

Module 3: Server Infrastructure and Content Deployment

Date published:

February 2009

Summary:

This paper describes how to plan and design server farms for WCM solutions, and how to manage the deployment of content between multiple environments.

See [Web Content Management Training Modules](http://go.microsoft.com/fwlink/?LinkId=141931) (http://go.microsoft.com/fwlink/?LinkId=141931) for a complete list of the available downloads.

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.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, SharePoint, SQL Server, Visual Studio, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

Module 3: Server Infrastructure and Content Deployment 1

Module 3 Overview 1

Objectives 1

Lesson 1: Server Farms for Web Content Management Solutions 1

Objectives 1

Scaling Out Web Servers and Database Servers 1

This section contains the following topics: 2

Issues Specific to Web Content Management 2

Baseline Topologies 2

Scaling Out a Server Farm 5

Scaling Out Application Servers 6

Dedicated Combined Query Server and Index Server 6

Dedicated Index Server with Combined Web Server and Query Server 7

Dedicated Index Server with Dedicated Query Servers 7

Dedicated Crawl Server 8

Development, Staging, and Production Server Farms 8

Development Environments 9

Test Environments 9

Staging Environments 10

Production Environments 11

Lesson 2: Content Deployment 11

Objectives 12

Features of the Content Deployment System 13

Paths and Jobs 13

Automated Deployment Options and Issues 14

Content Dependencies 14

Exclusions 14

Deploying Content Manually 16

Deploying Content from the Command Line 16

Deploying Content Programmatically 17

Planning the Content Deployment System 18

Planning Permissions 18

Custom Controls and Code 19

Planning Job Schedules 19

Quick Deploy and Page Scheduling 20

Quick Deploy 20

Page Scheduling 20

Site Variations and Content Deployment 21

Variations Overview 21

Publishing Process 22

Variations and Multiple Environment Deployments 22

Review of Module 3 22

Module 3 Overview

Web Content Management (WCM) solutions often consist of multiple server farms to provide isolated development, testing, staging, and production environments. Each of these server farms is configured to meet the requirements of a different set of users and different environmental constraints.

This module describes how to plan and design server farms for WCM solutions, and how to manage the deployment of content between multiple environments.

Objectives

After completing this module, you will be able to:

* Explain the main server farm design considerations for WCM solutions ([Lesson 1](#Lesson1))
* Describe the principal content deployment issues for WCM solutions ([Lesson 2](#Lesson2))

Lesson 1: Server Farms for Web Content Management Solutions

WCM solutions in Microsoft® Office SharePoint® Server 2007 are used in very different ways from other types of solutions such as personal sites or collaboration platforms. WCM solutions are more likely to be Internet-facing and configured to allow anonymous access. The number of users who access your site concurrently may be larger than those of other environments such as a collaboration solution. A much higher proportion of user operations will be read-only. You must consider all of these factors when you plan the server farm topology for a WCM solution.

This lesson describes the principal server farm design issues for WCM solutions in Office SharePoint Server 2007.

Objectives

After completing this lesson, you will be able to:

* [Design small and medium server farms for WCM scenarios](#ScalingOutWebServersandDbServers)
* [Design large server farms for WCM scenarios](#ScalingOutApplicationServers)
* [Explain the benefits of using multiple server farms in a WCM environment](#DevelopmentStagingProductionServerFarms)

Scaling Out Web Servers and Database Servers

When you plan the server farm topology for a WCM solution, you must plan for availability, performance, and capacity in the same way that you would for any Office SharePoint Server 2007 deployment. However, the usage profiles for WCM solutions are different from usage profiles for other Office SharePoint Server solutions, and you must take this into account during the planning phase of your deployment. The following table shows differences between WCM-specific issues and baseline topologies.

|  |  |
| --- | --- |
| **Aspect** | **Details** |
| WCM-specific issues | * Anonymous connections * More read-only operations * Request and query throughput is higher * Effective caching is more critical |
| Baseline topologies | * 1 Web/Application server, 1 Database server * 2 Web/Application servers, 1 Database server * 2 Web/Application servers, 2 Database servers * 2 Web/Application servers, 2 Database servers, Application server |

This section contains the following topics:

* [Issues Specific to Web Content Management](#IssuesSpecifictoWebContentManagemen)
* [Baseline Topologies](#BaselineTopologies)

Issues Specific to Web Content Management

An Internet-facing WCM solution is likely to have the following characteristics:

* A large proportion of user connections will be anonymous.
* Most user operations will be read-only operations.
* On the same hardware, request and search query throughput can be substantially higher for anonymous connections because there is no authentication or authorization overhead.
* In anonymous environments, optimally configured caching can improve performance significantly.
* Other than the search service, requirements for Shared Services Provider (SSP) features such as My Sites, Excel Services, and the Business Data Catalog are likely to be limited.

Baseline Topologies

As a starting point, select the baseline server topology that meets your availability requirements. You can then scale out your server farm to meet your performance and capacity requirements.

