

Implementing Microsoft® Office SharePoint® Server 2007 and Windows® SharePoint® Services 3.0 Solutions

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Vista, Windows, Windows NT, Windows Server, ActiveX, Excel, FrontPage, InfoPath, IntelliSense, JScript, OneNote, Outlook, PivotChart, PivotTable, PowerPoint, SharePoint, ShapeSheet, Visual Basic, Visual C++, Visual C#, Visual Studio, Visual Web Developer, and Visio are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

[License Agreement](#License)

Implementing Microsoft® Office SharePoint® Server 2007 and Windows® SharePoint® Services 3.0 Solutions

Rona Lustig
Microsoft Corporation

January 2008

**Applies to:** Microsoft® Office SharePoint® Server 2007, Windows® SharePoint® Services 3.0

Summary: This document outlines a methodology for team SharePoint development, customization, and content authoring that aims to accelerate implementation and mitigate production risks. (46 printed pages)

Note   In this paper, Microsoft Office SharePoint Server (MOSS) 2007 and Windows SharePoint Services 3.0 are collectively referred to as Microsoft SharePoint Products and Technologies.

#### Executive Summary

When you begin implementing a solution based on Microsoft Office SharePoint Server (MOSS) 2007 or Windows SharePoint Services on a new server, you may seek comprehensive guidelines to manage the implementation. Concerns may include the following:

* How to manage team development for large MOSS projects?
* How to deploy content and code between development and production environments?
* How to prepare your develop efforts for deployment in a remote hosted environment?
* How to enable developers to participate in several projects at the same time?

This document outlines a methodology for team SharePoint development, customization, and content authoring to help accelerate and mitigate production risks.

The document reviews implementation scenarios, tools, and the development environment. It does not address specific implementations (such as “how to write a Web Part” or “how to manage work items”); however, the document describes the requirements to achieve a successful implementation. In addition, it provides additional references for more research.

**Note:** To keep this document’s focus on solution implementation, we will examine development guidance for deployment in a remote hosted environment in a future document .

#### Table of Contents

[Chapter I - Document Goals 5](#_Toc189026945)

[Chapter II - Implementation Scenarios 7](#_Toc189026946)

[Chapter III - Implementation Environments 8](#_Toc189026947)

[Chapter IV - Implementation Activities 19](#_Toc189026948)

[Chapter V - Deployment Methods 21](#_Toc189026949)

[The SOLUTION Framework 21](#_Toc189026950)

[Content Deployment/Migration API 23](#_Toc189026951)

[Chapter VI - Tools for the Job 28](#_Toc189026952)

[Chapter VII - Implementation Project Plan and Team 30](#_Toc189026953)

[Chapter VIII - Implementation Worksheet 31](#_Toc189026954)

[Chapter IX - Hotfixes 33](#_Toc189026955)

[Chapter X - Testing 34](#_Toc189026956)

[Chapter XI - Summary 36](#_Toc189026957)

[Chapter XII - Glossary 37](#_Toc189026958)

[Chapter XIII - References 39](#_Toc189026959)

[Virtualization 39](#_Toc189026960)

[SDKs and Centers 39](#_Toc189026961)

[Dev Tools 40](#_Toc189026962)

[Packaging Tools 41](#_Toc189026963)

[SOLUTION Framework 42](#_Toc189026964)

[Bin or Global Assembly Cache 42](#_Toc189026965)

[Features 43](#_Toc189026966)

[Authoring and customization 43](#_Toc189026967)

[Content Deployment / Migration 43](#_Toc189026968)

[Team Development 45](#_Toc189026969)

[Testing, Source Control and MSF 46](#_Toc189026970)

[Patterns and Practices 47](#_Toc189026971)

[Chapter XIV - Credits and Thanks To 49](#_Toc189026972)

# Document Goals

#### Vision

This document examines one way to implement a SharePoint Products and Technologies solution.

The method proposed in this document aims to achieve the following goals:

* Accelerate development.
* Mitigate deployment risks.
* Minimize risk to production environment.

#### Background

In this document, we use the following terms and definitions:

* *SharePoint Products and Technologies:* Refers to both *Microsoft Office SharePoint Server 2007* and *Windows SharePoint Services 3.0.*
* *Implemented solutions:* Describes *development*, *customization,* and *content authoring*.
* SOLUTIONS (in uppercase letters)*[[1]](#footnote-2)*: Refers to the SharePoint Solutions Framework. When used with regular roman casing, the term refers to solutions generally (such as “solution to a problem” or a “Microsoft® Visual Studio® solution”).
* *Development*: Refers to the following:
* Features
* SOLUTIONS
* Custom fields
* Event handlers
* Custom controls
* Web Part assemblies
* Document converters
* Excel services functions
* STSADM command extensions
* Custom workflows development
* Configuration information (such as web.config files) or application data
* File system modifications such as CSS (cascading style sheets or site style sheets), definitions, custom lists, and site templates
* *Customization:* Refers to the following:
* Master pages
* Content types
* Custom fields
* Layout pages
* Site columns
* Site design
* CSSs
* *Content authoring:* Refers to the following:
* Web pages and resources used in them, such as images and style sheets

#### Beyond the scope of this document

This document does not address the following:

* Recycle Bin items and state
* Alerts
* Personal Views
* Autocopy references
* Search index and contents
* Workflow instances and associations
* Audit trail
* Change log history
* Check-in/check-out state
* Security state
* Workflow tasks and state
* Search/Shared Services Provider (SSP) settings
* Microsoft Office InfoPath® Forms configuration
* Permissions
* Caching
* Data connections

# Implementation Scenarios

Implementing a server running SharePoint Products and Technologies is a many-faceted process, depending on the purpose of your installation. Depending on the level of your implementation, you may encounter some complexities in setup and implementation, and certain requirements for a robust implementation. Scenarios for levels of implementation include:

* Level 1
	+ Internet facing corporate site
	+ Critical applications
* Level 2
	+ Corporate portal site
	+ Project management
* Level 3
	+ Team collaboration sites

Level 1 implementation scenarios usually require the most effort for getting the server up (up-time), performance, and accuracy. Level 3 scenarios are less complex.

You can engage different tools and skill sets for each level. For example, an end user can perform typical duties related to the management of a team site, such as setting security permissions on content within the site, creating lists, creating subsites, managing the navigation (Quick Launch and Top menu bar) of the site, and configuring the appearance (look and feel) of the site.  However, a professional developer will need to be engaged for the Internet-facing corporate site because this type of SharePoint site typically uses components that are customized, such as master pages and Page Layouts, specialized custom menu controls, and advertisement rotating controls.[[2]](#footnote-3)

Despite their complexity, even level1 implementations can benefit from SharePoint components that do not require code, such as easy-to-set sites, lists, and other objects which shorten development time, and make it possible for nondevelopers to engage in the implementation lifecycle.

This article focuses on level 1 implementations. Projects for level 2 or level 3 sites may also benefit from the concepts described here, but more often do not require this restricted way of implementation.

# Implementation Environments

The following is a preliminary review of various environments that can contribute to the three goals described in this document’s vision:

* Accelerate development.
* Mitigate deployment risks.
* Minimize risk to production environment.

In most cases, you implement a small team’s ad-hoc collaboration site on the production server, and there is no need to use the environments specified here. However, using these environments in development can provide value when a team of developers is engaged in implementing a more complex site, such as an internet facing corporate site. The four environments are:

* **Local, stand-alone development environment:** Specifies a local stand-alone development environment, which is the main contributor to the first goal to accelerate development.
* **Integration area:** Used in the second stage in the development phase, specifies using separate environments to develop the business logic, and to enhance integration, breaking a large problem into stages. This reduces complexity and speeds development (“divide and conquer”).
* **Authoring environment:** Enables content authors (if applicable in the implementation) to contribute content in a dedicated environment.
* **Pilot environment:** Specifies the main contributor to the third goal, to minimize risk to the production environment. The assumption is that if the implementation works well in a pilot environment, it will behave well in production.

After describing the environments, we can examine what you will be migrating, when you are migrating, who will do the migration, and how.

Transporting components repeatedly from the development environment to the production environment to assure a properly tested transformation mechanism achieves the second goal—mitigate deployment risks—because we would likely encounter a deployment issue long before we get to production.

As you review this document, be aware of the following:

* The environments specified in this document may be either real or virtual.
* Several of the environments can be combined into one, to serve as a multi-purpose environment (see details by the end of this chapter).
* The rule is to best fit the implementation requirements, the implementers’ skill sets, and so on.

#### Environment 1: Local, Stand-Alone Development

**Name**: Development Environment (MiniDev)

**Description**: Single-developer, single-machine development environment

**Purpose**: Enhance business logic development

**Attributes**: Single machine, the developer is a local administrator (can perform any function)

**Content**: Development environment and tools, servers and services or stubs



You begin a component’s development by implementing empty interfaces with a minimum of required logic. Such a component is referred to as a *stub* (also known as a “mock object”). A stub should enable reporting of its properties, and conditionally produce log events while running.

Stubs are stored with their complete counterparts for the entire duration of the component lifecycle, and may be used in other environments for troubleshooting (elimination) processes. For example, you may have a poorly performing Web Part Page in a pilot environment. With no source code, you might need to take memory dumps to detect the malfunctioning component. But because you have the stubs for your components, you can replace “real” functionality with its “stub” counterpart, and easily detect a malfunctioning component that is invoked somewhere in the page. You document a stub in the same way as any other complete component. Integration with other components is possible in this environment only by using stubs.

The development environment is a single-machine deployment. For instructions on how to set up this environment, see [Setting Up a Development Environment for the 2007 Microsoft Office System](http://msdn2.microsoft.com/en-us/library/bb330848.aspx).

Because a single developer works in each development environment, all error logs, services restarts and failures belong only to this developer, making it much easier to track issues, troubleshoot, and proceed.[[3]](#footnote-4)

In this environment, the developer has full authority, and no network aspects are involved. Everything is installed locally, and the developer has full control everywhere.

The goal for this environment is to focus on aspects of your business. The point of view is to divide and conquer.

