Volume 2
Issue
5

UNDERSTANDING SHAREPOINT JOURNAL

Bjørn Christoffer Aulie Thorsmælhum Furuknap

Beginning SharePoint 2010 Development

UNDERSTANDING SHAREPOINT JOURNAL

Beginning SharePoint 2010 Development

I dedicate this issue to my lovely wife, Lena.

USPJA Publishing LLC
831 Beacon Street
Suite 118
Newton Centre, MA 02459
United States
Phone +1 877 90 USPJA

http://www.uspjournal.com/
journal@understandingsharepoint.com

Credits

About the Author

Bjørn Christoffer Thorsmæhlum Furuknap is a senior solutions architect, published author of *Building the SharePoint User Experience*, speaker, and passionate SharePointaholic. He has been doing software development professionally since 1993 for small companies as well as multinational corporations. He has also been a teacher at a college-level school, teaching programming and development to aspiring students, a job that inspired him to begin teaching what he has learned and learns every day.

About *Understanding SharePoint Journal*

Understanding SharePoint Journal is a periodical published by UnderstandingSharePoint.com. The journal covers few topics in each issue, focusing to teach a deeper understanding of each topic while showing how to use SharePoint in real-life scenarios.

You can read more about *USP Journal*, as well as get other issues and sign up for regular updates, discounts, and previews of upcoming issues, at http://www.uspjournal.com/.

Other Credits

A great big thanks to Kim Wimpsett for doing the copyedit. The quality of work in this issue is greatly attributed to her skill.

Table of Contents

Additional Content

Included in this issue are multiple additional pieces of content. However, some online stores such as Amazon do not offer the option of providing additional content.

If you have purchased this issue and have not received the additional content, please email journal@understandingsharepoint.com, and we can generate a download link for you.

The following should be included in this issue:

- Beginning SharePoint Development, 1st edition,
- All videos from 1st edition (SharePoint 2007)
- Videos from this edition (SharePoint 2010)
- Report: What's New for SharePoint 2010 Developers
- USPJ Academy Lecture: Building the Perfect SharePoint Development Team

If any of this content is not available in your download, feel free to email us as well.

Introduction

Asking the right question at the right time to the right people can trigger a landslide.

Welcome, all aspiring SharePoint developers, to *Understanding SharePoint Journal*. If you are completely new to SharePoint, welcome also to SharePoint and a world of thrills and joys, horrors and quirks. You are about to embark on an exciting journey, and I'll modestly say that you have chosen an excellent place to start, reading this issue as you are.

I may be biased.

Throughout this issue, I will introduce you to SharePoint and some of the tasks you may encounter as a SharePoint developer. If you have done some development already, then that is a big benefit, and the benefit becomes huge if you have done ASP.NET development. However, you should be able to complete the exercises even without prior programming experience.

This issue came about after a member of the mailing list complained about the lack of beginner courses in SharePoint development. According to the reader, most online resources or books assumed you either had a lot of prior experience or were too shallow to provide any benefit.

Although I won't claim to have tried to cover the full depth of any topic in this issue, I have chosen tasks to show you a wide variety of scenarios you may encounter. Of course, you may want to dive deeper into these issues later, and I'll be immodest enough to suggest checking out the other issues of *USP Journal* to explore at least some of those topics in greater depth.

Before we get started, I would also like to extend a special thank-you to the team that has worked on this issue.

With that said, I think it is time to start….

Chapter 1

Preparing for Departure

Your first step on a wondrous journey

Welcome, welcome, welcome.

In this chapter, I want to make sure you have a development lab set up and ready for use. I realize that if you have never worked with SharePoint before, this may seem a bit bewildering. However, ensuring that we have a common setup allows us to better explore the development scenarios and get similar results.

Now, the first thing you want to do is create a new Visual Studio project. Feel free to use the WSPBuilder project types. Then, add a new feature with receiver, build and…

Oh, you haven't gotten that part yet. OK, sorry about that, I'm getting a bit excited. Before we move into the techy bits, there is some theory that you need to understand, and it relates to how you should learn from this journal issue, so it's a good idea to pay attention.

Before We Begin

This issue of *USP Journal* covers beginning SharePoint development. I'm sure you figured out that much. So, you're likely sitting there right now, expecting to learn about all the cool tools and the latest and greatest innovation in the newest bleeding-edge version of SharePoint.

Well, if those are your expectations, I strongly suggest you ask for a refund, because that's not what you'll learn here.

You see, SharePoint is far less about technology than you think, and it's even less about versions of technology. In this issue, I'm showing you screenshots from SharePoint 2010, and I have some preferences for development tools, but what you learn here is SharePoint, not a specific version of SharePoint, and least of all Visual Studio.

Instead, you'll learn about the core components of SharePoint and how you use those components to build solutions. You'll learn about data and how SharePoint organizes that, about how to teach your data to behave (yes, really), and also how to make your solutions look nice.

These core components change very little from version to version. In fact, when I was updating this journal issue from its former 2007 version, very little of the code actually changed.