Baseline Topologies: Two-Server Farm

The most basic server farm topology is a two-server solution. The following table describes how server roles are allocated.

|  |  |
| --- | --- |
| **Server** | **Roles** |
| 1 | * Web Application * Query server * Index server * Central Administration site host * Any other application server roles |
| 2 | * Database server |

The key points of the two-server solution are as follows:

* This topology does not provide any redundancy. If an availability of more than 99 percent is required, this topology will not provide a satisfactory solution.
* In a WCM solution, you are most likely to encounter a bottleneck at the Web server role. As user numbers increase, a single server may have difficulty handling all user requests and search queries. This will have an adverse effect on response times.

Baseline Topologies: Three-Server Farm

If you want to scale out your server farm to handle more user requests, you can move to a three-server solution. One option that uses three servers but does not fulfill the goal of increasing capacity is to duplicate the Microsoft® SQL Server® role in a SQL cluster; this does not increase throughput because only one SQL host is responding to Office SharePoint Server 2007 requests at any given time. Another three-server option is to add a dedicated application server. However, because WCM usage profiles tend towards large volumes of user requests, the addition of a dedicated application server does not usually increase Web Application role capacity as much as the addition of another Web application server.

Typically, the best three-server option for maximizing WCM user request throughput is to duplicate the Web server role and distribute the application server roles across the two Web servers. The following table describes how server roles are allocated.

|  |  |
| --- | --- |
| **Server** | **Roles** |
| 1 | * Web Application * Query server * Index server |
| 2 | * Web Application * Central Administration site host * Any other application server roles * Database Server |

The key points of this three-server solution are as follows:

* The Query server role is on the same server as the Index server role, so the Query server role cannot be distributed to multiple servers because the index will not be propagated.
* The Index server role cannot be distributed, so Server 1 is configured to perform search crawls at off-peak times.
* The two Web servers are network load-balanced. Incoming user requests are distributed evenly between the two servers. You can consider a more sophisticated load-balancing solution if the application server activities on these servers result in large inequalities in server demands. Off-peak indexing operations can alleviate the inequalities somewhat, but on server 1, both the Web Application and Query roles contend for memory. This means that if query operations are significant and each server receives an equal distribution of user requests, requests sent to Server 1 might have slower responses than requests sent to Server 2.
* The Central Administration site is hosted on at least one server, though it could be hosted on both servers 1 and 2 for redundancy.
* Other application server roles can be distributed between the two Web servers, or allocated to one specific Web server. In a WCM solution, the requirements for application server features such as Excel Calculation Services are likely to be limited.
* The three-server topology with two web servers provides substantially more capacity for concurrent users than the two-server solution. However, from an availability perspective, the Database and Query server roles do not include redundancy.

Baseline Topologies: Four-Server Farm

If you require high availability for your WCM deployment, you can build in redundancy for the Database Server role by moving to a four-server solution. In the same way as the three-server solution, you duplicate the Web Server role and distribute the application server roles across the two Web servers. You also use database mirroring or clustering to build redundancy into the Database Server role. The following table describes how server roles are allocated.

|  |  |
| --- | --- |
| **Server** | **Roles** |
| 1 | * Web Application * Query server * Index server |
| 2 | * Web Application * Central Administration site host * Any other application server roles |
| 3 | * Database Server |
| 4 | * Database Server |

The key points introduced in this four-server solution when compared to the previous three-server solution are as follows:

* Database clustering or mirroring (in Microsoft® SQL Server® 2005) is used to replicate the databases between server 3 and server 4 to provide continuing availability in case of hardware failure.
* The four-server option is the minimum server configuration if you require a high-availability solution for serving page requests.
* There is no redundancy for the Query role.

Baseline Topologies: Five-Server Farm

If you want to distribute the Query server role across multiple Web servers, you must introduce a dedicated application server that includes the Index server role. You cannot install the Query server role on multiple servers until you separate the Query server role from the Index server role, because the index will not be propagated. The following table describes how server roles can be allocated.

|  |  |
| --- | --- |
| **Server** | **Roles** |
| 1 | * Web Application * Query server |
| 2 | * Web Application * Query server |
| 3 | * Index Server * Central Administration site host * Any other application server roles |
| 4 | * Database Server |
| 5 | * Database Server |

When compared to the four-server topology in the previous example, this solution introduces the following new characteristics:

* Redundancy for the Query role: This is the minimum high availability configuration for all services directly involved in responding to user requests.
* Identical service allocations to servers with the Web application role: Simple load-balancing techniques should work well in most scenarios, because the Web servers have the same performance bottlenecks.

The following topic describes the different ways in which you can distribute application server roles.

Scaling Out a Server Farm

In a small-to-medium WCM deployment, you are most likely to require additional Web servers to handle increases in the number of concurrent user requests. Microsoft provides the following guidelines about server role ratios:

* You can add up to eight Web servers for each clustered or mirrored Database server.
* It is recommended that you maintain a ratio of no more than three Web servers for each domain controller. However, this is based entirely on the overall number of authenticated users who use the system.

In an Internet-facing environment with predominantly anonymous access, the authentication traffic sent to and from the domain controller will be substantially reduced when compared to other Office Server 2007 solutions. You are likely to be able to support more Web servers for each domain controller with no adverse effects on performance.