With Microsoft SQL Server™ and MOSS, the local, stand-alone development environment also includes Visual Studio and other development components. Microsoft Office system client products may also be installed in a development environment.

For a complete description of deploying a development environment, see the references at the end of this article.

For more information on development lifecycle, see the Microsoft Solutions Framework (MSF) documentation.

**Testing for the local, stand-alone development environment**

To ensure your development environment is providing what you need, you should do the following:

* Perform unit testing to ensure proper component functionality before moving on to an integration environment.
* Use personas to test your user’s experience in this early environment. A persona is an archetype of a fictional user representing a specific group of typical users. Because a SharePoint site may appear differently to various users, it is advisable to test it with various roles.
* Troubleshoot your components or applications in an environment which does not have a source-level debugger such as Visual Studio, especially in production environments. To prepare for this, produce a symbol (PDB) file along with your released binary, and enable logging, tracing, and other instrumentation options.
* Because changed content will be migrated to later environments (such as authoring), plan your environment to avoid overriding content modified by authors in the authoring environment. For example, to prevent a content author from modifying a critical master page, you should use another master page for content authoring to avoid the critical master page from being overridden by user modifications.

If a project does not include any custom development, a development environment is unnecessary, and the implementation should be performed on the next environment (the integration environment).

Do not develop security or networking-related components in a development environment, because this environment can lack security or secure networking aspects.

Best Practices

Use the following best practices for the local, stand-alone development environment.

1. Plan ahead. Do not let the ease of use of SharePoint Products and Technologies guide your development.
2. Until your code is running as expected, avoid using production components (such as Active Directory) in development environments.
3. Document your code with good comments for future problem solving.
4. Avoid logging events in the operating system event files, because in case of a problem – it is hard to detect operating system issues. General logging practices are provided in the Patterns and Practices reference at the end of the article.
5. Clean up/dispose of **SPWeb** and **SPSite** objects (for more information, see [Best Practices: Using Disposable Windows SharePoint Services Objects](http://msdn2.microsoft.com/en-us/library/aa973248.aspx)).
6. Avoid using the **SPWeb.AllowUnsafeUpdate** property, as setting this property to **true** can leave your code open to security risks, potentially introducing cross-site scripting vulnerabilities.
7. Avoid any direct access to the SharePoint databases. (For more information, see [Support for changes to the databases that are used by Office server products and by Windows SharePoint Services](http://support.microsoft.com/kb/841057)).Accessing these databases programmatically or manually can cause unexpected locking within Microsoft SQL Server that can result in overall performance problems.
8. For code access security, use the following table to help you determine where to install your assemblies, either in the Bin directory or global assembly cache.

|  |  |  |
| --- | --- | --- |
|  | Bin | Global assembly cache |
| Trust level  | As specified in web.config file.Can specify detailed policies. | Grants Full trust to your assembly without affecting the trust level of assemblies installed in the BIN directory. |
| Scope / availability | Web application (Internet Information Services (IIS) Web site). | Affects the whole physical server. |
| Strong name | Optional | Required |
| Requires restart | No | Restart IIS or at least recycle the application pool each time you recompile assemblies. |
| Tighten security |  | This option is less secure. Assemblies installed in the global assembly cache are available to all virtual servers and applications on a server running Windows SharePoint Services. This could represent a potential security risk as it potentially grants a higher level of permission to your assembly across a larger scope than is needed. |
| Licensing |  | Licensing issues may arise due to the global availability of your assembly. |
| Upgrade from SharePoint 2003 | Gradual upgrade does not upgrade items to the new \BIN folder, so you must redeploy your Web Parts. |  |
| Existence | If a Bin directory does not exist, you must add one. Do not store Web Parts in the \_app\_bin directory. | Exists |

For more information about code access security, see the references at the end of this article.

#### Environment 2: Integration

**Name**: Integration Environment

**Description**: A server or server farm, in the same Active Directory domain controller as the development environment.

**Purpose**: First integration point between working components. Use for functional testing of the complete solution.

**Attributes**: Single server or a farm, depending on the implementation requirements.

**Content**: Customization tools may be used from a client machine.



In this environment, you integrate the component with other components and test the entire page. When a component is brought to this environment, you assume that its logic is complete and has been confirmed via unit testing or a similar mechanism. (How components are brought into this environment is addressed later in this document.)

In the integration environment, you address security and network aspects. You install client programs on client computers, not on the server, to match the real production environment.

Unlike in the development environment, all implementation components in the integration environment use the same services (for example, SQL Server or IIS). Authorization is similar to a production environment, and even topology, proxies, and firewalls must be taken into consideration in this environment.

In addition, you can use this environment for customizations. Alternatively, you could use the authoring environment for customization. Methods for transferring customized content between the environments are described in [Chapter IV – Implementation Activities](#_Implementation_Activities).

####  Environment 3: Authoring

**Name**: Authoring Environment

**Description**: A server used for business content authoring.

**Purpose**: Enable content authors to edit and review their content in the earliest, fully functioning and stable environment.

**Attributes**: Can be a single server.

**Content**: Authoring tools, converters, and configurations.



The authoring environment must be an exact software match to the production environment, except for features such as document conversion that are required for content authoring.

You do not deploy source code in this environment, and there are no development tools (except, if needed, remote debugging components).

Content is authored here, preferably from a computer which contains the authoring tools, and then deployed to the pilot and production environments.

This environment can also be used for customization.

It is possible to start with a certain working methodology in one environment, and change it or move to another environment later. For example, assume you customize the pages in the authoring environment, and then you take this environment to a disconnected network on the customer site. You can now “copy” the content of this environment to the integration environment, “reset” the authoring environment, continue working in integration, and continuously bring updates to the authoring environment. The process of moving content (also referred to as deployment, migration, and so on) is addressed in [Chapter IV – Implementation Activities](#_Implementation_Activities).

To resemble production, install developed components in the authoring environment after they pass the integration tests.

Authored content may also be required in other (development and integration) environments, primarily to determine user experience and to perform other tests. Ways to move the content between environments is described later in this document.

Best Practices

1. Choose a single environment for customization, to avoid conflicts and the need for merging updates. Your decision depends on concerns such as the nature of the required customization, the location of this environment (local, remote, or disconnected), the amount of expected modifications, and their frequency.

#### Environment 4: Pilot or Test

**Name**: Pilot (also known as Test) Environment

**Description**: An environment for final tests, both for implementation updates and any related products or technology updates (such as for Windows, SharePoint Products and Technologies, SQL Server).

**Purpose**: Minimize risk to production environment by simulating every move in advance.

**Attributes**: Must be identical to production environment in every aspect that may affect software behavior (see the following sections).

**Content**: See production environment below.



The software installed for this environment must match the software installed in the production environment exactly; you should also attempt to match the hardware in both environments. If you will be conducting stress or load tests, the hardware must also match that used in the production environment.

Every deployment is tested in the pilot environment before it is applied to production.

The pilot environment should resemble the production environment. Therefore it may reside in a remote or hosting environment, or exhibit other characteristics which would make testing harder than in earlier environments. It is advisable to perform tests as soon as they make sense.

A comprehensive list of suggested tests is defined later in [Chapter X - Testing](#_Testing).

Best Practices

1. Make the pilot environment as similar to the production environment as possible, so that you can verify every installation and configuration before applying them to production.
2. In load-balanced environments, make sure your tests don’t stick to a single node.
3. Remember that upon unavailability, performance is no longer an issue. Performance is only one aspect to consider.

#### Environment 5: Production

**Name**: Production Environment

**Description**: The goal of this methodology.

**Purpose**: Answer customer requirements

**Attributes**: Server or server farm, or servers with NO source code, highly secured and optimized to meet customer needs.

**Content**: See planning and deployment at [Microsoft TechNet (Office System)](http://technet2.microsoft.com/Office).



The production environment may be internal or external to the organization. It must be properly configured, both for security and performance. Its configuration must assure a protected, high-availability, well-performing, updateable environment, which is thoroughly monitored, can be well debugged and is always on guard for disaster recovery situations. A clear service level agreement (SLA) should be specified for maintenance operations.

See the Microsoft Office planning and deployment articles on [Microsoft TechNet (Office System)](http://technet2.microsoft.com/Office) for guidance.

Best Practices

1. Keep the RDP instructions and the required setup kits stored in another location, not on the production server. Otherwise, when a production system is offline, online information is inaccessible.
2. The pilot environment is your test field, not the production environment. But ongoing monitoring should include frequent (minimal impact) tests.
3. Keep sampling performance for future comparison, to detect slow degradation of performance.

#### Environment 6: Source Control or Configuration Management

**Name**: Configuration Management (Source Control)

**Description**: Testing and storage for all other environments.

**Purpose**: Build and test automation and managed storage.

**Attributes**: May be used to automate deployment, after successful build, and may have to work with remote pilot and deployment environments.

**Content**: Visual Studio Team System or another source control and test environment.



Microsoft Visual Studio Team System (VSTS) 2005 or 2008 can be used to store, track, test, and manage the implementation lifecycle. This includes both Visual Studio development and exports of the SharePoint artifacts such as master pages, layouts, and style sheets.

As both the pilot and production environments may be disconnected from the organization network, packages may be prepared in the configuration management environment, using a formal build process, to be deployed and updated in the disconnected environments.

Using Team Foundation Server, tests that you conduct in the pilot environment (even if remote) can return the results to the configuration management environment for analysis and reporting.

Best Practices

1. Deploy code before artifacts. So if an artifact (for example, a Feature) relies on specific code, it will be present for activation.
2. Dispose of unused components before deployment to a next environment, including closed Web Parts, unused features and definitions, and so on.
3. Deploy updates to the environments (such as service packs) in the same way you deploy your implementation, starting with the development environments.
4. Keep all environmental deployed updates with your implementation updates, and document what the update is, how you got it, from whom you got it, when you got it, and why you got it. Even a SharePoint list of such notes can save days of research, trial, and error.
5. Document every deployment and test so that you can detect any degradation from the update.