You may think that this means what you learn is not the edge of technology, and you may be right. However, if you understand the core concepts and how SharePoint works under the hood, you'll find the tool- and version-specific information far easier to understand.

Another key point, and this may not be applicable to you, is that if you are working in multiple environments, for example as a consultant or in large organizations, you may not even know the target environment for your solutions. You may come to a client or a department, and they'll have a different environment than the one you learned, they may use an older (or newer) version of Visual Studio, or they may ask you to maintain a solution built with a tool with which you have not trained.

Even more importantly, learning how things work under the hood is much more future-proof than simply learning tricks in one version. New versions of SharePoint will continue to emerge, and if you focus on understanding how SharePoint works, you'll find that what you already know will be very valuable when the inevitable next version of SharePoint arrives.

Throughout this issue, I will assume that you will have or find your own tool and SharePoint version preference. I'll show you screenshots and how I prefer to work, but do not take that as anything but the personal likes of an old dog; feel free to use whatever makes sense in your environment.

Speaking of tools, I want to make one more important thing very clear: tools are there to save you from work, not save you from understanding what goes on. If you don't know how a tool works and, importantly, how you can do the same task without the tool, then you are essentially just a tool user.

That is why throughout this issue, you will find many seemingly tedious tasks and tasks that require a great deal of attention to details. These tasks have been designed to help you understand just what happens in those tools so that if you are faced with a new tool or a tool doesn't work the way you expect, you know how to work around that.

Let me illustrate this with a short anecdote. One tool I really love, and I'll introduce you to it later, is SharePoint Manager (SPM). I first became acquainted with SPM in the 2007 version, and it has saved me huge amounts of time and frustration over the years.

When SharePoint 2010 came out in its first beta in November 2009, I immediately wanted to start exploring how to build solutions. Although I could have done this without the help of SPM, having that tool would save me lots of time and allow me to see what goes on under the hood.

Well, it took me just around an hour to build a new version of SharePoint Manager that worked with SharePoint 2010. The source code is available, and I knew enough about how both the tool and SharePoint worked that as soon as the beta bits were available, I could rebuild SharePoint Manager. Carsten, the tool author, took a couple of more hours, but I think he was at a dinner then.

Note

You can read about how that update worked here:

http://blog.furuknap.net/sharepoint-manager-2010-sort-of

Here's another example, also from 2009. It's strange how easy it is to find examples of the importance of understanding SharePoint at its core during upgrade times.

A client for whom I worked was building an intranet to support their employees. Because of time constraints, we couldn't wait to build on SharePoint 2010 because the solution had to be in production by December 2009. However, they were very concerned that they would need to redo the entire development process a few months later to be able to upgrade to SharePoint 2010.

I won't bore you with all the details, and besides, those details are confidential, but suffice to say that I came back to the client the next day, having spent all of 45 minutes modifying the entire solution to work seamlessly in the then-beta version of SharePoint 2010.

It's not my intention to tell this story to brag about how clever I am but rather to illustrate to you how important it is to understand not just how to use the tools but also how they work. That, combined with a core understanding of SharePoint, will allow you to almost forget which version of SharePoint you're using or to which tools you have access.

Further, you will find that your knowledge does not become dated even if Microsoft churns out a new version every quarter. What you learn today about the SharePoint version you prefer will almost certainly be applicable also to the next version, especially if you focus on core functionality.

In turn, that allows you to spend less time learning basics when a new version arrives so you can get more quickly up to speed and have more time to focus on the version-specific features, if that's what you want.

Or spend time with friends, family, playing Skyrim, or whatever your desire may be.

Sorry, let's make sure you get your environment set up first.

Preparing Your Environment

You'll need a place to do your development, and contrary to what you may find when developing on other platforms, SharePoint development is somewhat peculiar about what you need.

Before we get down and dirty with the details, however, there are still a few pieces of information and practical advice that you may want to heed.

First, you'll be building plenty of development environments in SharePoint, because at any point in time, there is likely to be at least two or three actively used major version of SharePoint around. In addition, there are differences between the free version of SharePoint and the paid version, and even multiple feature levels of the paid version.

Further, SharePoint fans out beyond just itself and into other systems, like Active Directory, Exchange, Lync, Office, and so on. If you want to develop features that utilize these external systems, you'll need an environment to match this too.

You may be tempted to just buy a huge development machine and install everything. Don't. You can thank me later, but as a general rule you should install a development environment that matches, as closely as possible, the *target environment* for your solution.

Note

By target environment, I mean the environment in which your solutions are most likely to run.

This may seem a bit premature at this point; after all, we haven't even begun learning any development yet, so how can we possibly know what our target environment will be?

That's a very valid point, especially considering what I wrote earlier in the chapter about learning the core functionality and postponing the version-specific features until a later time.

My suggestion is that if you are just starting out and because it is more important to learn the core of SharePoint rather than version-specific features, you should go with what is most easily available for you. Later, once you become accustomed to working with SharePoint development, you can easily change the lab and development environment to suit your needs.

