

How ASPs Work

by [Marshall Brain](#)

The Web and the Internet began to really heat up and receive significant media exposure starting around 1994. Initially, the Web started as a great way for academics and researchers to distribute information; but as millions of consumers flocked to the Internet, it began to spawn completely new business models. Three good examples of innovative models include:

- Amazon - Amazon (which opened its doors in July, 1995) houses a database of millions of products that anyone can browse at any time. It would have been impossible to compile a list this large in any medium other than the Web.
- Ebay - Online auctions make it easy and inexpensive for millions of people to buy and sell any imaginable item. It would be impossible to do this at a reasonable cost or in a timely manner with any medium other than the Web.
- Epinions - Thousands of people contribute to a shared library of product reviews. One of the Web's greatest strengths is its worldwide view and collaborative possibilities.

These different business models are all visible to anyone surfing the Web. One of the most interesting behind-the-scenes business models that the Web has created is called the **ASP**, or **Application Service Provider**. ASPs are a completely new way to sell and distribute software and software services. Although ASPs were possible before the advent of the Web, the Web makes them so easy to create that they have proliferated hugely in the last several years.

The ASP model can be extremely appealing to businesses -- especially small businesses and startups -- because it can drastically lower the costs of software and services. In this edition of [HowStuffWorks](#), we will look at the ASP model from top to bottom so you can understand it completely. You will learn how it works, why it evolved and why it is becoming so popular.

Defining an ASP

ASPs tend to be made fairly complex and confusing in the media, but people have been using forms of the ASP for centuries. By looking at one of these existing models and seeing how simple they are, you can gain a great deal of knowledge about Internet ASPs. An **airline** is a classic example of a non-Internet ASP, and is extremely simple to understand. It therefore makes a great starting point into your understanding of ASPs.

Almost all Fortune 1,000 businesses, as well as many small businesses, use [airlines](#) extensively. Many individuals also fly frequently for business and pleasure. Yet the number of businesses and individuals that own their own [airplanes](#) is extremely small. Instead, we rely on airlines to provide travel services to us on a per-use basis.

The main reason for the lack of plane ownership is the extremely high cost of entry. Let's say that you would like to own and operate your own jet. Here are some of the costs involved:

- You have to purchase the jet. Jets cost millions of dollars.
- You have to maintain the jet.
- You have to hire people to staff the jet -- a pilot, for instance, is someone you will need, and pilots are extremely expensive.
- You have to hope that the jet is in the right place at the right time for the people who need it. If not, you need to move the jet around at a high cost in terms of fuel, maintenance, etc.

In almost all cases, these costs are so high that, compared to the cost of individual airline tickets, they make no economic sense. Even the most intrepid traveler who flies 52 weeks out of the year would spend at most \$2,000 per week (\$104,000 per year) on airline travel. That amount of money would not even cover the cost of the [pilot](#), not to mention the cost of the plane, fuel, maintenance, support, etc. involved in owning and operating a private jet. A private jet only makes economic sense in two possible cases:

- You are moving a group of people around frequently and in unison.
- The value of the people flying is so great that it washes out the cost of operating the jet. For example, if you have a group of executives whose value to the company is \$2,000 per hour (for example, a CEO making \$4 million per year), then obviously you want to waste as little of this group's time as possible. You also want these people to be as relaxed as possible so they can work optimally. In cases like that, a private jet may be well worth the money.

These two cases are extremely rare, hence the rarity of private jets. Note also that people who own private jets frequently travel between the United States and Europe on the [Concorde](#). The Concorde is an ASP for high-speed European travel. No company could justify the cost of owning and operating a supersonic jet.

Airlines are classic ASPs because they give you and/or your company a choice. You can own and operate your own jet, or you can charter a jet from an airline when you need one (see, for example, [How NFL Equipment Works](#) to find out how an NFL team uses chartered jet service), or you can pay a very low incremental cost to fly each time you need to travel (and share the cost of owning and operating the jet with hundreds of other passengers on the plane). The "pay a low price each time you use it" versus the "buy the service outright" option is a common feature of Internet ASPs, too.

