Background of Windows Live

The world is addicted to the Internet. We use it for our work, our meetings, our fun, even for our health. Whether you’re a die-hard techie who relies on the information superhighway to run your business and make a living or the avid web surfer Googling your favorite movies and blogging to your heart’s content, the web is the undeniable driving force behind all things tech. Unlike the colder days of the late 1990s when websites were mostly trinkets to hype overpriced startups, today’s Internet is truly personal and interactive. It has become part of the fabric of our daily lives, and nobody knows this better than Microsoft. The reigning Redmond giant, whose past is firmly rooted in client software and operating systems, recognizes this seismic shift to web as an inevitable need for flexibility, creativity, and innovation. To answer these needs, Microsoft gives us Windows Live.

In this first chapter, we will cover the following areas:

- Discuss some of the history of software development and the role Microsoft has played
- Discuss the different ways that the Internet has changed both software development and the business of technology
- Review the basic tenets of Windows Live
- Consider the benefits of the Windows Live business model

Windows Live is many things. On the surface, it is a collection of websites offering interactive online services to people all over the world. They boast rich user experiences, mimicking the more traditional client application, while enabling the user to view and work with their information from anywhere. Rebranded from the previously named MSN services, Windows Live applications are a comprehensive set of user experiences that attempt to bridge the web divide through seamless, Internet-enabled tools and features.

Yet the Microsoft Live initiative is more than just a tight confederation of online services. It is also a new business model for consumers and developers of Microsoft software, offering streamlined branding features for business and direct channels of monetization for Microsoft developers. It’s a reorganized set of service-oriented API, enabling developers to quickly leverage powerful, online...
capabilities and combine those features into highly specialized applications. Live is also the next generation of some of Microsoft’s most popular existing software and services, such as Office and MSN Search. With so many different features and services, it’s no wonder so many people are having difficulty fully understanding just what it is that Microsoft is trying to accomplish. Despite the many different layers that compose Microsoft Live, its main purpose is clear and undeniable: to unite users on the web and provide them with the information they need.

The Microsoft of Old

In order to fully understand the significance of Windows Live, it is useful to reacquaint yourself with Microsoft’s history of software development. Over the last 20 years or more, Microsoft has established itself as one of the world leaders of client-driven software providers. Client software can be thought of as any application that installs locally on a user’s computer and that operates almost entirely as an isolated, stand-alone unit. Two of the most obvious examples of popular client applications are industry top sellers Microsoft Word and Microsoft Excel. Both Word and Excel are purchasable as CD-ROM or DVD packages that install on one or a limited number of user workstations. The tools they provide are powerful, leveraging the entire set of features and capabilities available in a basic Intel computer. This software model was long thought of as the main channel of software sales and indicative of the only real way to make money selling computer software. As more and more consumers over the years bought cheaper, more powerful PCs, the need for more powerful software grew steadily. Other popular client applications such as Microsoft Mappoint, Microsoft Access Database, and Microsoft Outlook Mail began to proliferate on the market, giving users access to increasing amounts of information and giving them commanding tools to automate more of their lives.

However, tools like Word and Excel have their obvious limitations. If a user is away from their PC, they are also away from their documents and financial spreadsheets. As copies of documents are made, maintaining the correct version becomes increasingly difficult. For programs such as Microsoft Mappoint, which provides the user with large amounts of data subject to unpredictable change, newer versions of the application code have to be redistributed and directly installed on the user’s PC. If their programs aren’t updated, they inevitably became outdated, providing users with old information, often without their knowledge. Yet these limitations were once widely accepted as a natural part of personal computing.

Along Came the Web

This all began to change with the growth of the World Wide Web. The history of the web, as well as other Internet-driven platforms and programs is its own wild, wily tale, fraught with conspiracy, browser wars, piracy, romance, and tragic heroes (pardon the drama), and falls outside of the scope of this book. It is important to note, nevertheless, that the meteoric rise of the web forced the Redmond giant to rethink its business model entirely. Despite the web slow down in the early part of the new millennium, it was evident to all that the Internet was something much more than a fleeting phenomenon.