For learning what you need in this issue, then, you can safely assume that everything you learn will work with any version of SharePoint from the 2007 releases (WSS3 and MOSS). You can use any version of Visual Studio from 2005 and onward, and you can use any development tools you want.

My setup, and the one from which I will grab screenshots, is a SharePoint Foundation 2010 virtual machine, and I'm using Visual Studio 2010 (although I prefer to work with 2008 because it is lighter and faster) and a build tool called WSPBuilder for Visual Studio.

Note

Many companies will insist on using the Visual Studio 2010 Tools for SharePoint. I think these tools are bad because they remove you from what you are doing by disguising many tasks behind opaque visual interfaces.

If you want to get a walk-through of the features of these tools, however, you can check out the enclosed special issue of *USP Journal* called "What's New for SharePoint 2010 Developers":

What You Must Have

Let's take a look at what you must have in order to do SharePoint development.

Well, you'll need SharePoint. And you'll need a place to run SharePoint. And that place will need to be on a server operating system.

Yes, I know, you are likely not walking around with a server under your arm, but fear not, there are ways to solve this, even on a laptop. In fact, I do this all the time when I am demonstrating SharePoint to people while on the move.

What you want to get is one of several available virtual machine software packages. Personally, I prefer VMware Workstation, but you can also use Microsoft's free *Virtual PC*, Microsoft's *Hyper-V*, Oracle's *VirtualBox*, or VMWare's free *VMWare Player*. Today, all virtualizations platforms allow you to run a virtual server on your computer.

Technically, you can use Windows 7 as a development environment for SharePoint. I do not recommend this, and here are six reasons why that I wrote about in a blog post on the issue:

- **SharePoint is a server technology, built to utilize server features that are not available on client operating systems.**
  While you may be able to get SharePoint up and running, you won't be able to evaluate or use many of the really cool features.

- **SharePoint installs services that open security-sensitive features on your computer.**
  In a server environment, security people know what's going on and can prevent problems. On your laptop, you may not be as vigilant. A VM doesn't need access to the Internet at all, and it's very easy to prevent or limit such access.

- **For evaluation purposes or lab work, your environment will likely have to be rebuilt several times.**
  Uninstalling SharePoint is a hassle, especially if you have a failed environment. Deleting a VM you're no longer using takes about five seconds.

- **You cannot take snapshots of a physical machine.**
  Prior to testing new features or making big changes to your environment, it is a good idea to back everything up. That takes another 10 seconds with a VM snapshot, but backing up a physical machine takes far longer.

- **You cannot test multiserver features.**
  Doing farm installations opens up new features that are not possible on a single-server setup. For example, load balancing is not possible, nor is testing interserver communications or distributing services on different servers.

- **You cannot move your SharePoint 2010 installation from a physical machine.**
  A major benefit of VMs is that everything you need to run the server is stored in a single folder. This is a major advantage if you need to reinstall or change your physical host.

You can read the original blog post, including some reader comments, on http://blog.furuknap.net/6-reasons-why-installing-sharepoint-2010-on-windows-7-is-a-stupid-idea.

Even if you have a server you can use for running SharePoint, getting a virtual machine is still a better idea. You see, not only are you going to run SharePoint on that server, but you are also going to do your actual development on that server.

In addition, virtual machines allow you to easily revert to a previous state. As a developer, you will likely do some…weird stuff to your server. Having to reinstall the entire machine just because you want to see what happens if you do something really cool is a hassle.

Note

Even if you have a slower laptop with only a gigabyte or two of memory, you can run a server OS as a guest OS. It may be a bit slower, but I ran a Windows Server 2003 virtual machine on my 1GB laptop for a long time before I bought a faster computer.

I'll include a resource box with download links a little later in this chapter.

The second thing you need is a Windows Server OS. Just pop on over to the Microsoft store; I'm sure you can get something as cheap as a few thousand dollars. Or, you can simply download a 180-day evaluation version, which will allow you more than enough time to evaluate it if this is a path you want to take.

⌘ **Tip**

Even if you decide you want to keep working with SharePoint after your evaluation period expires, you still do not have to buy a full license to Windows Server. Microsoft offers several developer subscriptions that give you access to development and lab software, such as an MSDN subscription or a TechNet subscription. These are far more affordable than buying full licenses but are not for use in production environments.

Now, if you are running on a slower machine and you are satisfied with learning development in a SharePoint 2007 environment, getting Windows Server 2003 may be a better choice than Windows Server 2008. However, if you want to use SharePoint 2010, Windows Server 2008 is your only choice currently (Windows Server 8 is just around the corner at the time of this writing).

As I said, you are going to do your development work on the server. I recommend using Visual Studio as your development platform. You can use other tools as well or even Notepad if you are so inclined. In fact, Notepad is a useful tool for several tasks here, so don't brush this off as completely ludicrous yet.

Visual Studio is also available as a free version, called Visual Studio Express, and you can get 90-day evaluation versions of the paid versions as well. The link to both the free and trial versions is in the following resources box. Some SharePoint development tasks are not available in the Express versions, such as workflow development, so I recommend getting one of the paid trials and using it as a learning environment.