There are many other ASP-like models that most of us use every day. For example:

- Shipping companies - Instead of maintaining your own distribution network for packages, you pay a low incremental fee to ship a package with the post office, Fedex or UPS. BMW and McDonalds are examples of companies that do so much shipping that they actually own and operate their own truck fleets -- but this are a rarity.
- Telephone companies - It would be extremely difficult for a company to justify the cost of owning and operating its own nationwide [fiber optic](#) network, so we all pay an extremely low incremental cost for each minute of long distance time we use.
- Power companies - It would be possible for each homeowner and business to [generate power](#), but not for 10 cents per kilowatt-hour. Therefore, it makes sense to purchase power from a power company that distributes the high capital cost of a power plant across all of its customers. Some companies -- especially companies that deal in forestry products -- can actually generate their own power affordably because they have a source of free fuel or waste heat from some other process within the company.

There are cases where we do not go the ASP route. For example, a huge number of Americans own and operate their own automobiles instead of using the ASP called "public transportation." Most large businesses can justify the costs of large [copying machines](#), while smaller companies rely on the ASP called Kinkos.

The point of all this is simple -- ASPs are all around us in many different forms. We choose whether or not to use ASPs based on economic factors that are driven largely by our **frequency of use** and the **cost of entry and maintenance**.

Defining an Internet ASP

Even though airlines fit the model for an ASP, we generally do not refer to airlines as ASPs. The terms "ASP" and "Application Service Provider" are applied specifically to companies that provide services via the Internet. In most cases, the term ASP has come to denote companies that supply software applications and/or software-related services over the Internet.

Here are the most common features of an ASP:

- The ASP owns and operates a software application.
- The ASP owns, operates and maintains the servers that run the application. The ASP also employs the people needed to maintain the application.
- The ASP makes the application available to customers everywhere via the Internet, either in a browser or through some sort of "thin client."
- The ASP bills for the application either on a per-use basis or on a monthly/annual fee basis. In many cases, however, the ASP can provide the service for free or will even pay the customer.

Advantages of ASPs

The ASP model has evolved because it offers some significant advantages over traditional approaches. Here are some of the most important advantages:

- Especially for small businesses and startups, the biggest advantage is low cost of entry and, in most cases, an extremely short setup time.
- The pay-as-you-go model is often significantly less expensive for all but the most frequent users of the service.
- The ASP model, as with any outsourcing arrangement, eliminates head count. IT headcount tends to be very expensive and very specialized (like pilots in the airline example), so this is frequently advantageous.
- The ASP model also eliminates specialized IT infrastructure for the application as well as supporting applications. For example, if the application you want to use requires an Oracle or MS-SQL database, you would have to support both the application and the database.
- The ASP model can shift Internet bandwidth to the ASP, who can often provide it at lower cost.

One thing that led to the growth of ASPs is the high cost of specialized software. As the costs grow, it becomes nearly impossible for a small business to afford to purchase the software, so the ASP makes using the software possible.

Another important factor leading to the development of ASPs has been the growing complexity of software and software upgrades. Distributing huge, complex applications to the end user has become extremely expensive from a customer service standpoint, and upgrades make the problem worse. In a large company where there may be thousands of [desktops](#), distributing software (even something as simple as a new release of Microsoft Word) can cost millions of dollars. The ASP model eliminates most of these headaches.

Examples of ASPs

ASPs come in all shapes and sizes. One way to understand ASPs is to look at them from several different angles using real-world examples.

Simple ASPs

If you were to start a small business today, you would probably begin by contacting three or four extremely common and largely unnoticed ASPs:

- **A Web hosting company** - Companies like [Verio](#) and [WebHosting.com](#) provide a classic ASP scenario -- virtual Web hosting. These companies provide hardware, software, bandwidth and people to host Web sites for companies and individuals. Typically, they charge something like \$15 to \$30 per month for the service, and may host hundreds of accounts on a single machine.
- **An e-mail provider** - A Web hosting company usually provides some sort of [e-mail service](#) with your Web hosting account. There are two other alternatives:
 1. Free services such as Hotmail or Yahoo! Mail

2. E-mail server ASPs who run exchange servers, POP servers or IMAP4 servers and distribute them on a monthly-fee basis - For example, a company in the Raleigh area called [Interpath](#) offers a complete e-mail solution at a rate of \$8 per month per account (as of 4/10/2000).

The advantage of the second approach is that the e-mail address uses your company's [domain name](#).

- **A fax provider**
[Efax](#) provides a free fax service that delivers faxes to your e-mail box. This is a classic example of a free ASP.