#### Combined environments

If you cannot deploy all the separate environments described earlier, you have at least the following environments:

* A single environment in which to perform development, integration, and authoring
* A pilot or test environment
* A production environment

When implementing a local team site and even corporate portal requirements, you may not need all of the environments. Therefore, specify your implementation scenario at an early stage of the implementation so you know which environments you will need.

See KB article [909840](http://support.microsoft.com/kb/909840) regarding support for MOSS 2007 and Windows SharePoint Services 3.0 in production only when using Microsoft Virtual Server 2005 R2.

Now that we have defined the deployment environments, we can define the deployment process, or how to automate the transfer of developments, customizations, and content between environments.

Best Practices

1. Match installations on all environments (apart from specific environment tools, such as Visual Studio in the development environment).
2. Keep a baseline virtual environment to check against when you encounter an unexpected behavior in your implementation.
3. Backup is the means, restoration is the target. Make a habit to restore regularly and verify the restored environment, preferably using an automated batch test in which results appear on your favorite dashboard.

#### For more information

Deployment, security, manageability, and other aspects of the environments described in this document are beyond the scope of this document, but should be taken into account for every environment. Information is available on [Microsoft TechNet (Office System)](http://technet2.microsoft.com/Office/).

# Implementation Activities

The following table defines three levels of complexity of the implementation process: basic, intermediate, and advanced.

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Basic Level** | **Intermediate Level** | **Advanced Level** |
| **HOW TO DO?** | Manual implementation through the Windows SharePoint Services or MOSS UI and client applications. | Using Microsoft Office SharePoint Designer and basic scripting. | Using a development environment such as Visual Studio. |
| **WHAT IS REQUIRED?** | Requires content management control. | Requires source and configuration control, and thorough testing |
| All levels require means to transport updates between environment s and manage content, source updates, versions, check-outs, and so on. |

The following table describes sample activities, and their mapping to the various complexity levels.

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Basic Level** | **Intermediate Level** | **Advanced Level** |
| **Adding content to be published** | Use the Windows SharePoint Services UI and client applications to author, add, and update content in pre-defined content fields. |  |  |
| **Adding items such as a custom list, column, or content type to sites** | Use the settings pages in the Windows SharePoint Services or MOSS UI for columns, content type, custom lists, and so on.Create a list template (.stp) by using the Windows SharePoint Services UI. |  | Create a Feature that contains a list definition. |
| **Changing the appearance of an entire site** | Change the master page, theme, and so on.Create a site template. | Edit the master page, theme, and so on.Create a site template. |  |
| **Rendering custom content on pages.** | Use existing Web Parts. | Edit the page layout. | Develop a Web Part or a custom control. |
| **Running custom logic upon item change** | Use a built-in workflow. | Create a no-code workflow in SharePoint Designer. | Create a Visual Studio custom workflow. |
| **Extending existing sites with lists, Web Parts, and so on** |  |  | Create an object model script or develop a Feature. |
| **Offering a custom site as an application** | Create a site template .stp using the Windows SharePoint Services UI |  | Create a site definition based on an onet.xml file.Use the solution generator from VSeWSS |
| **Exposing SharePoint data to a custom application** |  | Query our RSS feeds | Web services, FPRPC, Business Data Catalog, and so on. |
| **Exposing application data in SharePoint Products and Technologies** | Save search queries.Use Business Data Catalog Web Parts. | Customize search UI and behavior. | Define and develop Business Data Catalog items, custom search and applications, and possibly use single sign-on or custom authentication provider. |

Finally, we have components that are implemented in Visual studio (these usually reside in the file system), components that are provided within the scope of a site (these usually reside in the SharePoint database), and other components (such as components that were not developed by using Visual Studio, and with a scope larger than a site collection). At this point, how do we transport all the components from one environment to another (providing we have more than a single environment in our implementation)? In [Chapter V - Deployments](#_Deployment_Methods), we provide this information.

# Deployment Methods

Before describing deployment methods, review what you are going to deploy, as specified in the [Background](#_Background) section for development, customization, and content authoring.

You should also understand the various means of generating what you will deploy, so that you can outline the transport mechanism. This outline is divided into three categories:

* Things that can be packaged as a Feature/ SOLUTION
* Content within the site collection scope
* Everything else (such as the items specified in [Beyond the scope of this document](#_Beyond_the_scope)), which is not addressed in this document

In a [Chapter VI - Tools for the Job](#_Tools_for_the), we discuss the tools for the job. For now we can assume we have all the tools we need.

Things that can be packaged as a Feature/SOLUTION can be easily transported by using the SOLUTION Framework, while content within the site collection scope is best transported by content deployment. The next sections describe both approaches.

Assemblies required for a deployment operation should reside in the target environment BEFORE the process of using them begins. For example, before you activate a Feature, as part of a deployment process, ensure the Feature’s assemblies exist in the target environment.

## The SOLUTION Framework

The Windows SharePoint Services SOLUTION framework provides a way to bundle the components for extending Windows SharePoint Services or MOSS in a new file, called a *SOLUTION* file. A SOLUTION file is a cabinet or .CAB-based format with a .wsp extension. A SOLUTION is a deployable, reusable package that can contain a set of Features, site definitions, and assemblies that you can apply to a site, and that you can also enable or disable individually. You can also use the SOLUTION file to deploy the contents of a Web Part package, including assemblies, class resources, .dwp files, and other package components.

To deploy content using the SOLUTION framework:

1. Package a custom Web application into a SOLUTION.
2. Copy the file to another front-end Web server.
3. Add the SOLUTION to the database.
4. Deploy the SOLUTION.

SOLUTIONs can contain the following:

* Feature definitions
* Site definitions
* Template pages and resources
* Resources
* Globally available or Bin assemblies



For more information, see Arjun Ohri’s presentation in the [Microsoft Office System Developers Conference](http://msdn2.microsoft.com/en-us/office/aa905380.aspx), in Windows SharePoint Services 2007: Site Definitions, Features, and SOLUTION Deployment (WS304).

To deploy a SOLUTION to a new farm, simply use the STSADM -addsolution -filename <path to SOLUTION file here> to upload the SOLUTION to the farm.  After you upload the SOLUTION, go into the "SOLUTION management" section under the **Operations** tab in the Central Administration Site, and deploy the SOLUTION.

For step-by-step instructions on how to make a SOLUTION file, see Chris Johnson’s blog on [How SOLUTION deployment has changed development with SharePoint technologies](https://blogs.msdn.com/cjohnson/archive/2006/09/11/749105.aspx).

For information about the SOLUTION framework, see the following sections of the Windows SharePoint Services 3.0 SDK:

* [SOLUTIONS Overview](http://msdn2.microsoft.com/en-us/library/aa543214.aspx)
* [Creating a SOLUTION](http://msdn2.microsoft.com/en-us/library/aa543741.aspx)
* [Deploying a SOLUTION](http://msdn2.microsoft.com/en-us/library/aa544500.aspx)
* [Upgrading a SOLUTION](http://msdn2.microsoft.com/en-us/library/aa543659.aspx)
* [Retracting a SOLUTION](http://msdn2.microsoft.com/en-us/library/aa543958.aspx)
* [Localizing a SOLUTION](http://msdn2.microsoft.com/en-us/library/aa544282.aspx)
* [SOLUTION Schema](http://msdn2.microsoft.com/en-us/library/ms442108.aspx)

In addition to the SOLUTION framework, you can also use the following:

* **SharePoint Solution Generator:** This stand-alone program generates a Site Definition project from an existing SharePoint site. The program enables developers to use the browser and Microsoft Office SharePoint Designer to customize the content of their sites before creating code by using Visual Studio. The Solution Generator is part of [Windows SharePoint Services 3.0 Tools: Visual Studio 2005 Extensions](http://www.microsoft.com/downloads/details.aspx?familyid=19F21E5E-B715-4F0C-B959-8C6DCBDC1057&displaylang=en#Overview).
* The guide to [Working with Features](http://msdn2.microsoft.com/en-us/library/ms460318.aspx) in the [Windows SharePoint Services SDK](http://www.microsoft.com/downloads/details.aspx?FamilyID=05E0DD12-8394-402B-8936-A07FE8AFAFFD&displaylang=en) with Mike Ammerlaan’s explanation on [The concept of a "Feature" in SharePoint](http://sharepoint.microsoft.com/blogs/mike/Lists/Posts/Post.aspx?ID=7).
* **SharePoint Feature Manager:** Provided on [Todd Baginski’s blog](http://www.sharepointblogs.com/files/folders/tbaginski/entry2449.aspx) assists with the deployment features and also with other STSADM commands.
* The [Solution Pack Generator](http://www.codeplex.com/WSPGenerator) and the [WSPbuilder](http://www.codeplex.com/wspbuilder), which help in the creation and generation of SharePoint solution packages.

## Content Deployment/Migration API

The content deployment/migration API deploys content and page modifications. This may be the responsibility of the content author, who is not a developer and does not write scripts. However, if you are handling scheduled batch updates to published content, this type of content deployment may be performed by the farm administrator.