The final item you must have is SharePoint. The content in this book focuses on the free version of SharePoint, and I won't touch SharePoint Server or MOSS topics at all.

To explain briefly why I focus on the free versions of SharePoint, it is important to understand the differences in these seemingly similar pieces of software.

When Microsoft talks about SharePoint, it is often inconsistent in whether it is talking about SharePoint *the platform* or SharePoint *the product*. In short, *the platform* is the foundation product, while *the product* is SharePoint Server (or MOSS on 2007). SharePoint Server is built on top of the SharePoint platform, so anything that Microsoft delivers in SharePoint Server, you can build yourself, if you have the inclination and time.

SharePoint is essentially a platform that allows you or organizations to build applications that solve problems. Many of these problems are similar across organizations, so rather than having every organization build similar solutions to these problems, Microsoft built a for-sale bundle that utilizes SharePoint the platform to give generic solutions to such problems.

Although there are plenty of scenarios for extending the Server set of features, all of these essentially revolve around doing something on SharePoint the platform. So, if you understand SharePoint the platform, then understanding SharePoint the product becomes much easier. If you don't understand SharePoint the platform, then understanding SharePoint the product becomes much more difficult.

Thus, focus on learning SharePoint the platform (WSS and SharePoint Foundation) first and move on to server products only when you understand that platform. This is why I focus solely on SharePoint the platform in this issue.

Tip

If you have problems with installing any of the components required for this issue, Microsoft offers premade virtual machines that you can download that have been set up with Windows Server, Visual Studio, and SharePoint.

What You May Want

The tools I list here are not required but are very useful in your SharePoint development toolkit. I will introduce you to how you use these tools throughout this issue, enough to make you see the benefit of them.

Before I introduce these tools, however, I would like to share with you some general advice.

Tools are very good as productivity tools but horrible as educational tools. If you start relying solely on tools, you will quickly face problems once the tools are not adequate for solving your requirements.

As such, although I recommend you get these tools, I will instruct you in *not* using them. This will also ensure that you are armed with enough knowledge to fully appreciate how much time these tools really save you.

WSPBuilder

Two of the most useful tools in the world of SharePoint development are made by a Dane named Carsten Keutmann. The first of these tools is WSPBuilder—or, more precisely, the WSPBuilder extensions for Visual Studio, but I'll just refer to the tool as WSPBuilder.

WSPBuilder integrates with Visual Studio to speed up your SharePoint development by a factor of lots. WSPBuilder does this by adding several useful project templates that do a lot of the tedious and manual labor required by a SharePoint project.

In addition, WSPBuilder extends Visual Studio by adding several very useful menu options that help you create, deploy, and upgrade WSP solution packages required by SharePoint for solution installation.

You probably won't realize all the power that rests inside this small package until later in the issue, but let me just say at this point that WSPBuilder will save you hours of work on every project you develop.

The even better part is that the tool is free, just like the other optional tools in this chapter. You can download WSPBuilder from CodePlex, but make sure you get the correct version because there is one version for SharePoint 2007 and one for SharePoint 2010, and the two don't mix.

That said, you can easily move solutions from one version to the other; you just need the version that matches your development lab to get all the integration features working correctly.

SharePoint Manager

The other tool from Carsten Keutmann is SharePoint Manager (SPM). SharePoint Manager allows you to inspect your entire SharePoint installation, even information that is not visible through the web interface.

SPM also allows you to make changes to your SharePoint sites, giving you speedy access to information and choices not readily available elsewhere or that require cumbersome command-line commands. SPM is also incredibly useful for verifying that changes you make to SharePoint are applied correctly.

As with WSPBuilder, no amount of explanation at this point will make you appreciate the power of these features of SPM, but throughout this issue, you will surely understand how SPM also will save you tons of frustration and time in your SharePoint development.

Make sure you get the correct version here too, because there is one version for 2007 and one for 2010.

.NET Reflector or ILSPY

Finding the source of a problem sometimes requires really getting your hands dirty. In .NET terms, this means reflection, meaning you split open a .NET DLL file and look at the source code.

Inspecting the SharePoint source code is incredibly useful if you are searching for information that is not written in the documentation. Sadly, this is quite often, and sometimes the documentation does not properly reflect reality either.

.NET Reflector is not a SharePoint tool at all, though, so if you have done your fair share of .NET development before, chances are you already have this tool.

The downside of .NET Reflector is that RedGate, the company that owns the product, recently changed its policy and now charges for .NET Reflector. As such, many developers, me included, have moved to other alternatives.

My current favorite tool is ILSpy, an open source alternative to .NET Reflector that is absolutely free and will remain so because it is released under GNU license terms.

If you already have .NET Reflector or are not concerned with cost, feel free to use that if you want.

SharePoint Designer

Whoa! Hold your horses, cowboy. Designer? I'm no stinking graphics monkey; I'm a developer for crying out loud!

OK, before you grow horns and starts cursing my parents for bringing me into the world to mess with your head, hear me out.

SharePoint Designer is no designer tool, or at least, it's not only a designer tool. In fact, SharePoint Designer, and especially the 2010 version, is a highly specialized application that allows you to develop highly complex no-code solutions for SharePoint.

