Deployment guide for Microsoft SharePoint Server 2010

Microsoft Corporation

Published: January 2011

Author: Microsoft Office System and Servers Team (itspdocs@microsoft.com)

Abstract

This book provides deployment instructions for Microsoft SharePoint Server 2010. The audiences for this book include application specialists, line-of-business application specialists, and IT administrators who are ready to deploy SharePoint Server 2010 and want installation steps.

The content in this book is a copy of selected content in the SharePoint Server 2010 technical library (*http://go.microsoft.com/fwlink/?LinkId=181463*) as of the publication date. For the most current content, see the technical library on the Web.

Microsoft

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Access, Active Directory, Backstage, Excel, Groove, Hotmail, InfoPath, Internet Explorer, Outlook, PerformancePoint, PowerPoint, SharePoint, Silverlight, Windows, Windows Live, Windows Mobile, Windows PowerShell, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

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.

Contents

Getting help xiv

Deployment for SharePoint Server 2010 1

Deployment overview (SharePoint Server 2010) 3
Concepts 3
Physical architecture 4
    Size 4
    Topology 5
Installation and configuration 5
    Prepare the servers 5
    Create the farm 7
    Configure settings, services, solutions, and sites 7
Deployment stages 7
    Planning 8
    Development 8
    Proof of concept (POC) 9
    Pilot 9
    User acceptance test (UAT) 10
    Production 10

Install prerequisites from a network share (SharePoint Server 2010) 11
Installer switches and arguments 11
Download and consolidate the prerequisites on a file share 12
Install the prerequisites from the command line 13
Install the prerequisites using an arguments file 13
    Known issues 15

Installing SharePoint Server 2010 17
Related resources 17

Prepare for deployment (SharePoint Server 2010) 19

Hardware and software requirements (SharePoint Server 2010) 20
Overview 20
Hardware requirements—Web servers, application servers, and single server installations 20
Hardware requirements—Database servers 21
Software requirements 22
    Minimum requirements 22
    Optional software 25
Access to applicable software 26

Administrative and service accounts required for initial deployment (SharePoint Server 2010) 29
Required permissions 29

Harden SQL Server for SharePoint environments (SharePoint Server 2010) 31
Summary of hardening recommendations 31
Configure a SQL Server instance to listen on a non-default port 31
Configure Windows Firewall to block default SQL Server listening ports 33
Configure Windows Firewall to open manually assigned ports 33
Configure a SQL client alias 33
Test the SQL client alias 34

Deployment scenarios (SharePoint Server 2010) 35

Deploy a single server with SQL Server (SharePoint Server 2010) 36
Overview 36
Before you begin 37
Install SharePoint Server 2010 37
    Run the Microsoft SharePoint Products Preparation Tool 38
    Run Setup 38
    Run the SharePoint Products Configuration Wizard 39
    Configure browser settings 41
    Run the Farm Configuration Wizard 42
Post-installation steps 44

Deploy a single server with a built-in database (SharePoint Server 2010) 45
Overview 45
Before you begin 46
Install SharePoint Server 2010 46
    Run the Microsoft SharePoint Products Preparation Tool 46
    Run Setup 47
    Run the SharePoint Products Configuration Wizard 47
    Configure browser settings 48
Post-installation steps 50

Multiple servers for a three-tier farm (SharePoint Server 2010) 52
Overview 52
    Topology overview 53
    Before you begin 54
Prepare the farm servers 55
Install SharePoint Server 2010 on the farm servers 56
Create and configure the farm 57
Add Web servers to the farm 59
Configure diagnostic logging and usage and health data collection 59

Create and configure a new Search Service Application 60
Create a site 62
Add content sources and configure a search crawl schedule 63
Post-installation steps 63

Quick start: Deploy single server in an isolated Hyper-V environment (SharePoint Server 2010) 65
Requirements and recommendations 65
    Virtualization server and virtual machine configuration 65
    Deployment environment 67
Required permissions 67
Pre-deployment tasks 67
Deploy SharePoint Server 2010 manually 68
Deploy SharePoint Server 2010 by using Windows PowerShell scripts 68

Deploy by using DBA-created databases (SharePoint Server 2010) 74
Before you begin 74
    Farm server requirements 74
    Database requirements 74
About configuring DBA-created databases 75
Create and configure databases for Central Administration 76
Create and configure additional databases 78

Deploy in a virtual environment (SharePoint Server 2010) 85

Virtual machine guidance (SharePoint Server 2010) 86
Networking 86
    Network adapters 87
Virtual machine configuration 88
    Memory 88
    Processor 89
    Controller and hard disk 89
Integration services 90
    Automatic stop and start 90
    Using snapshots 91

Install SharePoint Server 2010 by using Windows PowerShell 92
Farm server requirements 92
Prepare SPModule 93
Install Microsoft SharePoint Server 2010 by running Install-SharePoint 94
    Run Install-SharePoint using a PID key 96
Configure the farm by using New-SharePointFarm 97
    Configure SharePoint Server 2010 on a stand-alone server 97
Create a Web Application using Windows PowerShell 98