For these various scenarios, you may need a variety of content deployment methods. These are described in the following table. The table focuses on the farm-level transports that are generally applied on a schedule, instead of the more local types of transports, such as using the content and structure pages.

|  |  |  |
| --- | --- | --- |
|  | **Content Deployment** | **Content Migration** |
| **Method** | Central Administration | STSADM | Web Service | Object Model (API) |
| **Source and destination environ-ments connectivity** | Only connected scenarios | Supports exporting specified content from a source Windows SharePoint Services site collection to a cabinet (.cab) file in XML format. | jobName - File name to use for content migration packages | Recommended method in disconnected scenarios[[4]](#footnote-5) |
| **Scope** | Sites in a site collection | The entire Web site onlyYou cannot pick individual items or lists | The entire Web site onlyYou cannot pick individual items or lists | You can migrate anything, from a Web site to an item in a list, or a single document in a library |
| **GUIDs** |  | Don’t retain | Don’t retain | Retains |
| **Usage** | * Define path and job in the central administration pages
* Specify whether or not to deploy user names associated with the content and how to deploy permissions on the content
* Specify whether to deploy all content, or just content that has been changed or added
* Specify whether to send e-mail when a job succeeds or fails
 | * Export and import operations
* To preserve timestamps, security info, and users, Use the -*includeusersecurity* parameter
 | * **ExportWeb** and **ImportWeb** methods in the Sites Web service
 | * **Microsoft.SharePoint.Deployment** namespace
* Specify settings **SPExportSettings** and **SPImportSettings** objects
* Use are **SPExport** and **SPImport**
* You can choose whether to include information about security, versioning, user roles, and other metadata appropriate to the objects you are migrating
 |
| **Requires Code** | No | None or batch level | Web service invocation | Object model / API |
| **Complexity** | Administrator settings | CommonSample batch referenced | Most complexLess common | Most complexComplete samples with technical articles in the SDKs |
| **Notes** | Easy to set a schedule for timely content updateSpecial “Quick Deploy” job for site owners | Most useful when the goal is to move basic content from an entire Web site or re-parenting a Web site | Migrate data from a remote server | Provides the most control over your data migration scenarios |
| **How to Articles** | [Plan content deployment](http://technet2.microsoft.com/Office/en-us/library/edcdacca-8013-460e-95a0-d2b83b6cc7ef1033.mspx?mfr=true)[How to: Deploy Content Between Servers](http://msdn2.microsoft.com/en-us/library/ms496233.aspx) | [SharePoint Content Migration Object Model and Content Migration Packages](http://blogs.msdn.com/jackiebo/archive/2007/02/23/sharepoint-content-migration-object-model-and-content-migration-packages.aspx) | [Sites Methods](http://msdn2.microsoft.com/en-us/library/sites.sites_methods.aspx)[ExportWeb](http://msdn2.microsoft.com/en-us/library/sites.sites.exportweb.aspx)[ImportWeb](http://msdn2.microsoft.com/en-us/library/sites.sites.importweb.aspx) | [How to: Customize Deployment for Disconnected Scenarios](http://msdn2.microsoft.com/en-us/library/aa981161.aspx)[How to: Migrate a Web site from one location to another](http://msdn2.microsoft.com/en-us/library/ms438819.aspx) |

#### What is migrated by using the content migration API?

* Copies of content (original content is not migrated).
* Web pages and resources used by the copied pages.
* Any items in the content database that a deployed Web page depends on, such as images, style sheets, or layout pages.
* Dependent page layout and packages for a page– *even if the dependent resources aren’t in the same site*.
* Most recent major and minor versions of a content item.
* Scheduling information for an associated item.
* Web Parts (however, no assemblies are migrated; for custom Web Parts, install the DLLs at the destination location for proper functioning).

#### What is NOT migrated using the content migration API?

* Recycle Bin items and state
* Alerts
* Features and SOLUTIONS
* Personal Views
* Autocopy references
* Web Part assemblies
* Workflow instances and associations
* Anything outside the scope of the **SPWeb**
* Search index and contents
* Configuration information (such as web.config files) or application data
* Audit trail
* Change log history
* Check-in/check-out state
* Security state
* Workflow tasks and state
* File system modifications: CSS, definitions (custom list and site templates), etc.

#### To perform the migration process

1. Prepare a blank site at the destination.
2. Export content from the source site.
3. Use VSTS to manage, version, and schedule deployment for the exported package , as you would any code package ready to be moved to the destination.
4. After moving the package to the destination, import the package into the site.

You can download a tool that will demonstrate export and import based on the object model, from [Content Migration API in Action](http://blogs.msdn.com/ronalus/archive/2007/08/10/content-migration-api-in-action.aspx). The tool has two versions:

* A GUI version (see the following figures)
* A batch file version that you can run from a command line that depends on a settings file, which can be prepared using the GUI version and can be used as a task in MSBuild

 EXPORT

 IMPORT 

In addition to methods for the content deployment and content migration, there are also other methods that can be used to copy content between site collections. For more information, see the following links:

* [Export or import a Web package](http://office.microsoft.com/en-us/sharepointdesigner/HA100819231033.aspx) ([Microsoft Office SharePoint Designer 2007](http://office.microsoft.com/en-us/sharepointdesigner/FX100646991033.aspx))
* [Back up, restore, or move a SharePoint site](http://office.microsoft.com/en-us/sharepointdesigner/HA100699391033.aspx) ([Microsoft Office SharePoint Designer 2007](http://office.microsoft.com/en-us/sharepointdesigner/FX100646991033.aspx))
* [Save a site as a site template](http://office.microsoft.com/en-us/help/HA101577791033.aspx?pid=CL100605171033)
* [Share customizations by saving them as templates](http://office.microsoft.com/en-us/help/HA101577801033.aspx?pid=CL100605171033)
* [Administering backup and recovery for Office SharePoint Server 2007](http://technet2.microsoft.com/Office/en-us/library/468b8840-258c-42ba-a89e-9d09701a7cda1033.mspx?mfr=true)
* [Administering backup and recovery for Windows SharePoint Services 3.0 technology](http://technet2.microsoft.com/windowsserver/WSS/en/library/64171b8c-5608-4e69-881a-67996080b7ff1033.mspx?mfr=true)
* [Plan for backup and restore (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/054c3d6d-a0d3-448a-864b-93db6283d7ce1033.mspx?mfr=true)
* [Migrate Office SharePoint Server 2007 by using built-in tools](http://technet2.microsoft.com/Office/en-us/library/42511e01-ecdd-4dc5-b06f-35aaada8a5d81033.mspx?mfr=true)
* [Migrate Office SharePoint Server 2007 by using Central Administration](http://technet2.microsoft.com/Office/en-us/library/391fd37a-daf0-47e3-810b-5cd8c2a4fc341033.mspx?mfr=true)
* [Migrate Windows SharePoint Services 3.0 by using built-in tools](http://technet2.microsoft.com/windowsserver/WSS/en/library/d30dccce-a9cc-46b4-9622-00c7e115858d1033.mspx?mfr=true)
* [Plan for backup and restore (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/054c3d6d-a0d3-448a-864b-93db6283d7ce1033.mspx?mfr=true)
	+ [Plan for tools](http://technet2.microsoft.com/Office/en-us/library/054c3d6d-a0d3-448a-864b-93db6283d7ce1033.mspx#section1)
	+ [Backup Strategies](http://technet2.microsoft.com/Office/en-us/library/054c3d6d-a0d3-448a-864b-93db6283d7ce1033.mspx#section2)
	+ [Plan for backup types and schedules](http://technet2.microsoft.com/Office/en-us/library/054c3d6d-a0d3-448a-864b-93db6283d7ce1033.mspx#section3)
* [Administering backup and recovery for Office SharePoint Server 2007](http://technet2.microsoft.com/Office/f?en-us/library/468b8840-258c-42ba-a89e-9d09701a7cda1033.mspx)
* [Using Backup and Restore (Office SharePoint Server 2007)](http://technet2.microsoft.com/Office/ja-JP/library/5a39029f-0939-452b-acd2-44ff87e7ff711041.mspx?mfr=true)
* [Migrate content or sites after upgrade (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/16a7e571-3531-4a4e-baa7-f348a9f9d1d11033.mspx?mfr=true)
* [Troubleshoot backup and recovery for Office SharePoint Server 2007](http://technet2.microsoft.com/Office/f?en-us/library/991e55bd-b3e2-4260-beb3-b23861934f0a1033.mspx)

Content migration is much slower than other methods for backing up and restoring content (such as STSADM or SQL backup), because each object must be migrated rather than restoring objects in bulk. However, using the object model provides much greater control and offers a wide collection of parameters (as you can see from the tool API).

# Tools for the Job

Use the following tools and resources for implementing your SharePoint solutions.

**Development tools**

* MOSS and Windows SharePoint Services SDKs (see )
* Visual Studio 2005 (see )
* Windows SharePoint Services 3.0 Tools: Visual Studio 2005 Extensions (see )
* Internet Explorer Developer Toolbar (see )

**Packaging tools**

* Windows SharePoint Services 3.0 Tools: Visual Studio 2005 Extensions (see )
* MakeCAB (see )
* STSADM
* Content Publishing demo tool (see )
* The [Solution Pack Generator](http://www.codeplex.com/WSPGenerator) and the [WSPbuilder](http://www.codeplex.com/wspbuilder)

**Customization tools**

* A supported browser (see )
* SharePoint Designer 2007

**Content authoring tools**

* A supported browser (see )
* Smart client authoring tools (see )

**Content deployment, staging, and migration tools**

* Content deployment (see )
* Content migration API (see )
* The [Content Migration API in Action](http://blogs.msdn.com/ronalus/archive/2007/08/10/content-migration-api-in-action.aspx)

**Virtual environments**

* May be used to achieve multiple environments to perform various implementation scenarios, from content sharing and authoring to development
* See notes on supportability in

**Testing and Source Control**

* Visual Studio Team System2005
* Best Practices Analyzer (see )
* Data Population Tool (see )
* Fiddler (see )

Additional tools can be found on <http://www.codeplex.com/>.