Despite its legacy (it is actually the child of the much-hated Microsoft FrontPage), SharePoint Designer is now a powerful tool that allows you to quickly prototype solutions in SharePoint without having to resort to the more time-consuming Visual Studio development method.

I'll cover the usage of SharePoint Designer a couple of times later in this issue and also describe its main role as a tool in SharePoint's middle tier of development in Chapter 2.

Microsoft SQL Server

Although not required, having a SQL Server installation will allow you to inspect the databases created by SharePoint. You may also want to set up database backups and perform restores if something goes terribly wrong. Let's face it, as developers, we tend to break things to learn. I highly recommend using a full SQL Server installation.

SharePoint works well with SQL Server 2005, 2008, and 2012, so any version will do. Again, Microsoft provides trial software for you to use, available as a download from the Microsoft site. Installing SQL Server is not covered in this issue, however, so if you want to use a full SQL Server installation, you need to set this up on your own.

Windows SharePoint Services SDK

The software development kit for SharePoint mainly contains the documentation for SharePoint development in a downloadable format. However, although this is very useful in some scenarios, remember that downloaded and offline content will go stale, and the online version of the documents may be more up-to-date.

Although I find the SDK very useful, I also keep the MSDN documentation on speed-dial.

Now, How About Them Links?

OK, OK, you have been patient enough; here are the links to all you will need to follow along in this issue. Feel free to get all of these now, before moving ahead.

Resource Links

Virtual PC

http://www.microsoft.com/downloadS/details.aspx?FamilyID=04d26402-3199-48a3-afa2-2dc0b40a73b6&displaylang=en

VMware Workstation evaluation

http://vmware.com/products/ws/

VirtualBox

https://www.virtualbox.org/

Visual Studio, Express and paid version evaluation

http://www.microsoft.com/visualstudio/en-us/try/default.mspx

WSPBuilder

http://www.codeplex.com/wspbuilder

SharePoint Manager

http://www.codeplex.com/spm

.NET Reflector

http://www.reflector.net/

ILSpy

http://wiki.sharpdevelop.net/ILSpy.ashx

SQL Server Trial

http://www.microsoft.com/sqlserver/en/us/get-sql-server/try-it.aspx

SharePoint SDKs

2007:

http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=15942

2010:
http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=12323

Download the required software, and make sure you install and set up at least your server operating system as well as your preferred version of Visual Studio now. For your lab environment, you may choose any setup you like, and the default setup options will work.

All the .NET examples in this issue will be shown in C#. However, if you're comfortable with another language, don't be afraid to use that. In any case, the amount of .NET work we will do is not that complex, and you should be able to get away with only basic knowledge or even just reading and copying the examples if you have no programming background at all.

Setting Up SharePoint

At this point, you should either have set up or be setting up your Windows Server OS and Visual Studio. I am going to assume that you have done so, so if you haven't, following along with the exercises may prove difficult. Also, my assumptions would be wrong, and I really don't want that to be the case. ☺

You should also have set up SQL Server now. Although this is optional, I do recommend it.

> **Note**
>
> Remember to update all your software after installation. For Windows and Visual Studio, you can accomplish this using Microsoft Update.

I won't be covering the details of installing SharePoint in this issue. The number of possible variations, in Windows Server versions, SharePoint major versions, and service packs and feature packs (both free and the two versions of server), means that an exhaustive walk-through would cover multiple issues alone, and we're here to learn development, after all.

Instead, I'll point you to two good walk-throughs for installing and setting up SharePoint, and I'll give you some pointers to what you should consider during setup.

Regardless of the version you choose, for your development environment, I suggest using an English installation, even if you will mostly be working with a non-English setup. The reason for this is that error messages you encounter will be much easier to debug. I'll get to that later in this chapter, but first, let's look at some resources to get SharePoint installed.

Installing SharePoint 2007

For SharePoint 2007, the installation is more complex than for 2010 simply because you have more choices. While SharePoint 2010 supports only Windows Server 2008, SharePoint 2007 supports Windows Server 2003 and both flavors of Windows Server 2008 (regular and R2).

For a detailed walk-through on installing and configuring SharePoint 2007 on Windows Server 2003, I recommend you refer to the first edition of *Beginning SharePoint Development*, which should accompany this issue, which has both detailed written instructions and several videos to show you the process.

> **Note**
>
> If you did not receive the first edition as part of this issue, contact journal@understandingsharepoint.com, and we'll make sure you get a copy.

For Windows Server 2008, the process is somewhat simpler because more of the features required by SharePoint are installed by default so the quirky steps required on Windows Server 2003 are no longer required.

As such, you should be able to install SharePoint more easily on Windows Server 2008 using the instructions in the first edition of this issue, but if not, Microsoft has provided a free ebook that you can use as a guide.

http://go.microsoft.com/fwlink/?LinkID=123878

In that book, you'll find multiple scenarios for setup, but the one I recommend you follow for setting up SharePoint 2007 on Windows Server 2008 is the scenario called "Deploy a simple farm on the Windows Server 2008 operating system" (on page 53).

