServices folder. The AMAZON.COM_LICENSE.TXT file contains a copy of the license agreement that you saw online. Make sure you retain this file so that you can refer back to the usage terms as needed.

The GNU_GENERAL_PUBLIC_LICENSE.TXT file contains the distribution agreement for the Amazon Web Services Kit. These additional licensing terms don't affect how you use the kit. However, these terms do affect how to distribute any applications you create, so make sure you understand the licensing terms. You can learn more about the GNU General Public License at http://www.gnu.org/copyleft/gpl.html and http://www.gnu.org/fsf/fsf.html. This licensing policy is the same one used by other popular software such as Linux.

The READMEFIRST.TXT file contains useful information about the Amazon Web Services Kit and tells you where you can obtain additional information. This third file is very helpful because it contains URLs where you can obtain additional examples. It also has URLs for help sites and additional information. Finally, you'll want to read this file if you want to run the Java examples because it contains instructions for using them.

System Setup Considerations

Once you obtain the Amazon Web Services Kit and a developer token, it's easy to think that you're ready to write your first program. Theoretically, you can do just that. The problem with proceeding at this point though is that you don't know about the viability of your system configuration. For example, if you have a very fast processor and a lot of memory, it's easy to assume the page you've designed will work fine on all systems. However, once you load the resulting application on someone else's machine, it might not work very quickly (if at all).

Defining a usable development setup can save you considerable time and effort later. When you create a great development environment, you ensure that you'll see the application as the user does, which reduces the potential for deadly errors. Because the Amazon Web Services Kit is so accommodating, you'll need to spend a little extra time considering all of the possible usage scenarios. The following sections provide tips you can use to reduce the setup complexity.

Understanding Connectivity Requirements

You must consider three kinds of connectivity when you set up your development system. The first level of connectivity is your own machine. Make sure your machine has a connection to the Internet. Otherwise, any tests you run will fail. Remember that a Web service runs on the remote machine, not your local machine. You're borrowing the resources of that remote machine to perform useful work.

The second level of connectivity is the user's machine. If you create an associate Web site that simply contains links to Amazon's Web site, you can assume the user has a connection, but how fast is that connection? The best Web sites I've seen ask about the user's connection speed. This question allows the application to send the user the level of information that their connection can comfortably support. If you know that most users will rely on a dial-up connection for your Web site, make sure you also use a dial-up connection for testing. This additional step can greatly reduce the chances that you'll make the application too robust. Users who leave your site and don't use your application are users who are probably visiting someone else.

The third level of connectivity is the non-connected mode. You need to consider what happens when the user loses the connection or doesn't have one available. Applications can store static data locally to enable the user to continue using data they have already queried from the Web service. However, you need to observe any refresh requirements and ensure the data retains the same information the user would see online. For example, the local copy of the data must include any required copyright statements or trademarks.

Programming Setups for the Non-Developer

Many of the people reading this book have marginal experience with programming or do it as a hobby. It's true that Web services rely on the resources of the remote machine, but it's also true that the client must perform some work too. If you have a machine that's already marginal—that doesn't run applications well—trying to write a Web service application for it could make matters worse. The local machine must have some resources for using the Web service application.

Depending on the kind of application you create, you'll also need local resources for the programming environment. For example, VBA users have not only the Office application of choice running, but also the VBA development environment. The addition of the VBA development environment can reduce your system performance to a crawl and give you unrealistic performance for your application.

> **▶ NOTE**
>
> Amazon places these requirements on your application. If the user doesn't connect often enough to refresh the data, Amazon doesn't hold you responsible.

It's also possible for you to speed things up too much. If the target platform is a 400MHz Pentium and you're using a 3GHz development machine, your application performance will look nothing like the user's performance in most cases. For a Web site, the machine performance differences might not be quite as significant as when you develop applications that run on the desktop.

Considering the User

Depending on how you use the Web application you build, user needs will take on significant importance. Many applications start out as projects that the developer is creating for personal use. Some of the best applications I've written fall into this category. However, you need to consider the user no matter who will use the application because you're a user too. At one time, I wrote rough applications that I understood but couldn't use efficiently because they were only for test purposes. After I ended up rewriting a number of the applications because I couldn't figure them out or other people asked me for copies, I began writing every program as if it were for someone else.