# Implementation Project Plan and Team

This section describes a sample implementation project plan. Activities are presented in the order in which they should be performed. For each activity, the team members who should participate are indicated.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | Work Days |  | Architect | Designer | Developer | infrastructure specialist | Business | Milestones |
| Vision and scope |  |  |  |  |  |  |  | Scope Approve |
| **Total** |  |  |  |  |  |  |  |  |
| Development environments planning |  |  |  |  |  |  |  |  |
| Development environments installation |  |  |  |  |  |  |  |  |
| Readiness |  |  |  |  |  |  |  | Training completed |
| **Total** |  |  |  |  |  |  |  |  |
| Planning - functional requirements |  |  |  |  |  |  |  |  |
| Planning - Non-functional requirements |  |  |  |  |  |  |  |  |
| Initial design |  |  |  |  |  |  |  |  |
| Integration and authoring environments planning |  |  |  |  |  |  |  |  |
| Integration and authoring environments installation |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  | V1 prototype Approve |
| **Total** |  |  |  |  |  |  |  |  |
| Pilot and production environments planning |  |  |  |  |  |  |  |  |
| Pilot and production environments installation |  |  |  |  |  |  |  |  |
| Design |  |  |  |  |  |  |  |  |
| Application development |  |  |  |  |  |  |  |  |
| Deployment to integration, authoring and pilot environments |  |  |  |  |  |  |  |  |
| Pilot environment testing |  |  |  |  |  |  |  | V2 prototype Approve |
| **Total** |  |  |  |  |  |  |  |  |
| Deployment to production environment, using a proven procedure |  |  |  |  |  |  |  |  |
| Production environment testing |  |  |  |  |  |  |  | Project approved |
| **Total** |  |  |  |  |  |  |  |  |
| **Grand Total** |  |  |  |  |  |  |  |  |

# Implementation Worksheet

The following is a sample worksheet for use by a team implementing an enterprise portal or Internet site based on MOSS 2007:

|  |  |
| --- | --- |
| **Aspect to review** | **Examples** |
| **Vision and scope** | ... |
| **Implementation goal** | Corporate portal, collaboration sites, Internet facing organization site, record management, archive, storage garage, etc. |
| **Number of developers** |  |
| **Number of designers** |  |
| **Number of content authors** |  |
| **Development types** | Web Parts, IFilters, Workflow |
| **Customization types** | Templates, definitions, master pages |
| **Development tools and developers skill set** | Visual Studio 2005, SharePoint Designer |
| **Customization tools and designers skill set** | Templates, definitions, master pages |
| **Content authoring tools and authors skill set** | Templates, definitions, master pages |
| **Source control** | Visual Studio Team System, Team Foundation Server, etc. |
| **Testing methodologies** |  |
| **Virtual machines** |  |
| **Terminal services** |  |
| **Other constraints** | Low bandwidth, RAM and disk space in dev machines, etc. |
| **Experience in SharePoint** | 2001 (v 1) , 2003 (v2), 2007 (v 3) |
| **Experience in Microsoft .NET Framework development** | Microsoft .NET Framework 2.0,.NET Framework 3.0 |
| **Experience in Web development** | Microsoft .NET Framework |
| **Time to market** | Yesterday is not a valid answer |

For more planning considerations (for example, expected volumes, update frequency, security requirements, production monitoring), see the following worksheets:

* [Planning worksheets for Windows SharePoint Services 3.0](http://office.microsoft.com/en-us/help/HA101638791033.aspx?pid=CL100605171033)
* [Planning worksheets for Windows SharePoint Services 3.0](http://technet2.microsoft.com/windowsserver/WSS/en/library/9985f2fa-421d-4342-a17e-286055273c371033.mspx?mfr=true)
* [Planning worksheets for Office SharePoint Server 2007](http://office.microsoft.com/en-us/help/HA101638321033.aspx?pid=CL100605171033)
* [Planning worksheets for Office SharePoint Server 2007](http://technet2.microsoft.com/Office/en-us/library/49cf7edd-14ee-445b-8ab0-1d1339f2435f1033.mspx?mfr=true)

# Hotfixes

Some issues you encounter may require a hotfix. Because a hotfix is intended to correct only the problem that is described in its article, apply it only to systems that are experiencing that specific problem.

After release, a hotfix may receive additional testing and be included in a service pack. Therefore, if you are not severely affected by a problem, we recommend that you wait for the next service pack that contains this hotfix instead of installing an individual hotfix.

To look for hotfixes and updates you may need, use the following main resources:

* [Newly published content for Office SharePoint Server 2007](http://technet2.microsoft.com/Office/en-us/library/21dcf8aa-8a6e-4325-aa17-0188e491361c1033.mspx?mfr=true)
* [Newly published content for windows SharePoint Services](http://technet2.microsoft.com/windowsserver/WSS/en/library/8a9690e4-6917-4aab-82ce-cdbe989a20531033.mspx?mfr=true)
* The Microsoft Hel and Support knowledge base advanced search functionality, for example:
* <http://support.microsoft.com/search/default.aspx?catalog=LCID%3d2057&spid=global&query=%22Windows%20SharePoint%20Services%203.0%22&pwt=false&title=false&kt=PHRASE&mdt=186&res=100&ast=3&mode=a&adv=1&range=1-156>
* <http://support.microsoft.com/search/default.aspx?catalog=LCID%3d2057&spid=global&query=%22Microsoft%20Office%20SharePoint%20Server%202007%22&adv=&ast=3&mode=a&cat=False&kt=PHRASE&title=false&mdt=186&pwt=False&res=100&range=1-163>

If you need a hotfix, you can get it easily by submitting your requests at the [Hotfix Request Web Submission Form site](https://support.microsoft.com/contactus2/emailcontact.aspx?scid=sw;en;1410&WS=hotfix) . See a review at [Microsoft Offers Customers Hot Fixes via E-mail](http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9028080&intsrc=news_ts_head).

If you conclude that you do need a fix, and you deploy it, consider it as part of the installation kit. Record its exact source (whether 32-bit or 64-bit,, language, and so on), and store it with all other installed components.

# Testing

The following is a common list of test areas for a server-developed solution. You may need only some of the tests for your implementation; however, you should review the list to ensure that you don’t overlook a test you might need.

#### Preparation

Before you begin, be sure you have the following prepared for your tests:

* Working environment to be tested (client, server, other)
* Client computer, strong enough to test the server or close enough to expected user computer, to test the client applications
* List of user names and passwords to be used in the tests (personas are a great way to ensure that you test various user experiences; other tests might require a large number of each user type)
* Preparation and capture tools
* A user who is authorized to install the testing tools on server and client computers
* The following specifications:
	+ List of components composing the solution
	+ List of use cases to test and related actors
	+ User (or other) activities mix (such as search, browse, and upload document)
	+ Activities frequency (minimum and maximum)
	+ Authentication and authorization
	+ Security mechanisms (for example, Secure Sockets Layer [SSL])
	+ Hardenings employed
	+ Content volumes
	+ Affecting activities (such as backup and index)
	+ Expected concurrency or reentrancy
	+ Topology (such as bandwidth or latency )
	+ Special requests

#### Success criteria

Define your criteria for successful testing before testing starts, preferably having the customer create the definition, as this will influence the tests you employ, and the testing methods. Success criteria should be measurable, such as the desired SLA and so on.

#### Tests

* Functionality – ensure “the right solution for the correct problem”
* UI – by accepted design guidelines. Consider accessibility and localization globalization aspects
* Security – such as authentication, authorization, and input tests
* Performance
	+ Baseline performance – ensure the system exceeds performance goals
	+ Load test –determine the maximum number of a defined “mix” that conforms to a defined performance goal
* Reliability
	+ Exception management (for example, writing to a log when the disk is full)
	+ Stress test – resources limits and proper release
	+ Concurrency – smooth functionality of multiple “mixes”
	+ Special operations impact – operation when “affecting activities” (such as backup) occur
* High availability – usually include both software and hardware aspects
* Scalability
	+ Concerns the growth of content, working hours, activity type ,and actors
	+ A special case may be geo-scaling (such as spreading deployment to remote locations with specific topology characteristics)
* Troubleshooting
	+ Ability to troubleshoot in the production environment (such as use of debugging tools, correct symbols, instrumentation)
* Manageability
	+ Monitoring and problem alerts
	+ Backup (and restore testing!) – verify service window length with expected volumes
	+ Disaster recovery (from disconnection to complete system down)
* Documentation and operation guides (should also exist offline)
	+ Installation documentation
	+ User, operator, developer and other required documentations
	+ Special and regular operation procedures

# Summary

This document described an approach to implementing a SharePoint solution and some content and the techniques to use.

Beginning with implementation scenarios, and the tools to achieve them, the document outlined a methodology “to divide and conquer” so that you can accelerate development. It also specified the mechanisms to move the content (transform the ingredients) from one environment to another.

The approach outlined herein attempts to provide you with a safe and forward-moving methodology. The goal is to maximize stability and consistency of content within the servers, and rehearse the process enough to ensure that, when you get to production environment, you know your way, and can minimize the risk to damage this environment.

This document is only a high-level overview with some technical references. This document assumes you have addressed proper manageability, including but not limited to backup and restore, monitoring, proper configuration for performance and security, and other aspects that must be considered in both development and production environments. For more information on those aspects, see [Microsoft TechNet (Office System)](http://technet2.microsoft.com/Office).