As with all Office SharePoint Server 2007 deployments, your hardware scale-out requirements should be driven by a comprehensive performance monitoring strategy.

Note: For more information about performance and capacity planning, including test cases and sample scenarios, see [Plan for software boundaries (Office SharePoint Server)](http://go.microsoft.com/fwlink/?LinkID=95115&clcid=0x409) (http://go.microsoft.com/fwlink/?LinkID=95115&clcid=0x409) and [Estimate performance and capacity requirements for Internet environments (Office SharePoint Server)](http://go.microsoft.com/fwlink/?LinkId=140143&clcid=0x409) (http://go.microsoft.com/fwlink/?LinkId=140143&clcid=0x409).

Scaling Out Application Servers

Scaling out Web servers and Database servers is relatively straightforward. You can use performance monitoring to identify bottlenecks and then add resources accordingly. For example, if the database is a bottleneck to the overall throughput of the farm, increase Database server resources by either scaling up or scaling out. In larger deployments, however, you must consider how best to distribute Application server roles across your server farm. This can be very complex, especially for search-intensive WCM solutions. At the simplest level, when you scale a WCM environment, you can either configure dedicated Application servers or you can continue to distribute application server roles over increasing numbers of Web servers.

Search is by far the predominant application server role in WCM solutions, so this topic focuses on how you can distribute the Index and Query server roles. Several scenarios are presented, with possible advantages and disadvantages for each scenario. There is no right or wrong server farm topology: you must make decisions on how to scale out your farm based on the results of careful performance monitoring, rather than by following prescriptive topology models.

You can distribute Query and Index server roles in the following main ways:

* [Dedicated combined Query server and Index server](#DedicatedCombinedQueryandIndexServers)
* [Dedicated Index server with combined Web server and Query server](#DedicatedIndexServerwithCombined)
* [Dedicated Index server with dedicated Query servers](#DedicatedIndexServerwithDedicatedQuery)
* [Dedicated Crawl server (two options)](#DedicatedCrawlServer)

Dedicated Combined Query Server and Index Server

In this topology, server roles are distributed as follows:

* Two or more Web servers
* One dedicated Query/Index server
* Clustered or mirrored Database server

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| The Index server role is processor-intensive whereas the Query server role is memory-intensive. Resource contention between the two roles is unlikely to be an issue. | No redundancy for the Query server role |
| Removing the Query server role from the Web servers reduces memory contention on the Web servers. | If the Query role is on the same server as the Index role, you cannot add further query servers because the Index server will not propagate the index. |

Dedicated Index Server with Combined Web Server and Query Server

This is the most common scenario for small or medium deployments. In this topology, server roles are distributed as follows:

* Two or more Web servers, each of which also performs the Query server role
* One dedicated Index server
* Clustered or mirrored Database server

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Builds in redundancy for the Query server role. | Possible memory contention between the Web server role and the Query server role on each computer. This is especially likely in a WCM environment because Web servers will rely more heavily on caching. |
| Reduced network traffic at query time because each Web server has a local copy of the search index. | Increased network traffic at index propagation time because the Index server now has to propagate the index to a separate Query server. |
|  | Can be an inefficient use of hardware for search-intensive solutions. |

Dedicated Index Server with Dedicated Query Servers

In this topology, server roles are distributed as follows:

* Two or more Web servers
* Two or more Query servers
* One dedicated Index server
* Clustered or mirrored Database server

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| No memory contention between Web and Query server roles. | Increases network traffic between Web servers and Query servers, and between the Index server and Query servers. |
| Builds in redundancy for the Query server role. | Most expensive solution in terms of hardware and licensing costs. |
| Can provide high performance in search-intensive environments. |  |

Dedicated Crawl Server

The final design pattern for search architecture to consider in planning your WCM server farm is a dedicated crawl server. In this scenario, you remove one of the servers hosting the Web Application role from the load balancer so that it does not respond to user requests and direct all internal indexing traffic to it. There are general patterns that implement this option.

The most common pattern is to assign a dedicated crawl Web server. This results in decreased front-end Web server load among servers responding to user requests, and higher possible crawl rates for internal farm content. An example topology for this would be as follows:

* Two or more Web servers (load balanced)
* One Web server (not associated with the load-balanced address)
* Two or more Query servers
* One dedicated Index server
* Clustered or mirrored Database server

Another approach is to assign the Web Application role to the Index server and make that the dedicated crawl server. Here, it crawls itself; although this leads to increased CPU contention, it also has vastly decreased network contention for internal content crawls. An example topology for this would be as follows:

* Two or more Web servers (load balanced)
* Two or more Query servers
* One combined Index/Web server (not associated with the load-balanced address)
* Clustered or mirrored Database server

As with a decision to collapse or separate Web and Query roles, if you are considering a dedicated crawl server, there’s no right or wrong answer as to whether you should use a pure crawl server or a combined Index/crawl server. For example, if you are crawling far more external content than internal farm content, the combined Index/crawl server can still encounter network bottlenecks. Model your environment usage, test different topologies, and see what works best for your specific conditions.