Deploy services using the SharePoint 2010 Farm Configuration Wizard 99
Create a site collection by using Windows PowerShell 99
Perform additional configuration tasks 100
Add servers to the farm using Join-SharePointFarm 100
Configure the trace log 100

Initial configuration (SharePoint Server 2010) 102

Deploy language packs (SharePoint Server 2010) 103
About language IDs and language packs 104
Downloading language packs 105
Preparing the Web servers for language packs 106
Installing language packs on the Web servers 107
Uninstalling language packs 109

Configure farm settings (SharePoint Server 2010) 110

Configure usage and health data collection (SharePoint Server 2010) 111
Configure usage and health data collection by using Central Administration 111
Configure usage data collection by using Windows PowerShell 112
To configure usage data collection for a specific event type by using Windows PowerShell 113
Log usage data in a different logging database by using Windows PowerShell 113

Configure diagnostic logging (SharePoint Server 2010) 115
Best practices 115
Configure diagnostic logging by using Central Administration 117
Configure diagnostic logging by using Windows PowerShell 118

E-mail integration (SharePoint Server 2010) 120

Configure incoming e-mail (SharePoint Server 2010) 121
Overview 121
Install and configure the SMTP service 122
    Install the SMTP service 122
    Install IIS 6.0 Management tools 122
    Configure the SMTP service 123
Configure incoming e-mail in a basic scenario 124
Configure incoming e-mail in an advanced scenario 124
Prepare your environment for incoming e-mail in an advanced scenario 128
    Configure AD DS to be used with Directory Management Service 128
    Configure DNS Manager 131
    Add an SMTP connector in Microsoft Exchange Server 2010 132
    Configure permissions to the e-mail drop folder 133
Are attachments missing from e-mail messages that are sent to a SharePoint document library? 135

Configure outgoing e-mail (SharePoint Server 2010) 137
Install and configure the SMTP service 138
Install the SMTP service 138
Configure the SMTP service 138
Configure outgoing e-mail for a farm 140
Configure outgoing e-mail for a specific Web application 141

Configure a mobile account (SharePoint Server 2010) 144
Import a root certificate and create a trusted root authority 145
Configure a mobile account 145
Retrieve mobile account information 149
Delete a mobile account 150

Install and configure Remote BLOB Storage (RBS) with the FILESTREAM provider (SharePoint Server 2010) 152
Enable FILESTREAM and provision the RBS data store 153
Install RBS 154
Enable and test RBS 155

Configure and use the Exchange connector (SharePoint Server 2010) 157
Create a crawl rule 157
Add a content source for Microsoft Exchange public folders 158

Deploy Microsoft Business Intelligence Indexing Connector (SharePoint Server 2010) 161

Install and configure Business Intelligence Indexing Connector (back end) 162
Before you start 162
Considerations for crawling external data sources 162
Install the Business Intelligence Indexing Connector 163
Verification of configuration 165

Uninstall Business Intelligence Indexing Connector 166
To uninstall the Business Intelligence Indexing Connector 166

Install or uninstall SharePoint Solution Pack for Business Intelligence Indexing Connector (front end) 167
Before you start 167
SharePoint Solution Pack for SharePoint Server 167
Uninstall SharePoint Solution Pack 168

Install or uninstall SharePoint Solution Pack language pack for Business Intelligence Indexing Connector (front end) 170
SharePoint Solution Pack language pack 170
Uninstall SharePoint Solution Pack Language Pack 171

Troubleshooting Business Intelligence Indexing Connector 172
Data sources 172
Excel and PowerPivot files 172
Unified Logging Service log file or trace logs 172

Configure services (SharePoint Server 2010) 173

Service application and service management (SharePoint Server 2010) 174
In This Section 174

Configure the security token service (SharePoint Server 2010) 175
How Web applications that use an STS work 175
Configure a SharePoint claims-based Web application by using Windows PowerShell 176
Edit bindings 177
Configure a Web application that uses an STS 178

Configure the Secure Store Service (SharePoint Server 2010) 179
Initialize an instance of a Secure Store Service application 180
Refresh the encryption key 180
Generate a new encryption key 181
Create a target application 182
Set credentials for a target application 184
Enable the audit log 184

Post-installation steps for search (SharePoint Server 2010) 186
Configure basic search functionality 186
To go to the Search Administration page 187
Specify the default content access account 187
Specify the contact e-mail address 187
Configure farm-wide search settings 188
Create content sources for crawling content 188
Search Center site 189
Grant all authenticated users access to the Search Center 190
Enable search reporting 190
Verify that the Web Analytics service application is started 190
Create the Web Analytics service application 191
Start the Web Analytics Web service 191
Run search queries 192
View query reports 192
View query reports at the Web application level 192
View query reports at the site-collection level 192
View other search-related reports 193
Enable people search 193