# Glossary

|  |  |
| --- | --- |
| **Term** | **Description and references** |
| **SharePoint Products and Technologies** | Term to refer to both Microsoft Office SharePoint Server (MOSS) and Windows SharePoint Services 3.0 |
| Microsoft Office SharePoint Server (MOSS) 2007 | Integrated suite of server capabilities that can help improve organizational effectiveness by providing comprehensive content management and enterprise search, accelerating shared business processes, and facilitating information-sharing across boundaries for better business insight. MOSS 2007 supports all intranet, extranet, and Web applications across an enterprise within one integrated platform, instead of relying on separate fragmented systems. Additionally, this collaboration and content management server provides IT professionals and developers with the platform and tools they need for server administration, application extensibility, and interoperability. ([Microsoft Office SharePoint Server 2007 product overview](http://office.microsoft.com/en-us/sharepointserver/HA101656531033.aspx)) |
| Windows SharePoint Services | Versatile technology that organizations and business units of all sizes can use to increase the efficiency of business processes and improve team productivity. With tools for collaboration that help people stay connected across organizational and geographic boundaries, Windows SharePoint Services gives people access to information they need. Built on Windows Server 2003, Windows SharePoint Services also provides a foundation platform for building Web-based business applications that can flex and scale easily to meet the changing and growing needs of your business. Robust administrative controls for managing storage and Web infrastructure give IT departments a cost-effective way to implement and manage a high-performance collaboration environment. With a familiar, Web-based interface and close integration with everyday tools including the Microsoft Office system, Windows SharePoint Services is easy to use and can be deployed rapidly.([Windows SharePoint Services 3.0 Overview](http://www.microsoft.com/technet/windowsserver/sharepoint/techinfo/overview.mspx)) |
| **SOLUTION framework** | Provides a way to bundle all of the components for extending Windows SharePoint Services or Office SharePoint Server in a file, called a SOLUTION file. A SOLUTION file is a cabinet or .CAB-based format with a .wsp extension. A SOLUTION is a deployable, reusable package that can contain a set of features, site definitions, and assemblies that you can apply to a site, and can also enable or disable individually. You can use the SOLUTION file to deploy the contents of a Web Part package, including assemblies, class resources, .dwp files, and other package components. ([Solutions Overview](http://msdn2.microsoft.com/en-us/library/aa543214.aspx)) |
| **Implementation** | SharePoint related development, customization, and content authoring. |
| **Development** | Code component that integrates with SharePoint Products and Technologies.Most frequent uses of code are Web Parts and controls, but there are many others, such as custom workflows, custom event handlers, external applications, Features, and more. |
| **Customization** | Development of content-related artifacts such as master pages, style sheets, and layout pages. Was originally denoting a wide user-modification (usually by administrator), as opposed to personalization (performed by and for a single user).In SharePoint Products and Technologies, a lot more can be accomplished via the browser, and in SharePoint Designer, without using any traditional developer tool (for example, Visual Studio) and the scope of customization may be wider than a single Web site. |
| **Content authoring** | Publishing related term that refers to adding and updating content in preset containers within a Web page, and specifying content properties, such as publishing and expiration time, searchable meta-tags, and so onContent authoring may also involve several languages and span several correlated Web pages, and MOSS provides the tools and guidance for such activities. |
| **Microsoft Office SharePoint Designer** | Part of the 2007 Office system applications. The successor of Microsoft Office FrontPage® 2003 and the recommended customization client tool for SharePoint Products and Technologies. |
| **Virtualization solution** | Capability to enable IT managers to run multiple operating systems, applications, and middleware on a single physical computer, allowing customers to cut costs and response time without sacrificing resources. ([Microsoft Virtualization](http://www.microsoft.com/windowsserversystem/virtualization/default.mspx)) |

# References

## Virtualization

* [Virtual Server 2005 R2 Support for WSS 3.0 and MOSS 2007](http://blogs.msdn.com/sharepoint/archive/2007/01/05/virtual-server-2005-r2-support-for-wss-3-0-and-moss-2007.aspx)

Although there was no support for virtual environments in Microsoft Office SharePoint Portal Server 2003 and Windows SharePoint Services 2.0 in KB [909840](http://support.microsoft.com/kb/909840), there is now product support for Windows SharePoint Services 3.0 and MOSS 2007 with [Microsoft Virtual Server 2005 R2 x86 or x64](http://www.microsoft.com/windowsserversystem/virtualserver/default.mspx).  You can [download it for free or order a CD](http://www.microsoft.com/windowsserversystem/virtualserver/software/default.mspx).

* Virtual PC

Although Microsoft Virtual PC might be used by product support for reproducing issues or in nonproduction environments, there is no support for Virtual PC.  **Do not run Windows SharePoint Services 3.0 and Office SharePoint Server 2007 production environments with Virtual PC.** Support for third-party virtualization is similar to earlier SharePoint releases. Please refer to KB [897615](http://support.microsoft.com/kb/897615).  It's possible that product support will require you to reproduce an issue in an environment that is not virtualized, if they suspect that the virtual environment is a potential contributing factor in the issue.

* [Windows SharePoint Services, SharePoint Portal Server 2003, and SharePoint Server 2007 do not support Virtual PC and Virtual Server for production environments](http://support.microsoft.com/kb/909840)

Virtualization is supported for production when using Microsoft® Virtual Server 2005 R2 for:

* + Windows SharePoint Services 3.0
	+ MOSS 2007

Review the article for important notes on options that are not supported.

* [Support policy for Microsoft software running in non-Microsoft hardware virtualization software](http://support.microsoft.com/default.aspx/kb/897615)
* Microsoft Virtual Server 2005 R2
	+ [Overview](http://www.microsoft.com/windowsserversystem/virtualserver/default.mspx)
	+ [Download or Order for Free](http://www.microsoft.com/technet/virtualserver/software/default.mspx)
	+ [Virtual Server 2005 Migration Toolkit](http://www.microsoft.com/technet/virtualserver/downloads/vsmt.mspx)
* Microsoft Virtual PC 2007
	+ [Overview](http://www.microsoft.com/windows/products/winfamily/virtualpc/default.mspx)
	+ [Demo](http://www.microsoft.com/mac/products/virtualpc/virtualpc.aspx?pid=vpcdemo)
	+ [Download](http://www.microsoft.com/downloads/details.aspx?FamilyId=04D26402-3199-48A3-AFA2-2DC0B40A73B6&displaylang=en)

## SDKs and Centers

* Online
	+ [Windows SharePoint Services](http://msdn2.microsoft.com/en-us/library/bb264594.aspx)
	+ [MOSS](http://msdn2.microsoft.com/en-us/library/bb187390.aspx)
* Download
	+ [Windows SharePoint Services 3.0: Software Development Kit (SDK)](http://www.microsoft.com/downloads/details.aspx?familyid=05E0DD12-8394-402B-8936-A07FE8AFAFFD&displaylang=en)

Contains conceptual overviews, programming tasks, samples, and references to guide you in developing solutions based on Windows SharePoint Services 3.0.

* + [SharePoint Server 2007 SDK: Software Development Kit and Enterprise Content Management Starter Kit](http://www.microsoft.com/downloads/details.aspx?familyid=6D94E307-67D9-41AC-B2D6-0074D6286FA9&displaylang=en)

Contains conceptual overviews, “How Do I…?” programming tasks, developer tools, code samples, references, and an Enterprise Content Management (ECM) starter kit to guide you in developing solutions based on MOSS 2007.

* Centers
	+ [SharePoint Server 2007 Developer Portal](http://msdn2.microsoft.com/en-us/office/aa905503.aspx)

Microsoft Office SharePoint Server developers' home on MSDN. Find information about planning, coding, building, deploying, and migrating solutions for SharePoint Server.

* + [Windows SharePoint Services Developer Center](http://msdn2.microsoft.com/en-us/sharepoint/default.aspx)

Windows SharePoint Services continues to provide the solution platform for the next version of Microsoft SharePoint Products and Technologies. Windows SharePoint Services 3.0 takes full advantage of Microsoft ASP.NET 2.0 and the core Microsoft .NET Framework 2.0 runtime. The new features and added programmability support in Windows SharePoint Services 3.0 provide a wealth of development opportunities.

* + [Governance Information for SharePoint Server 2007](http://technet.microsoft.com/en-us/office/sharepointserver/bb507202.aspx)

Governance is the set of roles, responsibilities, and processes that you put in place in an enterprise to guide the development and use of a solution based on SharePoint Products and Technologies.

This page contains tools and resources to help business decision makers and IT professionals govern their SharePoint Products and Technologies environment. By using the governance techniques and best practices available from this page, an enterprise can align its policies for using SharePoint Products and Technologies with its culture and goals while still enabling teams and individuals to effectively collaborate and share information.

* + [Channel9 Screencasts](http://channel9.msdn.com/ShowForum.aspx?ForumID=38&TagID=59)
	+ [Office Developer How-to Center](http://msdn2.microsoft.com/en-us/office/bb266408.aspx)

This section of the portal compiles task-based samples that will help you learn the new features of the 2007 Microsoft Office system programs, servers, services, tools, and technologies. Each sample is a short walkthrough that showcases one feature. By integrating several of these code samples together, you can start building [Office Business Applications](http://go.microsoft.com/?linkid=6559427) on the 2007 Microsoft Office platform.

## Dev Tools

* [Visual Studio Products](http://msdn2.microsoft.com/en-us/vstudio/aa700919.aspx)
	+ Microsoft Visual Studio Team System 2005
	+ Microsoft Visual Studio 2005 Professional Edition
	+ Microsoft Visual Studio 2005 Tools for the 2007 Microsoft Office System
	+ Microsoft Visual Studio Standard Edition
	+ Microsoft Visual SourceSafe 2005
	+ Microsoft Visual Studio Express Editions
	+ Microsoft Visual Studio 2005 Tools for Applications
	+ Subscriptions
* [Windows SharePoint Services 3.0 Tools: Visual Studio 2005 Extensions](http://www.microsoft.com/downloads/details.aspx?familyid=19F21E5E-B715-4F0C-B959-8C6DCBDC1057&displaylang=en#Overview)

Visual Studio 2005 Project Templates

* + Web Part
	+ Team Site Definition
	+ Blank Site Definition
	+ List Definition

Visual Studio 2005 Item Templates (items that can be added into an existing project)

* + Web Part
	+ Custom Field
	+ List Definition (with optional Event Receiver)
	+ Content Type (with optional Event Receiver)
	+ Module

Review the download details page for important additional information.

* [Internet Explorer Developer Toolbar](http://www.microsoft.com/downloads/details.aspx?familyid=e59c3964-672d-4511-bb3e-2d5e1db91038&displaylang=en)

Provides a variety of tools for quickly creating, understanding, and troubleshooting Web pages.

* [Windows SharePoint Services 3.0 Sample: Example Master Pages](http://www.microsoft.com/downloads/details.aspx?familyid=7C05CA44-869A-463B-84D7-57B053711A96&displaylang=en)

Four sample master page sets compatible with the Application Templates for Windows SharePoint Services 3.0 that show some of the customization options master pages provide.

* [Windows SharePoint Services 3.0 Application Templates: All Templates](http://www.microsoft.com/downloads/details.aspx?familyid=5807B5EF-57A1-47CB-8666-78C1363F127D&displaylang=en)

Forty of the application templates for Windows SharePoint Services 3.0 plus the Application Template Core solution, which is required to deploy server administration templates.