Installing SharePoint 2010

For SharePoint 2010, you have a much easier installation process. Essentially, Microsoft has included a prerequisites installer as part of the SharePoint installation process. The prerequisites installer performs virtually all the steps required to prepare your environment for setup, including downloading any required components.

If you should require assistance with your SharePoint 2010 installation, Microsoft provides some decent guides for various scenarios too on the SharePoint Foundation resource center.

http://technet.microsoft.com/en-us/sharepoint/ee518642.aspx

Regardless of which method you choose, keep one thing in mind: do not, under any circumstance, choose the Basic or Simple installation modes. These modes are usable in demo scenarios only and take away a lot of your control over the environment. As developers, we need that control so that we can make sure our solutions work the way we expect, so simply opt for the advanced setup options when prompted.

Configuring SharePoint

After you have installed your chosen version of SharePoint, we'll need to configure SharePoint so we have an environment that we can use for our exploration.

Because we're going to focus on simple development in this issue, we won't actually require too much of our environment. In fact, as long as you ensure that at the end of your setup you have a team site set up, then you'll be good to go.

No idea how to set up a team site? Well, for SharePoint 2007, refer to the first edition of this issue. For SharePoint 2010, after you have installed your environment and enter the Farm Configuration Wizard, simply follow the wizard, and it will help you set up a team site.

After you are done, you should be able to open your site and see either Figure 1 (if installing SharePoint 2007) or Figure 2 (if installing SharePoint 2010).