Development, Staging, and Production Server Farms

For large Office SharePoint Server 2007 deployments, many organizations use multiple Web farms or multiple environments to isolate their development, test, and staging processes from their live environment. If you use multiple environments, you must decide whether to provide logical isolation by using separate Web applications and application pools for each environment, or physical isolation by deploying each environment on a separate Web farm. In most circumstances, you should do the following:

* + [Deploy development environments](#DevelopmentEnvironments) on a separate Web farm on a separate network. Although you could provide security isolation by using separate Web applications and application pools, the use of a separate Web farm for your development environment protects your live environment from unnecessary risk.
  + [Deploy test environments](#TestEnvironments) on a separate Web farm because you typically use your test environment to assess hardware requirements for a simulated deployment.
  + [Deploy staging environments](#StagingEnvironments) on the same Web farm as your production environment if you have sufficient capacity (on separate Web applications). For larger deployments, you might have to deploy your staging environment on separate hardware.
  + [Deploy production environments](#ProductionEnvironments).

Development Environments

A development environment enables developers to write code and test custom Office SharePoint solutions such as Web Parts or other assemblies. The development environment should be an isolated deployment of Office SharePoint Server 2007 where your developers can change settings and test solutions without compromising the security or functionality of your publishing environment.

Regardless of whether you use physical computers or a virtualization solution, you should make your development environment as close to your publishing environment as possible. In particular, you should do the following:

* Use the same network topology as your publishing environment.
* Ensure that software versions are identical to those in your publishing environment.
* Ensure that the network and Windows® SharePoint® Services security settings match those in your publishing environment.

To aid the debugging process, your developers are likely to change the security policy trust level in the Web.config file from **WSS\_Minimal** or **WSS\_Medium** to **Full**. Ensure that custom solutions are tested against the trust level you use in the publishing environment before the solution leaves the development environment.

Performance and Security: Your developers should use the FxCop code analysis tool to analyze custom Web Parts and assemblies before they leave the development environment. FxCop tests code assemblies for conformance to the Microsoft .NET Framework design guidelines, and provides recommendations for performance, security, and other areas.

You can obtain the FxCop tool from the [Microsoft FxCop download page](http://go.microsoft.com/fwlink/?LinkId=140144&clcid=0x409) (http://go.microsoft.com/fwlink/?LinkId=140144&clcid=0x409).

Test Environments

You can use an Office SharePoint Server 2007 test environment to test farm layouts, server configurations, and content taxonomies to help you to plan performance and capacity for your Windows SharePoint Services deployment. For example, you might want to evaluate whether a new Web server will improve the user response time of your deployment. You can use the Office SharePoint Server 2007 Test Data Load Tool to simulate the distribution of sites and content across your deployment and to test how it performs under different loads.

Your test environment should be an isolated network that represents your publishing environment infrastructure. The server processors, memory, disk, and network card specifications should match the specifications that you intend to use in your publishing environment. You can then vary individual factors and derive an accurate estimate of how that variation will affect performance in your publishing environment. The following table shows some of the areas that can affect performance.

|  |  |
| --- | --- |
| **Area** | **Considerations** |
| Farm configuration | * Server specifications * Number of load-balanced front-end Web servers * Number of Database servers * Clustering/mirroring configuration * Dedicated Search/Index server |
| Network | * Network interface cards (1 Gbps versus 100 Mbps) * Switches and routers * Number of domain controllers * Network utilization |
| Security | * Authentication method * Server-to-server encryption * Client-to-server encryption |
| Content | * Content capacity * Content taxonomy |
| Users | * Number of users * User concurrency * User behavior |

You can use the Office SharePoint Server 2007 Test Data Load Tool (MOSSDW.exe) and the Windows SharePoint Services Test Data Load Tool (WSSDW.exe) to populate your servers with Windows SharePoint Services and Office SharePoint Server test data entities that simulate the taxonomy of your published environment. Use these in conjunction with load testing and performance-monitoring tools such as Microsoft® Visual Studio® Team Test to evaluate a deployment scenario before you deploy to the published environment.

Note: The Office SharePoint Server 2007 and Windows SharePoint Services test data load tools are community software. They were not developed by Microsoft and are not supported by Microsoft. You can obtain the tools from the [CodePlex Web site](http://go.microsoft.com/fwlink/?LinkID=92678&clcid=0x409) (http://go.microsoft.com/fwlink/?LinkID=92678&clcid=0x409)

Staging Environments

Office SharePoint Server 2007 staging environments are often used in large deployments and in customer-facing scenarios such as extranet or Internet deployments. Staging environments enable you to preview, or “stage,” a deployment in a live environment before you roll it out to a wider audience.

The staging environment should be a live Office SharePoint Server 2007 deployment that is accessible to a finite number of users, typically those who are responsible for the usability testing or approval. You should deploy the staging environment and the published environment on separate SharePoint Web applications. Depending on the scale of your deployment, the staging environment could also be a separate server farm, or it could share physical hardware with your published environment.