As more home PCs connected online, the need for readily available services and ubiquitous information began to grow. Companies such as Yahoo! and Google realized massive growth in online traffic and advertising sales by offering information portal tools and simplified search capabilities. Powerful firms
such as America Online dominated the Internet service provider (ISP) market scooping up tens of millions of dollars in revenue and offering alternative tools to those otherwise bundled in the Windows operating system. Innovative sales and auctioning services such as eBay empowered small home businesses to sell items to people from virtually anywhere on the planet, shaping new Internet-based markets.

The world had suddenly become a much smaller place. While there may always be a place for packaged goods and client-installed applications, the spectacular growth and popularity of the Internet quickly developed into something more personal. More and more home users have begun to use online services as an integral part of their daily activity. The centralization of data and our ability to access information from any Internet-enabled PC has moved out of the realm of “cool” and rooted itself deeply into the realm of the expected.

Of course, Microsoft couldn’t ignore the Internet. It, like most other large software manufacturers, wanted to leverage growing online trends in lots of ways. Some of these early approaches were somewhat ineffective (does anyone remember Active Documents?!!), amounting to little more than building Internet-capable or browser-capable features into already popular platform tools. That wasn’t enough to keep up with the competition. Noting the undeniable popularity of its portal and search engine competitors, Microsoft finally created a series of powerful Internet services called the Microsoft Network and branded as MSN.

The MSN ERA

MSN was Microsoft’s first strong foray into the world of online services. Originally released in the Mid-third-quarter of 1995 to coincide with the release of Windows 95, it has since markedly expanded its services. Although the initial release offered a number of different online tools, most people came to quickly know MSN by its search services, email services, and instant messaging tool. MSN search has enjoyed a long, steady growth in use over time. At first, MSN search was yet another search engine in a sea of otherwise waning websites, such as Lycos and Excite.com. However Microsoft isn’t the world’s biggest software giant for nothing. Revamping their search algorithm and carefully integrating free email services, MSN’s usage began to take hold. Following the form of some of their competitors, MSN began to beef up its site to be more revisit-friendly. Daily news, stock quotes, horoscopes, and local weather forecasts peppered the user’s home page and encouraged people to visit the site for reasons other than simple searches. MSN has also become a well-established Internet service provider, with over 30 million paying subscribers worldwide. By partnering with popular PC manufactures and software distribution services, MSN quickly gained ISP market share, parlaying that advantage into even more MSN service users than ever before. Figure 1.1 shows a typical day on an MSN user’s home page, complete with articles, links to mail accounts, and other online tools, with the telltale search box at the top of the page.

AS MSN’s popularity grew so did the services it offered. Its Hotmail webmail service was one of the first free online tools offered by MSN to enjoy swiftly growing popularity. Hotmail allowed people to access their email from a web browser, freeing them from corporate firewalls and from the often cumbersome and limiting mail services that came with ISPs. Hotmail eventually became a strong unifying force within MSN’s multiservice framework and inevitably became an integral part of Microsoft’s first stab at universal online data access initiative, MSN Passport.
MSN Passport is Microsoft’s single sign-in service created to allow users to log in to multiple websites at one time with a unique ID. A Passport account is required to fully utilize other MSN services such as MSN Encarta, Microsoft’s online encyclopedia; MSN Shopping, a shopping tool affiliated with many popular online retailers; and MSN Spaces, an online blogging tool. MSN Services and MSN Passport continued to grow on their own merits for sometime, yet the original intent of these tools were never fully realized. In the early part of 2001, Microsoft unveiled the Hailstorm project. Hailstorm was intended to be a series of application building blocks that leveraged the MSN Passport model to offer subscription-based online services to their users. Users could enter their personal data one time into their Passport account, including their credit card numbers and personal demographic data, and access that data from any number of websites in order to create profiles, make purchases, receive orders, and so on. Despite Microsoft’s best efforts and diligent marketing campaigns, Hailstorm was ultimately dropped. The reasons for Hailstorm’s inevitable failure are highly debatable, though most experts agree that consumers simply didn’t trust an online service to safely store their most personal data. Thus, while MSN Services and its supporting technologies remained useful and continued to attract users, their intended purposes were never fully realized. MSN remains a strong online service competitor and powerful search engine.
The Rise of Interactive Services