Ensure service applications are running ... 193
Configure Web application and site collections ... 195
Add user profiles to the User Profile service application ... 196
Configure My Site settings ... 197
Add information to My Site sites ... 198
Crawl the profile store ... 198
Optional steps ... 200
Create a crawl rule (optional) ... 200

Configure a managed metadata service application (SharePoint Server 2010) ... 201

Prepare to host sites (SharePoint Server 2010) ... 202

Create a Web application (SharePoint Server 2010) ... 203

Create a Web application that uses Windows-classic authentication (SharePoint Server 2010) ... 205

Create a Web application that uses Windows-claims authentication (SharePoint Server 2010) ... 211

Configure claims authentication (SharePoint Server 2010) ... 218

Configure anonymous access for a claims-based Web application (SharePoint Server 2010) ... 219
Configure anonymous access for a claims-based Web application ... 219

Configure forms-based authentication for a claims-based Web application (SharePoint Server 2010) ... 220
Configure a forms-based Web application to use an LDAP provider by using Central Administration ... 220
Configure the LDAP Web.Config files ... 221
Configure a forms-based Web application to use an LDAP provider by using Windows PowerShell ... 225

Configure Kerberos authentication for the claims to Windows token service (SharePoint Server 2010) ... 227

Configure authentication using a SAML security token (SharePoint Server 2010) ... 228
Configure an Identity Provider STS (IP-STS) Web application by using Windows PowerShell ... 228
Configure a Relying Party STS (RP-STS) Web application ... 230
Establish a trust relationship with an Identity Provider STS (IP-STS) by using Windows PowerShell ... 231
Export the trusted IP-STS certificate by using Windows PowerShell ... 231
Define a unique identifier for claims mapping by using Windows PowerShell ... 232
Create a new authentication provider ... 232
Create a new SharePoint Web application and configure it to use SAML sign-in ... 233

Configure claims-based authentication using Windows Live ID (SharePoint Server 2010) ... 234

Configure the Window Live ID Security Token Service 235
Configure SharePoint for Window Live ID authentication 235
Convert a Window Live ID internal environment to a production environment 238
Create different types of SharePoint claims-based Web applications 238
Grant permissions to all Window Live ID authenticated users 254

Configure Kerberos authentication (SharePoint Server 2010) 255
About Kerberos authentication 255
Before you begin 256
    Software version requirements 256
    Known issues 257
    Additional background 257
    Server farm topology 258
    Active Directory Domain Services, computer naming, and NLB conventions 259
    Active Directory domain account conventions 259
    Preliminary configuration requirements 260
Configure Kerberos authentication for SQL communications 261
    Create the SPNs for your SQL Server service account 261
    Confirm Kerberos authentication is used to connect servers running SharePoint Server 2010 to SQL Server 262
Create Service Principal Names for your Web applications using Kerberos authentication 264
Deploy the server farm 265
    Install SharePoint Server 2010 on all of your servers 265
    Create a new farm 266
    Join the other servers to the farm 268
Configure services on servers in your farm 269
    Windows SharePoint Services Search 269
    Index server 269
    Query server 269
Create Web applications using Kerberos authentication 270
    Create the portal site Web application 270
    Create the My Site Web application 271
Create a site collection using the Collaboration Portal template in the portal site Web application 271
Confirm successful access to the Web applications using Kerberos authentication 272
Confirm correct Search Indexing functionality 274
Confirm correct Search Query functionality 275
Configuration limitations 275
Additional resources and troubleshooting guidance 275

Configuring Kerberos authentication for SharePoint 2010 Products (white paper) 277
Two options 277

Create a site collection (SharePoint Server 2010) 278

Create a site collection by using Central Administration 279
Create a site collection by using Windows PowerShell 280

Uninstall SharePoint Server 2010 281
Uninstall SharePoint Server 2010 281

Deploy customizations - overview (SharePoint Server 2010) 283
Process overview 283
Before you begin 283
About the two kinds of customizable site elements 284
Deploying developed site elements 285
Deploying authored site elements 285

Deploy solution packages (SharePoint Server 2010) 288
What is a solution package? 288
Deploying site elements by using solution packages 289
When to use solution packages 289
Deploying farm solutions 289
Adding a solution package 290
Deploying a solution package 291
About creating a solution package 292
Creating and deploying a custom Web Part solution package by using Visual Studio 2010 295

Deploy authored site elements (SharePoint Server 2010) 296
About deploying authored site elements 296
When to use the content deployment feature 297
When to use a content deployment package 298
Before you begin 298
Deploy content by using content deployment jobs 298
Deploy content by using the Content Migration API 299
Create a content deployment package by using Windows PowerShell 300

Deploy site elements by using Features (SharePoint Server 2010) 302
What is a Feature? 302
When to use Features 303
Create a Feature 303
Install and activate a Feature by using Windows PowerShell 305

Deploy templates (SharePoint Server 2010) 308
What are site definitions? 308
Site definitions and configurations 309
Uncustomized pages and page customization 309
Core schema files 310
Create a custom site definition and configuration 310