It is important to distinguish between the content approval workflow processes that are built into Office SharePoint Server 2007 and the distinct staging and published environments that are described here. If you want to make changes to individual site designs or layouts, or you need to implement approval processes for new content before you distribute it to a wider audience, you generally use content approval workflows in your published environment. You can enable approval workflows on a site-by-site basis. For example:

* + - A Web designer changes a number of content pages on a SharePoint site and submits his or her work for approval.
    - When site visitors and members access the SharePoint site, they continue to see the last published version of the content pages.
    - When site owners or other authorized users access the site, they see the new draft version of the content pages on the site.
    - When a site owner or authorized user approves the new version, these pages become the published version and are accessible to all users.

The staging environment is more suitable for changes on a larger scale. For example, you might want to evaluate a new site taxonomy, conduct usability testing for a new custom assembly, or preview revised branding or new features.

Production Environments

Your production, or publishing, environment is the live Office Server 2007 deployment that you expose to all users. These users could be internal employees, partners, or customers. Authentication, authorization, and IRM become particularly important considerations when you expose your content to large numbers of users or anonymous users.

If your production environment and your staging environment are deployed on separate server farms, remember that you may be able to use inter-farm shared services to provide the search and indexing service to both farms from a single Shared Services Provider (SSP).

Lesson 2: Content Deployment

If you want to migrate content from a staging environment to a production environment, you must plan how you will manage the content deployment process. With Office SharePoint Server 2007, you can deploy content between environments in several different ways. You must understand the benefits and drawbacks of each approach before you can develop a migration strategy that meets the needs of your organization.

This lesson describes the features of the Office SharePoint Server 2007 deployment system and the different ways that you can use to deploy content.

Note: There are several content deployment updates included in both the Infrastructure Update for Microsoft Office Servers and the Infrastructure Update for Windows SharePoint Services 3.0. These updates deliver performance improvements address many of the top customer reported issues with content deployment. You can read more about the infrastructure updates at the [Microsoft Enterprise Content Management (ECM) Team Blog](http://go.microsoft.com/fwlink/?LinkId=140251&clcid=0x409) (http://go.microsoft.com/fwlink/?LinkId=140251&clcid=0x409) .

Objectives

After completing this lesson, you will be able to:

* + [Explain the features of the content deployment system](#FeaturesoftheContentDeploymentSystem)
  + [Describe how to deploy content manually](#DeployingContentManually)
  + [Describe the principal planning issues for the content deployment system](#PlanningContentDeploymentSystem)
  + [Explain the Quick Deploy and page scheduling features](#QuickDeployandPageScheduling)

Features of the Content Deployment System

When you use multiple server farms or environments in a WCM solution, you must develop a strategy for migrating content between environments. Typically, you migrate content from your staging environment to your production environment.

In most circumstances, you can use features of the Office SharePoint Server 2007 content deployment system to automate the migration of content between environments. This topic describes these features. In the next topic, you will see when it is more appropriate to deploy content manually. This section contains the following topics:

* [Paths and Jobs](#PathsandJobs)
* [Automated Deployment Options and Issues](#AutomatedDeploymentOptionsandIssues)
* [Content Dependencies](#ContentDependencies)
* [Exclusions](#Exclusions)

Paths and Jobs

In Office SharePoint Server 2007, paths and jobs provide the principal mechanism for automated content deployment.

A path defines a connection between a source site collection and a destination site collection. The source and destination site collections can be on the same server farm, or in separate server farms. However, the site collections must be in separate content databases (placing them in separate Web applications guarantees this); otherwise, the migration will fail due to conflicts in the content database. When you create a path, you specify all the authentication information that Office SharePoint Server 2007 requires to enable communication between the farms or Web applications. You also specify the level of security information that you want to migrate with your content. The following table shows the levels of security information from which you can select.

|  |  |  |
| --- | --- | --- |
| **Level** | **Details** | **Scenario** |
| All | Users, role definitions, and access control lists that map users and role definitions to content are migrated. | If you move content from an authoring environment to a staging environment, you can require users to have the same permissions in both environments. |
| Role Definitions Only | Role definitions and access control lists that map role definitions to content are migrated. You must manually populate the roles with users on the destination farm. | If you move content from a staging environment to a production environment, you may want the same role definitions, but a more restricted membership for each role. |
| None | No security information is migrated. | If you move content from a staging environment to a production environment, you might want security in the production environment to be configured completely independently. |

A job defines a migration schedule for a particular path. When you create a job, you specify a path, a schedule, and the sites from the source site collection that you want to migrate. By default, the job migrates the entire source site collection to the destination site collection.

Automated Deployment Options and Issues

When you migrate content between environments, only the most recent published (major) version and the most recent draft (minor) version of each page are copied from the source site collection to the destination site collection. The access privileges for the content in the destination site collection match those in the source site collection. In other words, users will be able to view draft versions in the production environment only if they have permissions to view draft versions in the staging environment.

By default, a deployment job will only copy new or modified content to the destination site collection. You can configure the job to migrate all content, but this will have a heavier impact on performance during the migration process and should not be necessary under most circumstances.

Before you schedule deployment jobs, you must configure the destination farm to accept incoming deployment jobs. In most circumstances, you should require that the connection between the source farm and the destination farm is encrypted by using HTTPS. This is the default setting. Note that the destination site collection must initially be based on the Blank Site template.