The applications you write with Amazon Web Services will likely see use from other people, even if you don't know it right now. Consequently, you need to consider what a hypothetical user will need. For example, you might need to include a few special search options. Sure, you could get the same results by typing a little extra text, but adding the functionality directly into your application makes it easier to use (faster in most cases as well).

It's also important to consider users with special needs. The "Addressing Users with Special Needs" section of Chapter 11 contains details on this topic, but you might need to perform setups before you even begin coding. For example, if you work on a Windows machine, you'll probably want to set up the Accessibility features (these features normally appear in the Control Panel and within the `Start\Programs\Accessories\Accessibility` folder).

Using Multiple Test Devices

If your application will appear on the Internet, you need to perform testing using multiple devices. It's no longer safe to assume that only desktop users will have any interest in your application. You might attract some Personal Digital Assistant (PDA) and cellular telephone users as well. This is especially true of a Web application that helps users find a particular kind of product quickly. People often rely on these applications when time is tight and they don't have time to look for a product themselves.

It would be nice if everyone could afford to test every application on every device, but that's not realistic for the developer. Sometimes you need to use an emulator to perform the testing because you don't have the real device handy. Fortunately, you can find a vast array of useful emulators on the Internet—everything from the

▶ NOTE

This book doesn't teach you how to program, so make sure you spend at least a little time learning one of the programming languages discussed in this book before you begin working with the examples. I do provide good descriptions of the applications, but these descriptions won't be enough if you don't understand basic programming concepts.

Pocket PC to cellular telephones of all types. Emulators have limitations, but they do make good test devices in many cases. We'll discuss the advantages and concerns of using emulators in the "Working with Emulators" section of Chapter 9.

Sometimes it also helps to have multiple desktop machine setups. For example, you might need to consider how a Web page looks and acts in Netscape versus Internet Explorer. (Theoretically, you can run both browsers from the same machine, but doing so causes interference problems that some developers find distasteful.) Differences in how the browsers react to specific Web page designs could cause problems in your application. In some cases, you'll need multiple machines to perform this kind of testing. For example, you might need to consider how the application looks on a Macintosh versus a PC if your application has broad enough appeal. Obviously, you can still write Amazon Web Services applications if you don't have a multiple machine setup, but having more than one machine does make development tasks a lot easier and less error prone.

Emulating the Real World

Developers often live in a laboratory. In the laboratory, everyone has the proper equipment, fast machines, and an even faster connection. The user never disconnects unexpectedly and always knows how to get the most out of their computer. The problem with the lab is that it doesn't model the real world. In the real world, users get bored, try odd key combinations just to see what they do, don't understand their computer very well, but do know how to complain about the smallest application problems. If you want to avoid problems with the application you develop, you need to create a development environment that models the real world.

It's also easy to get lost in the development environment setup. Make sure you understand the person who uses your application. For example, it's quite possible that only desktop users will have any interest in your site on desktop machine maintenance, but you need to determine that fact in some way (online surveys work well). You also don't want to spend a lot of time testing the application to meet the needs of users who have no use for your product. Again, surveys and newsgroup polls are helpful in determining the real world environment that you must emulate with your system.

Your Call to Action

If you've read the entire chapter, you know what a Web service is, how Amazon Web Services fits within the general definition of a Web service, and what you can use Amazon Web Services to do. You can use this knowledge to create opportunities to use Amazon as a search

engine for all kinds of tasks, not just as a means to extend the sales potential of your Web site. At this point, you also have a machine that's set up to create an Amazon Web Services application of some sort and you have the Amazon Web Services Kit installed.

The next step of the process is to evaluate where you're going based on the content of this chapter. You need to consider what you want to do with the information Amazon provides, how you plan to present it, your own capabilities, and the capabilities of the person using your application. This may sound like a lot of work, but it's important to create a firm foundation for your application. When you take these preliminary steps, you begin thinking about problems and solutions to those problems.

Chapter 2 builds on the knowledge you gained in this chapter. You learn what to expect from Web services and see examples of how Amazon Web Services could work for your application. Obviously, these examples are just starting points, but they do help you understand what takes place with a Web service and provide you with a little more information on how they work. Chapter 2 helps you understand why Web services are such a great idea and what Amazon Web Services can do for you in particular.