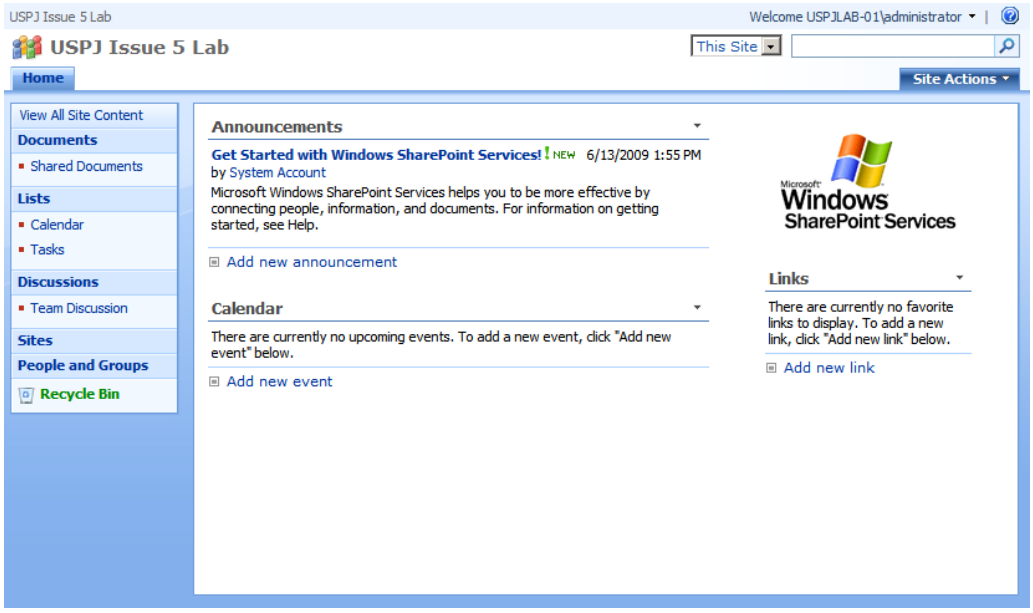

FIGURE 1. SHAREPOINT 2007 TEAM SITE

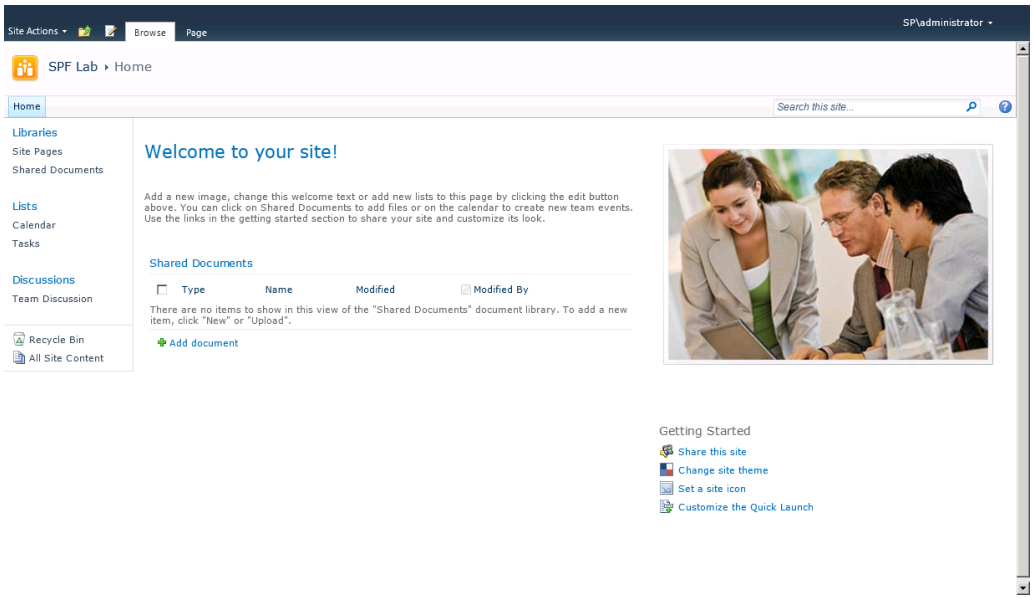

FIGURE 2. SHAREPOINT 2010 TEAM SITE

Note

Please note that from now, I will show screenshots from SharePoint 2010 only.

Help! It Doesn't Work!

I mentioned this earlier in this chapter too, but you will be building a lot of lab environments throughout your SharePoint development career. You're bound to encounter a range of problems, from simple misspelled server names to more complex bugs related to intricacies of SharePoint.

⌘ Tip Section

You're lucky, though, if something goes wrong, because you've taken my advice and installed on a virtual machine. So, even if your current environment is completely unsalvageable, you're not further away from a fresh start than a few menu clicks.

Before you scratch your efforts over something possibly trivial, however, I strongly advise you to do some basic error checking yourself, and to help you along, you have one of the best online communities in the world, ready to assist you. Here are some basic tips to help you resolve your issues.

Google It!

You'll be surprised at the amount of information posted online related to SharePoint. From the beginning of SharePoint, people have dug deep and explored thousands of various combinations of setups, software, third-party plug-ins, operating systems, and SharePoint versions.

Google is your friend in finding this information, so start there and search for any specific error messages you find. Now you'll probably be very grateful that you followed my advice on using an English installation for your development environment, because the bulk of help you'll find will be on English error messages.

Twitter!

Most of the online interaction in the SharePoint community happens on Twitter. If you're stuck, it's a great place to get help. Make sure you tag your tweets with #SharePoint at least, but more specific tags may also be helpful.

Of course, describing a problem in 140 characters can be a challenge, but if you can, you'll find the Twitter community a great source for quick help.

SharePoint StackExchange!

If you're a bit more patient or your question requires more than 140 characters, the SharePoint StackExchange is also a very good place to get help.

http://sharepoint.stackexchange.com/

Make sure you search the current questions before you post, in case someone has already answered your question or something very similar. You'll even earn points for asking interesting questions!

If you utilize these community resources, you'll quickly learn that they are as open and friendly to complete newbies as they are to seasoned professionals. Feel free to contribute yourself, too. Remember that it's not all answers out there; someone needs to ask the questions first.

What's Next?

By now, you should have a basic SharePoint environment set up, including your operating system, SQL Server, Visual Studio, and of course SharePoint.

Before we dive in and begin development, however, it is important to understand the SharePoint development tiers. These tiers, coined as such by Marc Anderson, allow you to understand better the various SharePoint development tasks you may face.

This issue focuses mostly on the third tier of SharePoint development, but as third-tier developers, we are more likely than any other tier developers to utilize all the other tiers as well, so it is important to understand what these tiers mean and include, and that's our topic for the second chapter.

Chapter

2

SharePoint Developer Mentality

It's not all programming

If you're coming from a development background, and it's a great thing if you are, then you may be surprised at what you'll learn in this chapter.

You see, SharePoint development isn't as much about programming and Visual Studio as other development normally is. In fact, you can do impressive amounts of development without using any tools at all, thanks to SharePoint's rich set of features.

In this chapter, I'll introduce you to the three SharePoint development *tiers*, which is terminology coined by Marc Anderson, an aficionado of the middle tier of SharePoint development.

I will also introduce you to other important ideas that you should adapt as a SharePoint developer. Although this isn't a technical chapter, I still highly recommend you read it before you continue your studies.

SharePoint Development Tiers

SharePoint development is a vast area. In fact, one of the most daunting tasks of any SharePoint developer is getting an overview over what they actually need to learn.

For those coming from a traditional development background, often related to some form of programming, this may seem like a strange statement. After all, you just read up on a few new object models and APIs, and you should be good to go, right?

Well, SharePoint development is far more than that. In fact, the portion of SharePoint development you will actually do using programming is miniscule compared to all the other tasks you need to accomplish, and a lot of the time, you won't even be doing development in Visual Studio.

Much of SharePoint's development happens outside the tier that uses Visual Studio and is done by people who may not even realize they are developers. These people do development in the first and middle tiers. Only one tier, the third tier, uses Visual Studio, and even that tier doesn't do programming all of the time.

It is important to understand these tiers because utilizing each tier's strengths and avoiding their weaknesses will allow you to be more efficient as a developer. As a SharePoint developer, especially as a SharePoint programmer, you will need to harness these other tiers to solve your problems.

Let's take a look at what these tiers are.