Content Dependencies

When you deploy content from the staging environment to the production environment, the content on the staging environment may require particular resources, such as page layouts, images, or updated master pages, that are unavailable in the production environment. The deployment job automatically identifies any content dependencies, packages the necessary resources with the updated content, and transfers the resources along with the content to the production environment. As such, you do not need to consider content dependencies within the site collection when you schedule content migration.

Exclusions

The content deployment system in Office SharePoint Server 2007, together with the content deployment **Stsadm** commands discussed in the next topic, are designed for Web publishing scenarios. You use the content deployment system to migrate content, together with any images, page layouts, style sheets, and master pages that are required by your content, from one Web application to another.

The content deployment system is not designed to clone the infrastructure and configuration of your Web application. For example, the content deployment system will not migrate the following:

* + Assemblies, including custom Web Parts or workflows
  + Web.config files, or modifications to Web.config files
  + SharePoint features and solutions

These exclusions are by design. If your organization needs to deploy content in separate and isolated environments, this indicates that each environment has distinct requirements and should be configured individually. If you have multiple environments with identical infrastructure and configuration requirements, you should consider why you need more than one environment in the first place.

Deploying Content Manually

Automated deployment usually provides the most efficient and convenient solution for migrating content between Office SharePoint Server 2007 environments. In some circumstances, however, you might require more granular control over how content is deployed. In this case, you can use the Stsadm command-line utility. Alternatively, you can extensively customize the deployment process by using the Content Deployment and Migration application programming interface (API).

The following table displays deployment methods and details.

|  |  |
| --- | --- |
| **Method** | **Details** |
| Automated deployment | * Most convenient solution for most migration * Lacks flexibility in some areas |
| Command line | * Stsadm **export** and **import** * More flexibility over versioning * Less flexibility over content |
| Custom development | * Content Deployment and Migration API * Complete flexibility for custom solutions * Requires development effort |

This section contains the following topics:

* [Deploying Content from the Command Line](#DeployingContentfromCommandLine)
* [Deploying Content Programmatically](#DeployingContentProgrammatically)

Deploying Content from the Command Line

You can use the Stsadm **export** and **import** operations to migrate content from one environment to another. The export operation copies an entire site collection to a file. The import operation copies content from the exported file to a site collection.

One of the major advantages of the **export** operation is that you can specify the level of versioning that you want to include in the migrated content. The following table shows the values that the **versions** parameter accepts.

|  |  |
| --- | --- |
| **Value** | **Details** |
| 1 | Exports the last major version of each file and list item. This is the default value if you do not include the **versions** parameter. |
| 2 | Exports the most recent version of each file and list item, regardless of whether it is a major or a minor version. |
| 3 | Exports the last major version and the last minor version of each file and list item. |
| 4 | Exports all major and minor versions of files and list items. |

For example, if you want to migrate every version of your files and list items, together with security settings for each user, you can use the following **Stsadm** command:

stsadm –o export –url http://staging.litwareinc.com –filename backup.dat –includeusersecurity –versions 4

Similarly, when you use the **import** operation, you can specify how existing files are updated when new versions are migrated in. The following table shows the values that the **updateversions** parameter accepts.

|  |  |
| --- | --- |
| **Value** | **Details** |
| 1 | Adds new versions to the existing file. This is the default value if you do not include the **updateversions** parameter. |
| 2 | Overwrites the existing file and all existing versions. |
| 3 | Ignores the file if it already exists at the destination. |

For example, if you want to overwrite all existing versions of each file at the destination when you migrate content, you can use the following **Stsadm** command.

stsadm –o import –url http://litwareinc.com –filename backup.dat -includeusersecurity –updateversions 2

Note: For more information about using the Stsadm tool to migrate content, please see [Stsadm command-line tool (Office SharePoint Server)](http://go.microsoft.com/fwlink/?LinkId=140255&clcid=0x409) (http://go.microsoft.com/fwlink/?LinkId=140255&clcid=0x409)

You can also use the Stsadm **backup** and **restore** operations to copy an entire site collection from one environment to another. Some administrators prefer this approach, particularly when they first create the production site collection from the staging site collection. The backup and restore approach effectively creates a duplicate of the staging site collection in the production environment. Although this approach can simplify the migration process, it offers no control over what content is deployed and to what extent security information is replicated.

Deploying Content Programmatically

If you require full, customizable control over your content deployment process, you can use the Content Deployment and Migration API (formerly known as PRIME) to create your own deployment solution. The manual Stsadm **import** and **export** operations and the automated path and job features both rely on the functionality provided by the Content Deployment and Migration API.

You can use the Content Deployment and Migration API to control all aspects of the deployment process. You can migrate everything from an entire site collection to an individual document between your environments, with full control over how dependencies, compression, and security settings are managed. You could also create custom solutions that automate the migration of content based on workflows, or by creating event receivers for document library, list, or site events.