Deploy a site definition by using a solution package 313
Add a SiteDefinitionManifest element 313
Add a TemplateFile element 313

Workflow deployment process (SharePoint Server 2010) 314
Overview 314
Before you begin 315
Deploying workflows 315
Deploy predefined workflows 315
Deploy SharePoint Designer workflows 316
Deploy Visual Studio workflows 318
Verification 319

Deploy software updates for SharePoint Server 2010 320

Software updates overview (SharePoint Server 2010) 321
Improvements and new features 321
Intended audience and scope 321
Software update process 322
Update phase 322
Upgrade phase 322
Software update strategy 323
Software update deployment cycle 323
Learn 324
Prepare 325
Test 327
Implement 327
Validate 329

Prepare to deploy software updates (SharePoint Server 2010) 331
Verify account permissions and security settings 331
Determine the update approach 331
Back up the environment 333
Document the environment 334
Determine whether related items need to be updated 334
Obtain the software update and prepare the installation source (optional) 334
Slipstream package 335

Install a software update (SharePoint Server 2010) 336
Verify the update strategy 336
Monitor installation progress 336
Handle update failures 337
Review update scenarios 337

Initial state and required conditions 338
Use the in-place method without backward compatibility 339
Use the in-place method with backward compatibility 340
Update phase 341
Upgrade phase 343
Use the database attach method for high availability of existing content 344
Verify update completion and success 347

Deploy Office Web Apps (Installed on SharePoint 2010 Products) 348
Understanding Office Web Apps deployment 349
Install and configure Office Web Apps on an existing stand-alone SharePoint server 350
Run Office Web Apps setup 350
Run PSConfig to register the services 350
Start the service instances 351
Create the service applications and the service application proxies 352
Activate the Office Web Apps Feature 353
Install and configure Office Web Apps on a new stand-alone SharePoint server 355
Run Office Web Apps setup 355
Run PSConfig to register the services, start the service instances, create the service applications and proxies, and activate the Office Web Apps Feature 355
Install and configure Office Web Apps on an existing SharePoint server farm 356
Run Office Web Apps setup 356
Run PSConfig to register services 356
Start the service instances 357
Create the service applications and the service application proxies 358
Activate the Office Web Apps Feature 360
Install and configure Office Web Apps on a new SharePoint server farm 361
Run Office Web Apps setup 361
Run PSConfig to register services 362
Run the SharePoint Farm Configuration Wizard to start the service instances, create the service applications and proxies, and activate the Office Web Apps Feature 362
Additional configuration (optional) 363
Configure the SharePoint default open behavior for browser-enabled documents 363
Troubleshooting 365

Getting help

Every effort has been made to ensure the accuracy of this book. This content is also available online in the Office System TechNet Library, so if you run into problems you can check for updates at:

http://technet.microsoft.com/office

If you do not find your answer in our online content, you can send an e-mail message to the Microsoft Office System and Servers content team at:

itspdocs@microsoft.com

If your question is about Microsoft Office products, and not about the content of this book, please search the Microsoft Help and Support Center or the Microsoft Knowledge Base at:

http://support.microsoft.com

Deployment for SharePoint Server 2010

Welcome to the deployment guide for Microsoft SharePoint Server 2010. The articles in this guide help you prepare to install and configure SharePoint Server 2010. The deployment guide includes information about deployment scenarios, step-by-step installation instructions, and post-installation configuration steps. It also describes how to upgrade to SharePoint Server 2010.

Before installing SharePoint Server 2010, be sure you have reviewed the information in Planning and architecture for SharePoint Server 2010 (*http://technet.microsoft.com/library/0ed0b44c-d60d-4b85-87de-19065d968835(Office.14).aspx*).

In this section:

- Deployment overview (SharePoint Server 2010)

  This article provides information about deploying SharePoint Server 2010. The goal of this article is to provide information that can help you and your teams make fundamental decisions about deploying SharePoint Server 2010-based solutions in your organization.

- Install prerequisites from a network share (SharePoint Server 2010)

  This article explains how to use PrerequisiteInstaller.exe to install prerequisites from a network share. Doing so can result in a more consistent deployment.

- Installing SharePoint Server 2010

  This video illustrates how to install and configure Microsoft SharePoint Server 2010 and how to create your first site collection.

- Prepare for deployment (SharePoint Server 2010)

  This section provides essential steps to take before you deploy Microsoft SharePoint Server 2010.

- Deployment scenarios (SharePoint Server 2010)

  Follow the steps in this section to deploy a single server or server farm with SharePoint Server 2010, or to deploy the Office Web Apps for use with SharePoint Server 2010.

- Initial configuration (SharePoint Server 2010)

  Follow the steps in this section to install language packs, configure farm settings and services, and create site collections.

- Uninstall SharePoint Server 2010

  This article explains how to uninstall SharePoint Server 2010.