The huge advantage of using these ASPs is the fact that you don't have to do anything to get started. Five years ago, a small business looking for these services would have needed to:

- Purchase [Internet connectivity](#) and a [router](#)
- Purchase one or more [servers](#) for the Web server software, e-mail software, etc.
- Hire a person to install and administrate the software
- In the case of a [fax machine](#), purchase the fax machine and a separate incoming line for it

Those are tremendous hurdles. Now, all of these services can be ordered and delivered on the same day, and the monthly cost for all three is probably less than \$50 per month. The latest product category to enter the list of start-up ASPs is **eCommerce Storefronts** -- a storefront might cost \$200 to \$400 per month.

The other thing to note is that ASP versions of these services will be significantly better than anything a small business owner can afford to provide. For example:

- In the case of Web hosting, the provider will normally have a huge amount of available bandwidth, and the bandwidth will be redundant at several levels.
- If there is a problem, trained staff on site 24 hours a day will fix it immediately.
- If you need more capacity, it is available with a [phone call](#) and a small adjustment of the monthly fee.
- The ASP will backup the data on a regular basis and is responsible for disaster recovery.

No small business could afford that level of service with a home-grown server infrastructure.

Traditional ASPs

The "traditional" ASP sells a large, expensive application to large enterprises, but also provides a pay-as-you-go model for smaller clients. A typical example might be ad-serving software or auction software for a Web site. For example:

- [Engage](#) offers ad-management software for Web sites. The software can be purchased on a yearly license costing tens of thousands of dollars per year. In addition, the software requires an Oracle database for the software to use. If the Oracle database is already installed and running in-house, then that is no problem, but if not it is a significant hurdle. The alternative is to let Engage manage the software as an ASP and pay Engage a CPM (cost per thousand) price for the service. Unless you are serving millions of ad impressions per month, the ASP model makes tremendous economic sense.
- [DoubleClick](#) (along with many similarly positioned companies) is essentially an ASP that offers advertising software plus an advertising sales force. What is so interesting about this ASP approach is that the ASP actually pays the customer!
- [OpenSite](#) is a leading supplier for auction software. You can purchase its software and

operate it with a database, or access the software using an ASP model.

Nearly any piece of expensive software, including giant applications like SAP, PeopleSoft and Oracle, now comes in an ASP version to allow these companies to reach smaller customers affordably.

Things to Ask a Prospective ASP

ASPs today offer nearly any service a company might need. Many of these services (like e-mail, Web hosting, ad serving, invoicing and bill delivery, payroll, etc.) are mission critical. It is therefore important to make sure that the ASPs you choose will handle your information and relationship in a mission-critical way. Here are a set of questions you should ask any ASP:

- **How do customers access the software?**
Is it through a browser or an application? If it is through a browser, how does the user experience feel?
- **How are customer service issues resolved?**
If you (or employees) have questions and/or problems with the software, what happens? Does the ASP provide training?
- **How secure is the data?**
You want to find out about internal security policies with ASP employees, passwords and access reports to protect your employees, [firewall](#) and other safeguards against external attack, and things like tape backups to handle hardware failures.
- **How secure is the connection between the ASP and the user?**
Data flows between the ASP and the user whenever the user accesses it. Is it secured by [encryption](#), a [VPN](#), proprietary techniques or some other system?
- **How is the application served?**
Is your data on a dedicated machine or a shared machine? Both techniques are common and you often have a choice (with dedicated service being more expensive).
- **How does the ASP handle redundancy?**
If a machine fails or an Internet pipe goes down, what levels of redundancy are in place to keep your servers online?
- **How does the ASP handle hardware/software problems?**
If a [hard disk](#) fails or the application hangs, what are the policies in place around recovery?
- **How does the ASP handle a disaster?**
If the building were to burn down or a [hurricane](#) came through, how would the ASP handle the complete loss of the facility? How long would it be before the ASP restored service?
- **Who owns the data?**
Obviously, the customer should, but this fact should be stated in the contract.
- **How can I get the data out if I choose to select a new ASP two years from now?**
This is a tricky question on more complicated applications, and one that bears some thought for mission-critical applications.
- **How can I move data between existing applications and the ASP?**
For example, if you have a home-grown ledger system and want to move data back and forth to a billing ASP, how would that work? Many ASPs have already thought of this and handle it very well.

There are probably many other questions specific to your situation, especially for advanced applications. If the ASP covers all of these bases well, then it is likely that the ASP can support your business adequately.