One of the key strengths of SharePoint is its unique ability to adapt to unique situations and also how SharePoint empowers its users to make those adaptions. Business users, the people who are actually using SharePoint in their daily work, can and should be able to make changes to SharePoint to solve the challenges they face, at least to some extent.

Most of this business user or end user empowerment happens in what we call the *first tier* of SharePoint development. The first tier primarily deals with web configuration and developing solutions through the web interface. Users can build taxonomy solutions, create page hierarchies, configure existing web parts, activate functionality through features and solutions, and so on.

Once you need more power, for example if you want to modify the design of a site, build more advanced data query and displays, or start building custom workflows (I'll talk more about these topics later in this issue), then you need to start using external tools to do your development.

This is when you move into the next tier of development, called the *middle tier*. The most important tools for this type of development are SharePoint Designer and InfoPath.

Finally, if you're still not satisfied with what you can accomplish or if you need to build redeployable solutions and harness the power of managed code (.NET), then you need to move into the realm of third-tier development. Most of third-tier development happens in Visual Studio. However, good third-tier developers utilize other tiers to increase efficiency and accuracy.

End of Preview

But hopefully not the end of your learning

I hope this preview of Beginning SharePoint 2010 Development has given you a taste of what the issue can teach you.

If you want to get the 147 page full issue, including the first edition of the issue, all the additional downloads, videos, and lectures, you can pick that up at

http://beginningsharepointdevelopment.com/

Thanks for reading ☺

.

Previous *USP Journal* Issues

If you are considering buying multiple issues, you can save big by purchasing a subscription bundle from http://www.understandingsharepoint.com/journal/subscriptions.

By purchasing a subscription bundle, you can select any 6 issues but pay for only 5 or you can go big and select 13 issues and pay for only 10.

On the next page, you can see an overview of previously published USP Journal issues.

Volume 1 (SharePoint 2007)

Issue 1: SPCurrentUsers Explained

This issue deals with the solution SPCurrentUsers available from CodePlex.
URL: http://www.understandingsharepoint.com/journal/volume-1/issue-1

Buy now!

Issue 2: Developing SharePoint Content Types

If you want to learn everything about developing SharePoint content types, you should get your hands on issue 2.
URL: http://www.understandingsharepoint.com/journal/volume-1/issue-2

Buy now!

Issue 3: SPTags Explained

If you have ever tried to build custom field types in SharePoint, you definitely want to pick up issue 3.
URL: http://www.understandingsharepoint.com/journal/volume-1/issue-3

Buy now!

Issue 4: SharePoint Designer Workflow

In issue 4 of *Understanding SharePoint Journal*, you will learn how to create workflows in SharePoint Designer 2007.
URL: http://www.understandingsharepoint.com/journal/volume-1/issue-4

Buy now!

Issue 5: Beginning SharePoint Development

Issue 5 introduces you to development for SharePoint. You will learn the basics of various SharePoint development techniques.
URL: http://www.understandingsharepoint.com/journal/volume-1/issue-5

Buy now!

Issue 6: SPThemes and SPSampleData Explained

Understanding SharePoint Journal presents two new solutions, SPThemes and SPSampleData, designed to teach you new aspects of SharePoint development.
URL: http://www.understandingsharepoint.com/journal/volume-1/issue-6

Buy now!

Issue 7: Introducing SharePoint Visual Studio Workflows

In Introducing SharePoint Visual Studio Workflows, I will introduce you to authoring Visual Studio workflows, and show you various aspects of developing workflows for SharePoint in Visual Studio 2008. URL:
http://www.understandingsharepoint.com/journal/volume-1/issue-7

Buy now!

Issue 8: Professional SharePoint Development

This issue teaches you a powerful method for agile SharePoint development, including the 42 lines of code that will save your SharePoint solutions forever URL:
http://www.understandingsharepoint.com/journal/volume-1/issue-8

Buy now!

Issue 9: Advanced Content Type Development

In this issue of USP Journal, I'm going to show you the power of content types in ways you may never have seen before. I'll show you how to extend SharePoint itself by creating custom functionality that you control through the development of content types. URL: http://www.understandingsharepoint.com/journal/volume-1/issue-9

Buy now!

Volume 2 (SharePoint 2010)

Introducing SharePoint 2010

The first SharePoint 2010 book to hit the market, introducing you to SharePoint 2010 as a developer, administrator, or end user.
URL: http://www.introducingsharepoint2010.com/

Buy now!

SharePoint Designer 2010 Workflows

In volume 2, issue 2 of Understanding SharePoint Journal, you will learn how to create workflows in SharePoint Designer 2010.
URL: http://www.introducingsharepoint2010.com/

Buy now!

SharePoint Content Types for Business Users

In this issue, I'll tell you all about why you should use content types and show you step-by-step exercises teaching you how to improve your taxonomy, business processes, and user experience.
URL: http://www.understandingsharepoint.com/journal/volume-2/issue-3

Buy now!

SharePoint Web Part Development

In this issue of Understanding SharePoint Journal, you will learn how to develop SharePoint web parts, including architecture, properties, connections, and user interface.
URL: http://www.sharepointwebpartdevelopment.com/

Buy now!