- Deploy customizations - overview (SharePoint Server 2010)

  The articles in this section describe how to deploy site elements that have been customized by developers or Web designers in a SharePoint Server 2010 environment.

- Deploy software updates for SharePoint Server 2010

  Microsoft periodically releases software updates for SharePoint Server 2010. Follow the steps in this section to apply these updates to your servers running SharePoint Server.

- Upgrading to SharePoint Server 2010 (*http://technet.microsoft.com/library/396c85d9-4b86-484e-9cc5-f6c4d725c578(Office.14).aspx*)

  Follow the steps in the upgrade guide to plan, prepare, and perform an upgrade to SharePoint Server 2010.

Note:
If you plan to use Office Web Apps, you must install and configure them to work with SharePoint 2010 Products. For more information, see Deploy Office Web Apps (Installed on SharePoint 2010 Products).

Additional resources:

- For a graphical overview of the deployment process, download the SharePoint 2010 Products Deployment model from the Technical diagrams (SharePoint Server 2010) (*http://technet.microsoft.com/library/bcbae7bd-656b-4003-969c-8411b81fcd77(Office.14).aspx*) article.
- For a demo video about installing, see Installing SharePoint Server 2010.

Deployment overview (SharePoint Server 2010)

This article contains an overview of a Microsoft SharePoint Server 2010 farm deployment. Although Microsoft SharePoint Server farms vary in complexity and size, a combination of careful planning and a phased deployment that includes ongoing testing and evaluation significantly reduces the risk of unexpected outcomes.

Note:
For information about site and solution planning, which is not in scope for this article, see Plan for sites and solutions (SharePoint Server 2010) (*http://technet.microsoft.com/library/f2cb570f-f893-4b42-b8a7-d944952b6626(Office.14).aspx*).

For a visual representation of the information in this article, see the SharePoint 2010 Products Deployment model in the Technical diagrams (SharePoint Server 2010) (*http://technet.microsoft.com/library/bcbae7bd-656b-4003-969c-8411b81fcd77(Office.14).aspx*) topic. Related technical diagrams include "Topologies for SharePoint Server 2010 and Services in SharePoint 2010 Products".

In this article:

- Concepts
- Physical architecture
- Installation and configuration
- Deployment stages

Concepts

The logical result of SharePoint Server's flexibility and richness can be a high degree of complexity around installing and configuring SharePoint Server correctly. A fundamental understanding of the following key structural elements in a SharePoint Server environment is required in order to correctly deploy and support SharePoint Server 2010 products:

- Server farm: The top-level element of a logical architecture design for SharePoint Server.
- Web application: An IIS Web site that is created and used by SharePoint Server 2010.
- Content database: Provides storage Web application content. You can separate content into multiple content databases at the site collection level.
- Site collection: A set of Web sites that have the same owner and share administration settings.
- Site: One or more related Web pages and other items (such as lists, libraries, and documents) that are hosted inside a site collection.

For more information about these and other architectural components, see Logical architecture components (SharePoint Server 2010) (*http://technet.microsoft.com/library/aaed3a01-f4dc-4353-abda-0beced2080b6(Office.14).aspx*).

In addition to understanding the elements of a SharePoint Server environment and how they have to be configured for your solution, you must consider the following additional factors: physical architecture, installation and configuration, and the various stages of deployment.

Physical architecture

The physical architecture, which consists of one or more servers and the network infrastructure, enables you to implement the logical architecture for a SharePoint Server solution. The physical architecture is typically described in two ways: by its size and by its topology. Size, which can be measured in several ways, such as the number of users or the number of documents, is used to categorize a farm as small, medium, or large. Topology uses the idea of tiers or server groups to define a logical arrangement of farm servers.

Size

Size uses the number of users and number of content items as a fundamental measure to indicate whether a server farm is small, medium, and large, as follows:

- A small server farm typically consists of at least two Web servers and a database server. One of the Web servers hosts the Central Administration site and the other handles additional farm-related tasks, such as serving content to users.

  The small farm can be scaled out to three tiers using a dedicated application server in response to the number of users, the number of content items, and the number of services that are required.
- A medium server farm typically consists of two or more Web servers, two application servers, and more than one database servers. We recommend that you start with the preceding configuration and then scale out to accommodate the workload placed on the servers.

  In scenarios where services are known to use a disproportionate amount of resources, you can scale out the application tier. Performance data will indicate which services you should consider off-loading to a dedicated server.
- A large server farm can be the logical result of scaling out a medium farm to meet capacity and performance requirements or by design before a SharePoint Server solution is implemented. A three-tier topology environment typically uses dedicated servers on all the tiers. Additionally, these servers are often grouped according to their role in the farm. For example, all client-related services can be grouped onto one or two servers and then scaled out by adding servers to this group as needed in response to user demand for these services.

  Note:
  The recommendation for scaling out a farm is to group services or databases with similar performance characteristics onto dedicated servers and then scale out the servers as a group. In large environments, the specific groups that evolve for a farm depend on the specific demands for each service in a farm.