## Packaging Tools

* [Windows SharePoint Services 3.0 Tools: Visual Studio 2005 Extensions](http://www.microsoft.com/downloads/details.aspx?familyid=19F21E5E-B715-4F0C-B959-8C6DCBDC1057&displaylang=en#Overview)

SharePoint Solution Generator

* + Stand-alone program that generates a Site Definition project from an existing SharePoint site. The program enables developers to use the browser and Office SharePoint Designer to customize the content of their sites before creating code by using Visual Studio.
* MakeCAB
	+ <http://search.live.com/results.aspx?q=site%3Amicrosoft.com++sharepoint+2007+makecab&src=IE-SearchBox>
	+ [Solution Deployment with SharePoint 2007](http://msdn.microsoft.com/msdnmag/issues/07/08/OfficeSpace/)
	+ [Office Visual How To - Creating a Solution Package in Windows SharePoint Services 3.0](http://msdn2.microsoft.com/en-us/library/bb466225.aspx)
	+ [Development Tools and Techniques for Working with Code in Windows SharePoint Services 3.0 (Part 2 of 2)](http://msdn2.microsoft.com/en-us/library/bb530301.aspx)
* [Content Migration API in Action](http://blogs.msdn.com/ronalus/archive/2007/08/10/content-migration-api-in-action.aspx) demo tool

When running the application, look at the session log on the right side of the screen. It lists the properties used in the deployment and may help you develop your own tools.

## SOLUTION Framework

* [SOLUTIONS and Web Part Packages](http://msdn2.microsoft.com/en-us/library/ms413687.aspx)

This section of the Windows SharePoint Services 3.0 SDK provides information about the Windows SharePoint Services solution framework. Use solutions to package and deploy custom Features, site definitions, templates, Web Parts, and assemblies.

* [Creating a Windows SharePoint Services 3.0 Web Part Using Visual Studio 2005 Extensions](http://msdn2.microsoft.com/en-us/library/aa973249.aspx)

Learn to use the project templates included in Microsoft Visual Studio 2005 Extensions for Windows SharePoint Services 3.0 to speed your Web Part development.

* [How SOLUTION deployment has changed development with SharePoint technologies](http://blogs.msdn.com/cjohnson/archive/2006/09/11/749105.aspx)

The Windows SharePoint Services solution framework provides a way to bundle together all of the components for extending SharePoint in a new file called a solution file (a CAB-based format with a WSP extension). A solution is a deployable, reusable package that can contain a set of features and site definitions, templates, Web Parts, and assemblies that you can apply to a site, and individually enable or disable.

The Solution Framework also deploys the solution to **ALL** front-end Web servers in the farm without the administrator performing this on each computer manually

* [SOLUTION deployment](http://www.andrewconnell.com/blog/articles/UsingVisualStudioAndMsBuildToCreateWssSolutions.aspx)

Configure your Visual Studio 2005 project, using MSBuild and MakeCAB.exe, to create the \*.WSP file for you.

* See Arjun Ohri in the [Microsoft Office System Developers Conference](http://msdn2.microsoft.com/en-us/office/aa905380.aspx), in Windows SharePoint Services 2007: Site Definitions, Features, and SOLUTION Deployment (WS304).

The Microsoft Office System Developers Conference 2006 featured more than 60 breakout sessions organized in eight technical tracks. From the new servers to the OpenXML Formats, to new client user interfaces and extensibility tools, there's a lot to tell you.

## Bin or Global Assembly Cache

* [Microsoft Windows SharePoint Services and Code Access Security](http://msdn2.microsoft.com/en-us/library/ms916855.aspx) (SharePoint 2003 technical articles)

Learn how to implement policies for code access security for SharePoint Products and Technologies and how to customize default security settings for Windows SharePoint Services. This document also answers some of the most commonly asked questions about code access security and its applicability to Windows SharePoint Services.

* [Creating a Basic Web Part](http://msdn2.microsoft.com/en-us/library/ms948909.aspx) (SharePoint 2003 technical articles)

This programming task provides the steps for creating a basic custom Web Part that you can add to your Web Part Page. It is a very simple Web Part that allows you to change the Web Part's **Title** property. The **Title** property is a Web Part base class property that sets the text in the part's title bar.

* [A Developer's Introduction to Web Parts](http://msdn2.microsoft.com/en-us/library/ms916848.aspx) (Microsoft Office SharePoint Portal Server 2003 technical articles)

Learn what Web Parts are and how to create them. Developers can build Web Parts as ASP.NET custom controls. Administrators can install Web Parts on any site based on Windows SharePoint Services. Users can add Web Parts to pages by dragging and dropping in a browser, and they can personalize them by setting properties. Web Parts can connect to other Web Parts using standard interfaces.

## Features

* [Working with Features](http://msdn2.microsoft.com/en-us/library/ms460318.aspx)

Features reduce the complexity involved in making simple site customizations, and are robust when upgrades are applied to a deployment. Features eliminate the need to copy large chunks of code to change simple functionality. Features thus reduce versioning and inconsistency issues that may arise among front-end Web servers. Features make it easier to activate or deactivate functionality in the course of a deployment, and administrators can easily transform the template or definition of a site by simply toggling a particular Feature on or off in the user interface.

* [Troubleshooting Feature and Site Definition Development](http://msdn2.microsoft.com/en-us/library/aa543273.aspx)

This troubleshooting topic can help you determine a cause if you find that the changes you intend to make through a definition or Feature do not take effect.

* [Todd Baginski’s blog SharePoint Feature Manager](http://www.sharepointblogs.com/files/folders/tbaginski/entry2449.aspx)

This application allows you to easily manage Features on your SharePoint server farm, without using the command line.

## Authoring and customization

* For a list of supported browsers, see:
	+ [Plan browser support (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/ff6c5b8c-59bd-4079-8f0b-de4f8b4e0a861033.mspx?mfr=true)
	+ [Plan browser support (Windows SharePoint Services)](http://technet2.microsoft.com/windowsserver/WSS/en/library/7dd4fd50-6ede-4d21-a5d5-87b4c4d493161033.mspx?mfr=true)
* [Smart Client Authoring](http://blogs.msdn.com/ecm/archive/2006/06/13/629525.aspx)

Smart Client Authoring (SCA) allows you to take a document stored in a SharePoint library and convert it into a Web page. Authors can easily do this on any document in SharePoint Products and Technologies, provided it’s saved in a supported format.

## Content Deployment / Migration

#### Content Deployment

* [Deploying Content between Servers](http://msdn2.microsoft.com/en-us/library/ms549024.aspx)

MOSS 2007 provides a rich deployment user interface. Most deployment scenarios can be accomplished by an IT professional through the user interface, without the need for scripts. However, you can still use the object model to handle other scenarios, such as deploying content between servers that are not on the same network or writing scripts to automate common tasks. This topic provides an overview of the MOSS 2007 content deployment feature and gives developers the background and resources necessary to build and implement custom deployment solutions.

* [How to: Customize Deployment for Disconnected Scenarios](http://msdn2.microsoft.com/en-us/library/aa981161.aspx)

Content deployment works well when a clear connection between the source farm and the destination farm is always available. However, a reliable, functional connection between a source authoring farm and a destination production farm is not always available. For example, geographic barriers, firewalls that prevent access, or network outages are all situations in which using a network connection to transport an export package to a destination production farm is not reliable. In these situations, Microsoft recommends that you work with the MOSS 2007 object model and the Windows SharePoint Services 3.0 Content Migration API to programmatically complete the export and import steps of content migration and find an alternative way to transport the export package to the destination production farm and then run the custom import code.

* [Plan content deployment](http://technet2.microsoft.com/Office/en-us/library/edcdacca-8013-460e-95a0-d2b83b6cc7ef1033.mspx?mfr=true)

Content deployment copies content from a source MOSS 2007 site collection to a destination site collection. The entire source site collection can be copied, or a subset of sites can be copied. In either case, content deployment is incremental by default, deploying only changed pages and related assets (such as images).

* [Design content deployment topology](http://technet2.microsoft.com/Office/en-us/library/1d6d6040-6cbb-4685-a40e-1e9086d426831033.mspx?mfr=true)

Content deployment copies content from a source MOSS 2007 site collection to a destination site collection, either based on a schedule or manually. This article describes elements of topologies designed for content deployment and shows typical content deployment topologies. For an overview of content deployment using MOSS, see [Plan content deployment](http://technet2.microsoft.com/Office/en-us/library/edcdacca-8013-460e-95a0-d2b83b6cc7ef1033.mspx?mfr=true).

* [Plan content approval and scheduling](http://technet2.microsoft.com/Office/en-us/library/b43e9421-66b8-4cfc-ba06-f772ae7420e81033.mspx?mfr=true)

In addition to planning how much control you want users to have over altering the appearance of your MOSS 2007 Web site, you also need to plan how much governance you want content contributors to have over your site content. Governance refers to the rules and processes that you want to establish for your site. For example, you may want users to impose restrictions and functionality when authors create content. You have the option of giving users no control, simple moderation, or the ability to start a workflow after they submit content. You can also plan restrictions on where and what type of content an author can place in certain areas of your site.

* [Enable access for end users (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/1e7884ca-c514-4d08-aba7-5ce0d8ad12381033.mspx?mfr=true)

After you create your site collection and populate it with content, you are ready to grant access to end users. This article helps you configure administrative and user permissions for a site collection. You can also configure permissions for the following securable objects within a site collection: site, list, library, folder, document, or item. For more information about assigning permissions for different securable objects within a site collection, see [Plan site security (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/1e7884ca-c514-4d08-aba7-5ce0d8ad12381033.mspx?mfr=true).

* [Content Migration Overview](http://msdn2.microsoft.com/en-us/library/ms453426.aspx)

The Content Migration APIs provide a simple but flexible solution for migrating content between Windows SharePoint Services Web sites. You can export the content related to a Windows SharePoint Services Web site, along with any dependencies (for example, security, roles, versioning, workflow, and other metadata), into single or multiple XML-formatted files called content migration packages. On import to the destination Web site, the packaged data is extracted and interpreted. You can also save the packages to a file server before migrating to a different server.