Implementation details for the Content Deployment and Migration API are beyond the scope of this topic. However, the following C# code example shows a simple example of how you can export an entire site collection. This performs an identical export to the Stsadm **export** operation described earlier:

SPExportSettings mySettings = new SPExportSettings();

mySettings.SiteUrl = "http://staging.litwareinc.com";

mySettings.FileLocation = @"C:\backup.dat";

mySettings.ExportMethod = SPExportMethodType.ExportAll;

mySettings.IncludeSecurity = SPIncludeSecurity.All;

mySettings.IncludeVersions = SPIncludeVersions.All;

SPExport myExport = new SPExport(mySettings);

myExport.Run();

Note: For more information about the SharePoint Content Deployment and Migration API, see [Microsoft.SharePoint.Deployment Namespace](http://go.microsoft.com/fwlink/?LinkId=140145&clcid=0x409) at (http://go.microsoft.com/fwlink/?LinkId=140145&clcid=0x409).

Planning the Content Deployment System

If you want to migrate content between a staging environment and a publishing environment, you must plan the content deployment process. In addition to choosing the right deployment methods for your organization, you must plan how the deployment will work in practice. Your considerations should include how to enable communication between the environments, how to migrate custom controls or assemblies, and how to plan migration schedules to meet the requirements of your organization.

The following table displays deployment methods and details.

|  |  |
| --- | --- |
| **Aspect** | **Details** |
| Planning permissions | * Staging and publishing environments are typically on separate server farms and Active Directory domains. * Must have permissions to the Central Administration site on destination server farm. |
| Custom controls and code | Assemblies must be deployed to the destination farm manually. |
| Planning job schedules | * Consider effects on response time. * Consider requirements for custom assemblies. |

This section contains the following topics:

* [Planning Permissions](#PlanningPermissions)
* [Custom Controls and Code](#CustomControlsandCode)
* [Planning Job Schedules](#PlanningJobSchedules)

Planning Permissions

If your organization uses a staging environment and an Internet-facing publishing environment, you are likely to deploy each environment on a separate server farm to provide a greater degree of security isolation for the production environment. In most circumstances, you will also maintain each server farm on a separate Microsoft® Active Directory® directory service domain, with the production farm in a perimeter network and the staging farm in the internal network.

In deployments such as these, you will plan user permissions separately for each environment. For example, an Internet-facing environment typically requires the granting of some read-only privileges to anonymous users, whereas this is not necessary for an internal staging environment. However, you must also plan the accounts that your server farms will use to communicate with each other during the deployment process.

When you configure a content deployment path on the source farm, you must have permissions to the Central Administration site on the destination server farm. In practice, this means that you must be a member of the Farm Administrators security group on both server farms (although you may use different accounts to access each).

Custom Controls and Code

Earlier in this lesson, you saw how the path and job features in Office SharePoint Server 2007 automatically manage content dependencies when you migrate content between environments. The job identifies any resources that are required to render your content and ensures that those resources are migrated with your content.

However, this only applies to resources within the scope of the site collection. If you want to migrate custom controls, Web Parts, or any other functionality that relies on an assembly, you must manually install the required assemblies on the destination server. You must also take any necessary steps to ensure that the assembly functions correctly, such as modifying the security policy or adding safe control entries to the Web.config file. You can use SharePoint features and solutions, with custom install batch files, to package and deploy custom controls and assemblies.

Planning Job Schedules

When you create jobs to migrate content from a staging environment to a production environment, you should consider the following:

* Content deployment might hurt response times for users, even if you have dedicated Web servers for content import and export. If possible, schedule deployment jobs at off-peak times.
* If your content relies on custom controls or code, ensure that you allow sufficient time to deploy the required assemblies on the destination server before the job migrates your content. This includes both the manual deployment effort time and the timer jobs used to deploy solutions.

Quick Deploy and Page Scheduling

In most cases, you schedule content migration at regular intervals by using either paths and jobs or your own custom solution. Occasionally, however, you might need to add or modify content on your production environment with more urgency. You may also want to exercise control over when individual pages are added and removed. Office SharePoint Server 2007 provides two features, Quick Deploy and page scheduling, that can help you in these situations, as shown in the following table.

|  |  |
| --- | --- |
| **Name** | **Description** |
| [Quick Deploy](#QuickDeploy) | Enables users to deploy an individual page without waiting for the scheduled migration job |
| [Page scheduling](#PageScheduling) | Enables users to specify when a page is to be published and when it is to be removed |

Quick Deploy

In some circumstances, you may want to deploy content to your production environment without waiting for the scheduled migration job. For example, you might want to deploy a press release or news item as quickly as possible. Office SharePoint Server 2007 includes a feature called Quick Deploy that enables users with the necessary privileges to bypass the migration schedule and deploy single pages to a production environment quickly and easily.

When you create a new migration path, Office SharePoint Server 2007 automatically creates a Quick Deploy job, provided that the Publishing Infrastructure feature is enabled on the site collection. This adds a Quick Deploy menu item to the Page Editing toolbar. If a user edits a page and clicks the Quick Deploy menu item, the new page version is added to the Quick Deploy Items document library. The Quick Deploy job checks the Quick Deploy Items library for new content every 15 minutes by default (this can be set to 10, 15, 30, 45, or 60 minutes), and deploys that content to the destination site collection.