For specific numbers related to small, medium, and large farms, see Performance and capacity management (SharePoint Server 2010) (*http://technet.microsoft.com/library/8dd52916-f77d-4444-b593-1f7d6f330e5f(Office.14).aspx*).

Topology

Topology uses tiers as a model for logically arranging farm servers according to the components that they host or their roles in a server farm. A SharePoint Server farm is deployed on one, two, or three tiers, as follows:

- In a single-tier deployment, SharePoint Server and the database server are installed on one computer.
- In a two-tier deployment, SharePoint Server components and the database are installed on separate servers. This kind of deployment maps to what is called a small farm. The front-end Web servers are on the first tier and the database server is located on the second tier. In the computer industry, the first tier is known as the Web tier. The database server is known as the database tier or database back-end.
- In a three-tier deployment, the front-end Web servers are on the first tier, the application servers are on the second tier, which is known as the application tier, and the database server is located on the third tier. A three-tier deployment is used for medium and large farms.

Installation and configuration

After you finish planning your SharePoint Server solution you can create a SharePoint Server farm to host the solution. The first step is to install SharePoint Server 2010 and create the farm that is required for the solution. The process of preparing your environment consists of the following phases:

1. Prepare the servers
2. Create the farm
3. Configure settings, services, solutions, and sites

Note:
The farm that you create and deploy will undergo significant changes in size, topology, and complexity as you move through the different deployment stages illustrated in the SharePoint 2010 Products Deployment model. This is typical and the expected result of a phased deployment. This is why we recommend that you follow all of the stages described in the "Deployment stages" section of this article.

Prepare the servers

In this phase, you get your servers ready to host the product. This includes the supporting servers and the servers that will have SharePoint Server installed. The following servers must be configured to support and host a farm:

- Domain controller: The required farm accounts have to be configured for the domain and directory synchronization must be configured.

  Important:
  SharePoint Server 2010 does not support single label domain (SLD) names. Because the use of SLD names is not a recommended practice, SharePoint 2010 Products are not extensively tested in this scenario. Therefore, there may be incompatibility issues when SharePoint 2010 Products are implemented in a single label domain environment. For more information, see Information about configuring Windows for domains with single-label DNS names (*http://go.microsoft.com/fwlink/?LinkID=193849*) and the DNS Namespace Planning Solution Center (*http://go.microsoft.com/fwlink/?LinkId=198010*).

  For information about required accounts, see:

  - Administrative and service accounts required for initial deployment (SharePoint Server 2010)
  - About Directory Synchronization (*http://go.microsoft.com/fwlink/?LinkId=193169*)
- Database server: The required version of SQL Server, including service packs and cumulative updates must be installed on the database server. The installation must include any additional features, such as SQL Analysis Services, and the appropriate SharePoint Server logins have to be added and configured. The database server must be hardened and, if it is required, databases must be created by the DBA. For more information, see:
  - Hardware and software requirements (SharePoint Server 2010)
  - Harden SQL Server for SharePoint environments (SharePoint Server 2010)
  - Deploy by using DBA-created databases (SharePoint Server 2010)
- Application servers and front-end Web servers: The farm servers that will have SharePoint Server installed must be prepared as follows: verify that they meet the hardware requirements, have the operating system hardened, have the required networking and security protocols configured, have the SharePoint Server 2010 software prerequisites installed and hardened, and have the required authentication configured. For more information, see:
  - System requirements (SharePoint Server 2010) (*http://technet.microsoft.com/library/64233599-f18c-4081-a3ce-450e878a1b9f(Office.14).aspx*)
  - "Installing software prerequisites" in Hardware and software requirements (SharePoint Server 2010)
  - Plan security hardening (SharePoint Server 2010) (*http://technet.microsoft.com/library/763613ac-83f4-424e-99d0-32efd0667bd9(Office.14).aspx*)
  - Plan authentication (SharePoint Server 2010) (*http://technet.microsoft.com/library/a0f2c651-e353-4197-b0dd-0cbe6f18c811(Office.14).aspx*)

Create the farm

In this phase, you install the product and configure each server to support its role in the farm. You also create the configuration database and the SharePoint Central Administration Web site. The following servers are required for a SharePoint Server farm:

- Database server: Unless you plan to use DBA-created databases, the configuration database, content database, and other required databases are created when you run the SharePoint Products Configuration Wizard.
- Application server: After you prepare the application server, install any additional components that are required to support functions such as Information Rights Management (IRM) and decision support. Install SharePoint Server on the server that will host SharePoint Central Administration Web site and then run the SharePoint Products Configuration Wizard to create and configure the farm.
- Front-end Web server: Install SharePoint Server on each Web server, install language packs, and then run the SharePoint Products Configuration Wizard to add the Web servers to the farm.

  Note:
  After you add and configure all the front-end Web servers, you can add any additional application servers that are part of your topology design to the farm.

For more information about supported deployment scenarios, see Deployment scenarios (SharePoint Server 2010).