While Microsoft was struggling to turn MSN Services into a comprehensive blend of subscription services, the competition was getting stronger. Newcomers to the web market were creating new online services for work and home that replaced otherwise traditional types of software. Salesforce.com, a provider of online customer relationship management (CRM) tools gained popularity among small and medium-sized firms looking to keep their sales personnel’s data available on the go. Their web-based model allowed members of company sales personnel to access up-to-date information from any computer, using only the web browser as a single vehicle of data delivery. Weblogs (“blogs”), an online, publicly viewed discussion forum where a person can enter opinions and thoughts in a diary format became a fast-growing phenomenon. People began flocking to other people’s blog pages to read their entries, post feedback, and connect with other people with similar interests. Figure 1.2 shows the home page of Blogger.com, one of the more popular free blogging websites.

![Blogger.com Home Page](image)

Figure 1-2

Perhaps Microsoft’s toughest competitor, Google Inc., was growing its relatively limited services at a frenzied pace. In addition to a unique search algorithm that trumped that of other search engines, Google’s pay-per-click advertising model won it both popularity and huge earnings. Expanding past the confines of web searches, Google began to release ancillary services, many of which have met with great
success. Some of these services have targeted popular MSN services, adding valuable features free of charge. GMail, for example, Google’s answer to webmail, competes directly with MSN’s Hotmail and offers integrated web chatting and unlimited email storage. Many of Google’s newer services also boast rich user experiences, while remaining entirely web-based. Google Maps competes with the extremely popular Mapquest web service to provide online driving directions. Although Mapquest has been integrated into many third-party websites as a paid mapping service, Google Maps offers a point-and-drag feature that allows users to move about the map without constantly reposting the page for more map data. Google Maps also combines satellite imaging, so users can get a zoomable bird’s-eye view of their destination. As these new features each met with positive fanfare, Google began to shift its focus away from web searches and more towards interactive services. Bold new initiatives were instated to move users off their desktops and into the web for more personal, tangible needs, while downloadable Google tools began to infiltrate the mostly homogenous Windows desktop. Google Desktop search, for example, applied Google’s popular, effective search algorithm to items on a user’s computer, bypassing native Windows search tools and bridging the divide between local desktop applications and directly accessible online tools. The popularity of Google’s new services began to change the way people perceived their software experiences.

The trend towards interactive services really began to peak with the advent of user content-driven websites. Often referred to as “Web 2.0,” this trend of highly interactive sites got people to stop thinking of the web as a tool for their lives, and to think of it more as a place where they live. Community-based websites like Friendster, LinkedIn (displayed in Figure 1.3), and the ever-controversial MySpace.com swept the young adult demographics and became the driving force behind a whole new generation of online users. Whole social communities began to build themselves using simple person-to-person networking tools built right into the site. Musicians, comedians, and actors took the initiative and began to market themselves heavily through these channels, often creating true celebrity for themselves as the result. Personal posting websites like Craigslist.org moved from the realm of the conveniently local to the coveted tech stardom of a top traffic-producing website. Finally, the Internet came to be regarded as a destination for users and client bases, and not simply as a means of transport.

The rise of so many interactive services presented a new challenge to Microsoft. Not only was their ever-popular client software model beginning to fade into the sunset, but the tools on which much of their software was built no longer fulfilled the needs of the industry. Developers began to reconsider their programming preferences, switching to languages and development kits that could build functional code on all different kinds of platforms. Although ASP and its more powerful successor ASP.Net were fine platforms for website server processing, there were no client-driven tools or services provided by Microsoft to enhance the user experience. Website developers started to build on existing services and libraries, combining them into their own applications to create “mashups.” Mashups were interesting combinations of data feeds from disparate online sources. Yet they lacked the uniformity of a single development source. For example, web developers would combine an RSS feed from public postings sites such as Craigslist.org with the mapping features provided by Google. What the development community needed was a fresh new set of tools that were expedient for developing a mashup and provided them with the ability to easily integrate services in a predictable, controllable fashion. Likewise, Microsoft wanted to provide a new set of development tools that would capture the web developer’s interests. To satisfy both of these needs, Microsoft released Windows Live services.
So What Is Windows Live?