* [The official blog of the SharePoint Product Group - Content Deployment](http://blogs.msdn.com/sharepoint/archive/2006/05/02/588140.aspx)

Content Deployment takes care of transporting the content across the wire for you and instantiates the remote import as well.

* [Migrate content or sites after upgrade (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/16a7e571-3531-4a4e-baa7-f348a9f9d1d11033.mspx?mfr=true)

After you have completed the upgrade process, you can redistribute content or sites as needed to fit your new environment. It is easiest to move content or sites before you open the sites to users again, so that they do not have to experience more than one outage window.

#### Windows SharePoint Services 3.0 Content Migration API

#### Object Model

* [Microsoft.SharePoint.Deployment Namespace](http://msdn2.microsoft.com/en-us/library/microsoft.sharepoint.deployment.aspx)

* [SPExportSettings Properties](http://msdn2.microsoft.com/en-us/library/microsoft.sharepoint.deployment.spexportsettings_properties.aspx)

* [SPImportSettings Properties](http://msdn2.microsoft.com/en-us/library/microsoft.sharepoint.deployment.spimportsettings_properties.aspx)

* [SPExport Class](http://msdn2.microsoft.com/en-us/library/microsoft.sharepoint.deployment.spexport.aspx)

* [SPImport Class](http://msdn2.microsoft.com/en-us/library/microsoft.sharepoint.deployment.spimportsettings.aspx)
* [How to: Deploy Content Between Servers](http://msdn2.microsoft.com/en-us/library/ms496233.aspx)
* [How to: Migrate a Web site from one location to another](http://msdn2.microsoft.com/en-us/library/ms438819.aspx)

#### Windows SharePoint Services 3.0 Content Migration Web Service

* [Sites Web Service](http://msdn2.microsoft.com/en-us/library/sites.sites_methods.aspx)
* [Sites.ExportWeb Method (Sites)](http://msdn2.microsoft.com/en-us/library/sites.sites.exportweb.aspx)
* [Sites.ImportWeb Method (Sites)](http://msdn2.microsoft.com/en-us/library/sites.sites.importweb.aspx)

#### STSADM

* [Export: Stsadm operation (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/65788bb9-0345-42c8-a216-e99e558b173d1033.mspx?mfr=true)

* [Import: Stsadm operation (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/1178aa35-20b1-45b0-bcb6-4249aa34ea481033.mspx?mfr=true)

## Team Development

* [Team-Based Development in Microsoft Office SharePoint Server 2007](http://msdn2.microsoft.com/en-us/library/bb428899.aspx)

Learn to properly conduct team-environment development of Microsoft Office SharePoint Server 2007 sites and assemblies (Web Parts, site templates, custom list templates), as well as develop Microsoft Office SharePoint Designer artifacts (master pages, workflows, CSS sheets).

* [Conduct Team-Based Development of Windows SharePoint Services and SharePoint Portal Server Applications](http://msdn2.microsoft.com/en-us/library/aa505325.aspx) (SharePoint 2003)

Learn to conduct team-based development of Windows SharePoint Services and Microsoft Office SharePoint Portal Server 2003–based applications. These applications can include customized Web Part Pages, Web Parts, supporting Web applications, and Web services.

* [Walkthrough: Creating a Custom Enterprise Search Web Part](http://msdn2.microsoft.com/en-us/library/ms551453.aspx)

With Enterprise Search in Microsoft Office SharePoint Server 2007, you can customize the look and functionality of the Search Center pages and Web Parts from the browser. However, if you want to make customizations that are not possible through the browser, you can create custom Web Parts that use the Query object model to execute queries against the search component.

In this walkthrough, you'll create a custom search Web Part and add it to your site. The Web Part described in this walkthrough provides very basic search functionality.

* [Setting Up a Development Environment for the 2007 Microsoft Office System](http://msdn2.microsoft.com/en-us/library/bb330848.aspx)

Learn how to set up a development environment for the 2007 Microsoft Office system. Create sample projects that demonstrate the types of application solutions that you can create using this system.

## Testing, Source Control and MSF

#### Testing

* [Microsoft Best Practices Analyzer for Windows SharePoint Services 3.0 and the 2007 Microsoft Office System](http://www.microsoft.com/downloads/details.aspx?familyid=CB944B27-9D6B-4A1F-B3E1-778EFDA07DF8&displaylang=en)

Use this tool to create detailed reports to help administrators achieve greater performance, scalability, and uptime.

* [SharePoint 2007 Test Data Population Tool](http://www.codeplex.com/sptdatapop) and the article: [Tools for performance and capacity planning (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/301ed832-95da-4251-b266-7be6288f7ea01033.mspx?mfr=true)

This tool is a capacity planning and performance testing tool that populates data for testing SharePoint deployments. The SharePoint 2007 Test Data Population Tool is available as a command-line executable program that extracts information about how to populate the server from an XML configuration file, and calls Microsoft .NET Framework assembly WSSDWLib.dll. The tool is explained in the article [Tools for performance and capacity planning (Office SharePoint Server)](http://technet2.microsoft.com/Office/en-us/library/301ed832-95da-4251-b266-7be6288f7ea01033.mspx?mfr=true).

* [Visual Studio – Testing](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vsent7/html/vxoriTestingOptimizing.asp)

Testing is the process of examining an application to ensure it fulfills the requirements for which it was designed and meets quality expectations. More importantly, testing ensures the application meets customer expectations.

* [Black Box and White Box Testing for Application Blocks](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/MTF_ch06.asp)

Detailed overview of structural testing (white box testing) and output testing (black box testing) an application block, addressing subjects such as code path profiling, instrumentation, and testing external interfaces.

* [Testing Methodologies](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/MTF_ch02.asp)

An introduction to various test methodologies with an example around test-driven development.

* [Design Review for Application Blocks](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/MTF_ch04.asp)

Describes how to perform design reviews for an application block for functional, deployment, performance, scalability, security, and globalization considerations.

* [Checklist for Testing SharePoint Web Parts](http://msdn2.microsoft.com/en-us/library/ms916830.aspx)

Use this checklist to assist with the deployment and maintenance of SharePoint Products and Technologies Web Parts.

* [Best Practices for Developing Web Parts for SharePoint Products and Technologies](http://msdn2.microsoft.com/en-us/library/ms916817.aspx)

Using the programming model behind Windows SharePoint Services, you can create your own Web Parts that provide new functionality to enhance your Web Part Pages. Learn best practices to improve performance and usability of Web Parts, and ways to create Web Parts that integrate well with other components of a Web Part Page.

* The [Fiddler](http://www.fiddlertool.com/fiddler/)

Fiddler is an HTTP debugging proxy which logs all HTTP traffic between your computer and the Internet. Fiddler allows you to inspect all HTTP Traffic, set breakpoints, and "fiddle" with incoming or outgoing data. Fiddler includes a powerful event-based scripting subsystem, and can be extended using any Microsoft .NET Framework language.

Fiddler is freeware and can debug traffic from virtually any application, including Internet Explorer, Mozilla Firefox, Opera, and others.

#### Source Control and Microsoft Solutions Framework

* [Building the Avanade Software Lifecycle Platform using Visual Studio 2005 Team System](http://msdn2.microsoft.com/en-us/vstudio/aa718955.aspx)

Microsoft Visual Studio Team System 2005 is an extensible lifecycle tools platform that helps software teams collaborate to reduce the complexity of delivering solutions. The purpose of this paper is to describe how Avanade is building a value-added software lifecycle platform using Visual Studio Team System. The intended audience for this paper is software developers and project managers.

* [Integrated Software Development at Microsoft Using Visual Studio 2005 Team System](http://www.microsoft.com/technet/itshowcase/content/msvststcs.mspx)

Microsoft Visual Studio Team System 2005 provides an integrated software development environment that enables the Microsoft eBusiness Integration Services business-to-business IT development group to incorporate the complete Microsoft IT Software Development Life Cycle (SDLC) process in one solution.

* [MSF Sample Project Lifecycle Deliverables](http://www.microsoft.com/downloads/details.aspx?FamilyId=9D2016AD-6F8A-47F5-84FA-BEC389DB18C1&displaylang=en)

Download sample project lifecycle deliverables that work with the Microsoft Solutions Framework (MSF) lifecycle version 3.0.

## Patterns and Practices

* [Logging Application Block](http://msdn2.microsoft.com/en-us/library/aa480464.aspx)

This page provides an overview of the Enterprise Library Logging Application Block. An application block is reusable and extensible source code-based guidance that simplifies development of common logging functionality in Microsoft .NET Framework applications.

* [Microsoft patterns & practices](http://msdn2.microsoft.com/en-us/library/ms998572.aspx)

This page contains deep technical guidance and tested source code based on real-world experience. The technical guidance is created, reviewed, and approved by Microsoft architects, product teams, consultants, product support engineers, and by Microsoft partners and customers. The result is a thoroughly engineered and tested set of recommendations that you can follow with confidence when building your applications.

# Credits and Thanks To

* Arjun Ohri , Program Manager, Windows SharePoint Services
* Joel Oleson, Senior Product Manager
* Tyler Butler, Program Manager
* Jackie Bodine, Consultant
* Mike Ammerlaan, Program Manager, Windows SharePoint Services
* Chris Johnson, Program Manager
* All authors of technical articles
* Yaron Ben Shalom, CTO, Tvuna-Millenium
* Robert L. Bogue, MS MVP
* Andrew Connell, MVP, Microsoft Office SharePoint Server
* Todd S. Baginski, Vice President, SharePoint Experts
* Gary A. Bushey, MOSS MVP

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering this document or the subject matter included in this document. The furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2008 Microsoft Corporation. All rights reserved.

1. Credit to Robert Bogue for the casing notion [↑](#footnote-ref-2)
2. Thanks to Todd Baginski! [↑](#footnote-ref-3)
3. Thanks to Gary Bushey! [↑](#footnote-ref-4)
4. When the pilot environment and production environment have no network connection to internal LAN. This is common in secured organizations and many others. [↑](#footnote-ref-5)