Configure settings, services, solutions, and sites

In this phase, you prepare the farm to host your site content by completing the following tasks:

- Configure global settings. For more information, see Configure farm settings (SharePoint Server 2010)
- Configure services. For more information, see Configure services (SharePoint Server 2010)
- Deploy solutions and customizations. For more information, see Deploy customizations - overview (SharePoint Server 2010)
- Create and populate the sites. For more information, see Prepare to host sites (SharePoint Server 2010)

Note:
Farm configuration steps are not isolated to a specific tier in the server infrastructure.

Deployment stages

By deploying a SharePoint Server 2010 solution in stages, you gain the benefits that are provided by a systematic approach, such as collecting performance and usage data that you can use to evaluate your solution. Additional benefits include verifying your capacity management assumptions and identifying issues before the farm is put into production.

We recommend that you deploy your farm in the following stages:

- Planning
- Development
- Proof of concept (POC)
- Pilot
- User acceptance test (UAT)
- Production

Planning

Before you can deploy a farm, you must plan the solution that you want to deploy and determine the infrastructure requirements, such as server resources and farm topology. When you finish the planning stage, you should have documented the following:

- An infrastructure design to support your solution
- A detailed description of how you will implement the farm and the solution
- A plan for testing and validating the solution
- A site and solution architecture
- An understanding of the monitoring and sustained engineering requirements to support the solution
- A record of how the solution will be governed
- An understanding of how the solution will be messaged to the user to drive adoption of the solution

We recommend that you use the planning resources and articles described in Planning and architecture for SharePoint Server 2010 (*http://technet.microsoft.com/library/0ed0b44c-d60d-4b85-87de-19065d968835(Office.14).aspx*).

Important:
Resource and time issues may pressure you to be less rigorous during the planning stage. We recommend that you try to be as diligent as possible because missed or lightly touched planning elements can resurface as significant issues after you are in production. These issues can create much additional work, consume unbudgeted resources, and potentially take away from the success of your SharePoint Server.

After the planning stage, you move through the following deployment stages, updating and revising your plans, configurations, and topologies as you test.

Development

During the development stage you will deploy SharePoint Server on a single server or on multiple servers to develop, test, evaluate, and refine the solution that you intend to implement. This environment is scaled according to your needs during solution development and can be retained as a scaled down environment for future development and testing. This is not a stable environment and there are no service-level agreements.

Proof of concept (POC)

During the proof of concept stage, the objective is two-fold: to understand SharePoint Server and to evaluate SharePoint Server in the context of how it can address your business needs. The first level of product evaluation can be done by installing all of the product components on a single server. You do a more extensive product evaluation by a proof-of-concept deployment.

A proof-of-concept deployment on a single server or on a small farm enables you to expand the scope of your evaluation. In this deployment, non-IT staff is added to the evaluation team, which provides a broader view of how SharePoint Server features might be actually be used in the organization. The benefit of a proof-of-concept deployment is that you can collect data that can be used to refine your original plan. This data—such as page views, user behavior patterns, and server resource consumption—also enables you to start to build a benchmark for sizing your farm. A proof of concept is also good when you evaluate service applications and determining what feature sets that you will offer your end users.

It is important during the proof-of-concept stage that you understand the unique characteristics and functionality of these features because this understanding will help you define your overall topology. Be aware that a proof-of-concept deployment requires additional resources and extends the time required to put SharePoint Server into production.

Tip:
Virtualization provides a good platform for evaluating SharePoint Server because a virtual environment provides flexibility, rapid deployment capability, and the ability to roll back virtual machines to previous states.

Pilot

A pilot is used to test your solution on a small scale. There are two approaches to using a pilot deployment. In the first approach, the focus is on functional testing without using real data. By using the second approach you test for production characteristics by using real data and have your pilot users test different kinds of tasks. We recommend the second approach because of the broader scope and real-world data that you can collect and use to refine your solution design.

A pilot deployment provides many benefits. It enables you to collect data that you can use to validate the following aspects of your farm design:

- Infrastructure design
- Capacity management assumptions
- Site and solution architecture
- Solution usage assumptions

The pilot stage also enables you to determine additional data that should be collected to increase the breadth and depth of your benchmarks. This is important if you want to assess the potential effect of additional features or services that you want to add to the farm before the user acceptance test.

At the conclusion of the pilot deployment, you can use the data that you collect to adjust the various components of the solution and its supporting infrastructure.

User acceptance test (UAT)

A user acceptance test deployment—also known as a pre-production environment—is used by organizations as a transitional step from the pilot deployment to a production deployment. An organization's business processes determine the scope, scale, and duration of user accept testing.

The topology of the pre-production environment should be the same as, or very similar to the planned production topology. During user acceptance testing, the SharePoint Server solution is tested against a subset or a complete copy of production data. This deployment stage provides a final opportunity for performance tuning and validating operational procedures such as backups and restores.