Windows Live is Microsoft’s answer to the demand for online services. It is a collection of newly developed websites and recently revamped tools that can be accessed via a web browser, accessing data on a remote, centralized server. Currently, Live is composed of many different sites, and the number of sites is growing each month. In its core services, Live is a rebranding of popular existing Microsoft Internet services, such as MSN search, MSN Messenger, and MSN Spaces. However, Live is also a fresh new batch of potent websites that unshackles the user from their PC, letting people access and save information from just about anywhere. Websites such as Windows Virtual Earth are commanding replacements of client software, such as Microsoft Mappoint, that implement client-like user experiences right through the web browser.

New Programming API

At the time of this writing, Windows Live is composed of roughly 8–10 official website platforms, each providing its own application programming interface (API). As Windows Live grows, there are countless more services to come, though not all of them are necessarily suited for mashup-type development. Figure 1.4 shows a list of all of the different development SDKs that now fall under the guise of a
Chapter 1: Background of Windows Live

Windows Live service. Many of these API are older development kits, renamed and subsumed within the Windows Live brand. APIs for existing services such as MSN Search, MSN Messenger, and MSN Spaces have remained relatively static during the unveiling of Live. Developers who have used these API in the past will find that consuming their Live counterparts will ultimately require only minor changes to nomenclature. The purpose of this, of course, is to ease the transition of current MSN developers into the Live programming platform, preventing older programs from becoming obsolete.

New services offered through Live.com offer a series of programmatic extensions that are more flexible and simpler to use than previous Microsoft SDKs. These newer API are language and platform independent, enabling web developers from just about any environment to build applications leveraging Windows Live services. These APIs are exposed mostly as public web methods and browser client code and live almost exclusively within public Jscript or JavaScript libraries and Soap-based web service definitions. Using these flexible, public libraries, developers have an opportunity to combine their existing applications with Live tools or build Live-powered sites from the ground up. At the time this book is being written, there are eight official Windows Live platform SDKs:

- Windows Live Search
- Virtual Earth
Chapter 1: Background of Windows Live

- Windows Live ID
- Windows Live Expo
- Windows Live Spaces
- Windows Live Messenger
- Windows Live Custom Domains
- Windows Live Gadgets

Each of these platforms provides its own public API, allowing users to integrate and leverage Windows Live services in their own sites with ease. We will be exploring each of these platforms in this book, taking time to understand their core API and apply them to tangible, real-world applications.

Principles of Windows Live

Understanding Windows Live means more than just knowing how to create a Live blog or showing map pushpins in a Virtual Earth application. Windows Live was created to provide support for both the developmental and business aspects of software development. As a Live programmer, you should be aware of the core principles of the Windows Live services and how they affect consumers and developers alike.

Make It Easy for the Developer Community to Integrate with Windows Live and Make Money

The Windows Live model provides various avenues of business development to help software programmers and web developers optimize their revenue streams. This business model is an important aspect of Microsoft’s new development model. In the past, Microsoft provided no such business-enhancing services. The ability to leverage services and build revenue streams sets Microsoft apart from other similar online service providers. The combination of these Windows Live services, the broad MSN/Windows Live user base, rapid application development tools, and revenue-building options create unique opportunities for developers seeking to build their business value through one of four business scenarios:

1. Direct Monetization

   The primary means of earning revenue using Windows Live services is through Microsoft’s contextual advertising program branded as Microsoft adCenter. Note that at the time of this writing, adCenter has not been fully implemented. Microsoft adCenter is a pay-per-click advertising model that enables website publishers and developers using Live services to earn money directly through user clicks. This idea is not necessarily new, mind you, as other successful competing services such as Google’s AdSense and Yahoo! Search Marketing (formerly Overture Inc.) have made profits that rival the GNP of third world nations using the pay-per-click model. However the Microsoft version has far more depth than the aforementioned competitors. Rather than using simple context-based patterns to provide targeted advertisements, Microsoft adCenter will also be using demographic data provided and stored through Windows Live services to narrow advertisements to an extremely personalized, laser-precision targeted advertisements. The combination of Microsoft advertisements with your Live-powered website works directly to your advantage.
2. Customer Relationship Management (CRM)

Windows Live services helps you perform ongoing relationship management with existing customers. That enables increased frequency of activity or revenue, longer retention as an active customer, or decreased support costs. Generally speaking, CRM can be quite a hassle. The overhead of maintaining data integrity, providing tools for simple upkeep, and making data available to peoples or programs in different parts of your company can be staggering. Using Windows Live services, much of the CRM burden no longer sits with you, the business owner. The acquisition of personal data and the sharing of that data across applications is one of the core features of the Windows Live initiative.