An author must be a member of the Quick Deploy Users security group to use the Quick Deploy feature. By default, the Quick Deploy Users group contains only site owners.

Page Scheduling

If you publish time-sensitive content to your production environment, you can use page scheduling to specify when the content is published and when it is removed. For example, you might want to deploy specialized content for festive periods, ensure monthly sales reports are only accessible for the current month, or display press releases for a fixed time only.

Page scheduling enables you to specify a publishing start date and time and a publishing end date and time when you create a new version of a page. If you create a publishing schedule in your staging environment, and then deploy that page to your production environment, the page schedule is migrated along with the page content. For example, if you migrate a page at 1:00 P.M. but it is not scheduled for publication until 2:00 P.M., then the scheduled version of that page will not be visible in your production environment until 2:00 P.M.

In Office SharePoint Server 2007, each site collection contains a special document library called the Pages library that manages all the Web pages that are contained in that site. Before you can schedule start dates and end dates for a particular page, you must configure the Pages library to use major and minor versions and to require content approval for submitted items. You must then explicitly enable item scheduling for the Pages library.

Note: To access the settings for the Pages library, on the **Site Actions** menu, click **Site Settings,** **Site Libraries and Lists**, and **Customize “Pages”**.

The page-scheduling process effectively adds a stage to the page moderation process. The following table shows a typical life cycle for a scheduled page.

|  |  |
| --- | --- |
| **Page Status** | **Description** |
| Draft | An author edits the page, thereby creating a new draft version. |
| Pending | The author submits the page for approval. |
| Scheduled | A user with approval permissions approves the page. The page status remains at scheduled until the publishing start time. |
| Approved | The publishing start time arrives. The page becomes the major (published) version until the publishing end time. |
| Draft | The publishing end time arrives. The page reverts back to a draft version. |

Site Variations and Content Deployment

In Office SharePoint Server 2007, you can use site variations to create different versions of your SharePoint sites for different markets. For example, you can use variations to create page versions in different languages, or to support different browser-enabled devices. The following table details basic aspects of site variation and content deployment.

|  |  |
| --- | --- |
| **Aspect** | **Details** |
| [Variations overview](#VariationsOverview) | * Alternative versions of a site. * Based on language, browser device, or any other criteria |
| [Publishing process](#PublishingProcess) | * Page amendments on source sites are copied to target sites as draft versions. * Workflows route the draft versions to translators or page designers. |
| [Multiple environment issues](#VariationsandMultipleEnvironmentDeploy) | * Implement variations in staging site or publishing site? * Workflows are not migrated. * Changes to variation logic are not migrated. |

Variations Overview

Variations are created automatically when you create a variation label for a particular site. Office SharePoint Server 2007 copies content from source variation sites to target variation sites as draft versions, and initiates workflows that enable translators or page designers to make the necessary changes to the alternative versions. Users are directed to a site variation based on their browser language, or based on any criteria that you configure in the VariationsRootLanding.ascx file for the site collection.

Publishing Process

A typical scenario for the use of site variations could be a dual-language site in English and Spanish. In this case, Office SharePoint Server 2007 manages the publishing process as follows:

1. An author edits a page in the English variation site, and then submits the page for approval.
2. An approver checks the page amendments, and the page is published on the English variation site as a major version.
3. The amended page is copied to the Spanish variation site as a draft version.
4. A workflow routes the amended page on the target variation site to a translator.
5. The translator translates the draft page into Spanish, and then submits the page for approval.
6. An approver checks the page amendments, and the page is published on the Spanish variation site as a major version.

Variations and Multiple Environment Deployments

If you use a staging environment and a publishing environment, you must consider how site variations will be managed by your content deployment process. If you add site variations to a staging environment, be aware of the following key points:

* Workflows are not deployed with content. If some translation tasks are outstanding when the deployment job executes, you may end up with inconsistent major versions in the publishing environment.
* The VariationsRootLanding.ascx file is an ASP.NET Web control, and will not be migrated by the deployment job. If you make any changes to the variation logic in the staging environment, you must replicate these changes manually in the production environment.

Because of these issues, it can be more straightforward to implement site variations directly in the production environment. If you do this, however, you must be able to support the approval process in the production environment.

Review of Module 3

* Server farms for Web Content Management solutions
* Content deployment

References

The following videos provide supplemental information to these modules. There is no one-to-one correspondence between the modules and the videos:

* [Video 1](http://go.microsoft.com/fwlink/?LinkId=140097&clcid=0x409) (http://go.microsoft.com/fwlink/?LinkId=140097&clcid=0x409)
* [Video 2](http://officecpub/Teams/itpro/Documents/White%20paper%20library/WCM%20modules/Video%202) (http://go.microsoft.com/fwlink/?LinkID=140063&clcid=0x409)
* [Video 3](http://officecpub/Teams/itpro/Documents/White%20paper%20library/WCM%20modules/Video%203) (http://go.microsoft.com/fwlink/?LinkID=140068&clcid=0x409)
* [Video 4](http://officecpub/Teams/itpro/Documents/White%20paper%20library/WCM%20modules/Video%204) (http://go.microsoft.com/fwlink/?LinkId=140101&clcid=0x409)