3. Customer Acquisition

Windows Live services aren’t just great for maintaining customer data; they’re ideal for finding new customers. As more people begin to share their data across applications, leveraging Windows live features, developers can use this sharing of data to turn people into loyal clients.

4. Brand Awareness

Although no direct monetization is available through this scenario, developers can leverage the Windows Live service to facilitate brand awareness through a broad contingent of end users.

Easy to Develop

In order to attract the widest cross-section of web developers, Windows Live services have been made available in a manner that facilitates simple, easy-to-use development patterns. Programmers developing for any platform and in just about any language can consume Live services, integrating them directly into their existing applications with minimal effort. The result is a set of powerful tools that allows programmers to concentrate on creating sites that work for their business.

Simple HTTP-based Protocols

Just about all of the major method calls available from Windows Live services can be invoked using simple HTTP-based calls over the Internet. Using XML-based formats such as the Simple Object Access Protocol (SOAP), developers can invoke methods from any web-enabled application. These calls are readily available to developers using any language. Thus, Java developers who want to incorporate specific search features using Windows Live Search (formerly MSN Search) can do so without the cumbersome overhead of bridging their non-Windows code with Microsoft’s API. Just about every public method that creates use accounts, modifies existing data, or manipulates features within an integratable Live service can be called using SOAP.

Public JavaScript/Jscript Libraries

HTTP-based calls are fine for retrieving and saving data, but what about the user interface (or “UI”) itself? Does Windows Live offer features that deliver rich content right into a running application? You bet it does. Using publicly available Jscript libraries, users can integrate fantastic client UI right into their own applications with minimal overhead. This is especially useful when using graphic-intensive Windows Live services such as Live Virtual Earth. Virtual Earth supplies participating websites with all of the graphics, methods, and code required to inject interactive maps directly into web pages. Developers need only to reference the correct Virtual Earth libraries, define the size and location of their map layer, and call Jscript methods as if they wrote them locally.
Chapter 1: Background of Windows Live

In fact, Microsoft made sure that these client libraries were flexible enough to also allow for plenty of customization. Let’s say that you are a web developer with an online database of local restaurants. It would be great to add interactive mapping so that your users could easily get directions to their favorite eateries. Well, not only can you add these features with only a few lines of code, but you can also create customized thumbnails for each type of restaurant, and change the way the directions are presented, all using a fraction of the code otherwise required without Windows Live Jscript libraries.

**Straightforward Licensing and Accommodating Terms of Use**

Windows Live is an initiative based on integration, and the licensing shows it. Unlike other competing platforms, Microsoft allows developers to integrate and leverage Windows Live services according to the following guidelines (taken directly from the dev.live.com website).

- Obey the law.
- Obey any codes of conduct or other notices we provide.
- Obey the Microsoft Anti-spam Policy, which is available at [http://privacy.msn.com/anti-spam](http://privacy.msn.com/anti-spam).
- Keep your service account password secret.
- Promptly notify us if you learn of a security breach related to the service.

Users should take the time to read through the contractual agreement to be sure that they are not in violation of any spamming or indecent conduct as defined by Microsoft. However, most developers will certainly appreciate that implementing the Windows Live services API can be done with only nominal degree of legalese.

**Any Platform, Any Language**

Windows Live is about interaction and cooperation. The tools provided, method calls, APIs, and SDKs available to a Live developer are without the encumbrances of platform-specific technologies. Any programmer can use Live services. Although the manner of implementation might vary depending on the types of applications used, the delivery of data by Live websites is done using industry-standard, open protocols and patterns. No need for ActiveX, plug-ins, specialized runtimes, or downloadable libraries—Windows Live works in your web browser or HTTP-supporting client application. . . that’s it!

**Easy to Use**

Your users will find the new Windows Live–powered features of your site to be not only powerful but simple to use as well. Live tools such as interactive maps and gadgets make accessing data easy for everyone, no matter what type of computer you use.

**Rich User Experience**

Windows Live services provides the next generation of interactive web applications. Unlike the web apps of old, these new sites embrace powerful client-based methodologies to facilitate a much richer end user experience. By using web client patterns such as AJAX (Asynchronous JavaScript and XML) and Microsoft’s newly released ASP.Net AJAX framework (formerly known as ATLAS), the web app has moved beyond boring, static HTML. Highly responsive images, tags, and layers make the need for constant page reposts almost completely obsolete. As a result, applications using Windows Live services enjoy the combined convenience of responsive client-like functionality and a portable web architecture.
Chapter 1: Background of Windows Live

Although the objects used in AJAX applications are in no way new to web programming, their implementation within a flexible, reusable framework is. Much of the ASP.Net AJAX model falls outside of the scope of this book. For more information about ASP.Net AJAX patterns, readers should visit the official website shown in Figure 1.5 at http://ajax.asp.net.

Figure 1-5

Accessible, Up-To-Date Information

Perhaps one of the most compelling reasons to integrate with Windows Live is that your users will have access to information at the speed of the web. Windows Live delivers content in real-time, providing website users with the most accurate, up-to-date information available. Website designers and administrators don’t have to update their files or add data to their databases—Windows Live does all the footwork. Services such as Live Search, Virtual Earth, and Live Expo connect your users with vast amounts of online resources such as maps, satellite imagery, user postings, website searches, addresses, phone numbers, and more. With each service you add to your application, the less updating and maintenance you need to perform.
Why Should I use Windows Live?

Now that we’re past the part where I tow the company line, let’s discuss why Windows live works for you, the developer. At its core, Live services is simply another SDK. It is a collection of advanced APIs and services that you may choose to use or to ignore. If you are a traditionalist, deeply rooted in monolithic systems, you may not be particularly turned onto the Live methodology. Likewise if you are a staunch anti-Microsoft developer, you may choose to cast Windows Live aside, telling yourself it is a useless whim on the part of the Mother Ship (I have always loved calling Microsoft that) to compete with similar market-absorbing services. Yet before passing judgment, there are certain undeniable advantages to the Live platform you should consider.

The first thing one needs to understand is that the Live methodology is here to stay. The notion of leveraging other online services and incorporating them into your own is neither a passing whim nor a contrivance of the Microsoft development team. Rather, it is the natural progression of software development in an increasingly connected web world. The services offered by Windows, such as Live Virtual Earth, Live Spaces, Live Search, and even Live Expo are commonly used services in websites and applications of all kinds. Offering these features communally to developers of all technological backgrounds gives everyone the power to create software more quickly. Also, consuming Microsoft’s services in a manner loosely connected to your own software places the onus of maintenance and upgrading almost completely on Microsoft. That means that you, the developer, can spend more time concentrating on providing software that directly benefits your business. So, even if you despise the idea of combining your software with the services of other websites, you might as well get used to it, because it’s here to stay.

Now if you’re like me, you welcome the idea of third-party service integration, but you might be skeptical of using Microsoft’s services. Let’s be frank, shall we? Windows Live isn’t the only provider of consumable online features. Companies like Google and Yahoo! have also released public APIs for mapping, blogging, and the like. So why use Microsoft? Why Windows Live? There are plenty of reasons. To begin with, you have the services of at least 10 different platforms at your fingertips. That means that integrating not just one, but many, online services into your websites will be easier using Windows Live. Take, for instance, a website that provides information for people shopping for new and used boats. The website owner would like to list classified ads for various locations and maybe even provide an interactive map so that buyers can easily find their way to sellers. The site developer could certainly elect to grab RSS feeds from one site and implement mapping tools from another. However, wouldn’t it be easier to use Windows Live Expo for the listings and Virtual Earth for the maps? The APIs are all maintained through Windows Live services, and the implementation patterns are far simpler to use. In fact, the site owner might even decide to take their features a few steps further, providing some blogging capabilities for his/her clients and maybe even a cool chat tool where people can exchange boat pictures. Using Windows Live, a developer can implement all of these features without referring to multiple third-party resources.

Another compelling reason to choose Windows Live services is familiarity. Most of us, have worked with Microsoft development tools at one point or another. If you’re reading this book, chances are that you’ve worked with Microsoft languages and platforms as well, such as Visual Basic, C#, and ASP.Net. Although Windows Live APIs are in no way platform-specific, Microsoft developers will find that extending many offered features is quite simple using Microsoft patterns. That means that you’ll be able to create powerful applications with greater ease. Live Gadgets are a good example of this advantage. Gadgets can be written for any platform that supports Jscript or JavaScript. Many gadget developers will no doubt want to employ JavaScript AJAX patterns. People familiar with ASP.Net development will find that Microsoft’s ASP.Net AJAX project provides an ideal client/server toolset for AJAX development. ASP.Net AJAX not only simplifies the otherwise cumbersome client coding required for rich, responsive UI, but it also integrates nicely with the Live Gadget model. Using Windows Live, seasoned Microsoft developers will find that their skills can be a definitive advantage when leveraging features and services.
Perhaps the most compelling reason to develop with Windows Live tools is data. Windows Live is a comprehensive initiative, providing tools and features for many different types of applications. At its core lives a data engine that can collect demographic information in an opt-in model, allowing users to access and reuse their data in situations that suit them. This might not seem all that innovative on the surface, but consider the implementation. Perhaps you have created an application that uses Windows Live ID as your web domain gateway. Users registering with your application provide various private pieces of demographics, such as their home address, their age, their marital status, and favorite websites. Over time you decide that you’d like to expand your services to include specialized searches using Windows Live Search based on geographic locale and preferences. Since the user’s information is centralized, you can request users to share some of their data with you application, explaining the value-added features their data will expose. Using the data already stored using Windows Live ID services, your users can elect to make that information available, benefiting from your new search tools each time they authenticate through your website. The users’ data is available only upon their permission, giving them full control over their personal information. Once they elect to share their data with your site, their experience is enhanced and repeat traffic is born! Now add in services for instant messaging, blogging, and chatting and you get something that other online services can’t easily provide: synergy. That’s what Windows Live does best for you. By leveraging features that help capture and consume your user’s information, you can expand your site to cater to each user’s individual needs.

**The Windows Live Paradigm**

The full embracing of Windows Live services requires a change in the way you think of your website. Up until now the lion’s share of a web developer’s experience centered around building a stand-alone website. The tools implemented benefited only the site’s owner, and any proceeds earned fed a business model completely disjoint from other, possible similar web based businesses. This seems practical for the most part. No need to share services, traffic, or even customer information with competitors, right? Wrong. Not only is this line of thinking counterproductive, but it doesn’t necessarily follow more traditional lines of retail marketing. If every store in the world followed the web business model, we’d inevitably be saddled with hundreds of thousands of retail outlets scattered all over the world in an entirely inconvenient manner. Yet we know this is not the case. Quite often in the retail world groups of stores are strung together to optimize the consumer shopping experience. Strip malls give you five or ten small shops conveniently located in one complex. Although some of these stores might compete on some level for consumers’ attention, they also benefit from one another’s traffic. A mother who goes to a strip mall to pick up a movie at the local DVD rental shop might also decide to grab a slice of pizza from the Italian restaurant two doors over. She may then also remember that she’s out of milk and quickly swing into the convenience store at the end of the parking lot to pick some up. Although this person might have gone out another time to make her purchases, the convenience of having all shops in one location saved her time and effort, and ultimately lead to purchases for three of the stores in the mart.

Shopping malls provide an even stronger example of this retail-sharing model. When a store or shop opens inside a shopping mall, they are not only benefiting from a large stream of localized traffic, but they are also able to share feature and services. A shopping mall offers features that most retail shops would need to provide to their own customer base, such as public bathrooms, handicap accessibility, daytime and nighttime security, basic utilities, ample parking, and much more. Provided that the cost of running a shop in a particular shopping mall isn’t exorbitant, the benefits are obvious. In fact, it stands to reason that unless you’ve got a well-known product, a great location, and a flush bank account for some serious marketing, opening a stand-alone shop can be a more difficult way to start a business.
The same idea of retail sharing can be applied to the web world. In some cases, we’ve already seen this pattern. Portal websites that connect the user to other like-purposed sites is a simplified way of sharing traffic. Rather than having to surf with search engines for hours on end to find websites offering appliance repair, a portal solution would be more direct. Users could come to one website that lists many appliance repair websites, rates their customer feedback, and offers direct links to each if a user wants to surf further. Portals help websites of a similar business type or service share traffic with one another. Portals are often limited in scope, however. Listing a website within the portal might increase traffic, yet the person who comes to a portal for appliance repair will buy only that. There is no chance for cross-selling a slice of pizza or a carton of milk as discussed in the strip mall example. Since the grouping of links to other websites doesn’t really provide the same level of convenience as a physical grouping of retail shops, most portals keep their themes homogenous. As a result, new websites interested in getting new business pursue other options such as search engine optimization (SEO). SEO can be an effective way to get new business; however, that practice is more about direct site marketing, and is more akin to taking out big ads in the yellow pages than it is to sharing traffic implicitly.

The Virtual Smart Mall

Windows aims to rectify web business model by providing many of the same utilities and services available to non web-based businesses. Using the shopping mall analogy as a vehicle for comparison, there are three levels of services that Windows Live offers that can help web businesses. These three levels of service offer a kind of “smart mall” model. The small mall model mimics the conveniences of a regular shopping mall described in the proceeding section, while also adding services around client demographics. Imagine a shopping mall that not only provides you with utilities, bathrooms, and 24-hour security but also with a service that helps you figure out who your clients are and how to best cater to their needs and trends. The result would be a series of stores that shared traffic, shared services, and implicitly helped one another to optimize their marketing, while providing a high degree of convenience to the customer base.

The first level of services is shared utilities. These are the many different tools and features offered through the Windows Live API that can be added to your own website. Live tools, such as Virtual Earth, which allows you to add interactive maps to your application, and Live Search, a comprehensive searching and indexing tool that works directly within your web domain, can be valuable additions to your site. They embody commonly used features that add value to your application but that you’d otherwise have to build and maintain yourself.

The second level of service offered to your website by Windows Live is shared data. Windows Live services builds on a long history of MSN services and leverages from that quite a bit of user data. This data centers around user accounts and comprises demographic information, such as a user’s name, email address, contact list, and other profile-specific data collected from applications such as MSN Messenger. Microsoft has rolled these profiles, along with their aggregate data, into Windows Live services. The result is the ability to create an account using Live Messenger and opt to have this data available to other applications using Windows Live services. What’s more, the amount of personal information available from an account is entirely up to the user. They are in full control of their own information, and may choose to share certain pieces of it in ways that may benefit their website experiences. Newer Live services such as Live Contacts and Live Data are available for precisely this purpose. Again, the amount of data available is entirely up to the user. Clients need to opt in to data-sharing relationships before they can be seen by other users in Live-powered websites. Once they do, they can conveniently reuse much of their information in different websites.
The third level of service offered by Windows Live is shared demographic and trend-based data. Although an individual Live user is completely anonymous to the Windows Live reporting engine, his or her demographic information is tracked and built into a series of trend-producing data stores that ultimately allow businesses to determine their core demographic and to provide services to those users directly. Here is where the virtual smart mall model really comes to life. Online services and advertising solutions such as the previously mentioned Microsoft adCenter and Microsoft adLab offer targeted advertising postings and demographic targeting reports to help website administrators target their business effectively. These tools use the data generated from statistics related to Windows Live traffic and trends. So, as more people use Live-powered websites, there is more information available to help you market your site or services. Imagine a shopping mall that remembers what people buy, what they browse, and where they like to go, and makes that anonymous information available to shop owners. Consumer trends could be discerned and needs quickly fulfilled. The result is a more efficient mall market where people always find things they need and shop owners can consistently determine and sell precisely what their clients desire.

In addition to the growing online community connected through Live ID and rich service functionality, this business model is the driving force behind Windows Live. Participating websites get the benefit of free upfront features and services that drive traffic to the engines that ultimately improve the way they run and market their services.

**Summary**

In this chapter, we took a step back and reviewed the history of software. We discussed the difference between the client software development models of old and the newer ubiquitous online development patterns and how they shaped the online services market. We then took an in-depth look at Windows Live itself and the benefits it offers, discussing some of the similarities that online business has to retail industries and considering the different ways that Windows Live aims to improve upon website marketing. In the next chapter we will explore the various technologies that drive Windows Live, and discuss the ways Live services can facilitate building mashup web applications.