Production

The final stage is rolling your farm into a production environment. At this stage, you will have incorporated the necessary solution and infrastructure adjustments that were identified during the user acceptance test stage.

Putting the farm into production requires you to complete the following tasks:

- Deploy the farm.
- Deploy the solution.
- Implement the operations plan.
- If required, deploy additional environments such as authoring and staging farms, and services farms.

Install prerequisites from a network share (SharePoint Server 2010)

This article describes how to install Microsoft SharePoint Server 2010 prerequisites from an offline shared network location using the prerequisite installer (PrerequisiteInstaller.exe) tool.

Installing prerequisites from an offline location is typically required when the servers on which you want to install Microsoft SharePoint Server are isolated from the Internet. Even if this is not the case, installing prerequisites from an offline central location enables you to ensure farm server consistency by installing a well-known and controlled set of images.

Note:
The Microsoft SharePoint Products Preparation Tool is a user interface built on PrerequisiteInstaller.exe. The Microsoft SharePoint Products Preparation Tool accepts no user input.

In this article:

- Installer switches and arguments
- Download and consolidate the prerequisites on a file share
- Install the prerequisites from the command line
- Install the prerequisites using an arguments file
- Known issues

Installer switches and arguments

By using PrerequisiteInstaller.exe with switches and arguments, you have control over which versions of the required software are installed and the location from where they are installed.

PrequisiteInstaller.exe accepts single or multiple switch and argument pairs. A switch identifies the prerequisite and the argument specifies the action and the location of the prerequisite.

A switch and argument pair uses the following format:

/switch: *<path>*

Where:

- */switch* is a valid switch to identify a prerequisite. For example, /NETFX35SP1: is the switch for .NET Framework 3.5 Service Pack 1.
- *<path>* is expressed as the path to a local file or the path to a file share, for example, "C:\foldername\dotnetfx35.exe " or "\\<servername>\<sharename>\dotnetfx35.exe".

Each switch and its argument are separated by a colon and a space. The argument is enclosed in quotes.

The switch and argument pairs can be passed to PrerequisiteInstaller.exe at the command prompt or read from an arguments text file.

Download and consolidate the prerequisites on a file share

The process for downloading and consolidating prerequisites consists of the steps described in the following procedures.

To identify prerequisites

1. Refer to the Hardware and software requirements (SharePoint Server 2010) article, which contains a list of all the required and optional software for SharePoint Server 2010. Additionally, this document provides the download location for each prerequisite that is available for download on the Internet.
2. From the command prompt, navigate to the root of the SharePoint Server 2010 installation media or folder location.
3. At the command prompt, type **PrerequisiteInstaller.exe /?**. This displays a list of the command-line options and switches and their corresponding arguments for installing a prerequisite from the command-line.

   Tip:
   To copy the contents of the active About window to the Clipboard, press CTRL+C.
4. Verify that you have an accurate list of the required software. Compare the output from the prerequisite installer to the list of prerequisites in Step 1.
5. Download the prerequisites to a computer that has Internet access.

Next, use the following procedure to create a central location that you can use for installing SharePoint Server prerequisites on all the farm servers.

To consolidate prerequisites

1. Create a shared folder on a computer that can be accessed by the servers on which the prerequisites will be installed.
2. Copy the files that you downloaded from the Internet to the shared folder.

After you finish creating an accessible network location for the prerequisites, use the procedure in the following section to install SharePoint Server 2010 prerequisites on a server.

Install the prerequisites from the command line

You can install one or all of the prerequisites from the command line using the following procedure.

To install from the command line

1. From the **Start** menu, open the Command Prompt window using **the Run as administrator** option.
2. Navigate to the SharePoint Server source directory.
3. Type the prerequisite program switch and corresponding argument for the program that you want to install, and then press ENTER, for example:

   PrerequisiteInstaller.exe /SQLNCli: "\\o14-sf-admin\SP_prereqs\sqlncli.msi"

   Note

   To install more than one prerequisite, type in each switch and argument pair, taking care to separate each pair by a space, for example:

   PrerequisiteInstaller.exe /SQLNCli: "\\o14-sf-admin\SP_prereqs\sqlncli.msi" /ChartControl: "\\o14-sf-admin\SP_prereqs\MSChart.exe" /W2K8SP2: "\\o14-sf-admin\SP_prereqs\Windows6.0-KB948465-X64.exe" /NETFX35SP1: "\\o14-sf-admin\SP_prereqs\dotnetfx35setup.exe"

Install the prerequisites using an arguments file

You can install the prerequisites from the file share using an arguments file that consists of switches and corresponding path statements to the programs that need to be installed.

When you run PrerequisiteInstaller.exe with an arguments file, the following happens:

1. PrerequisiteInstaller.exe reads the argument file to verify that each switch is valid and that the program identified in the path statement exists.

   Note:
   If you specify an argument, PrerequisiteInstaller.exe ignores the arguments file and only processes the command-line argument.
2. PrerequisiteInstaller.exe scans the local system to determine if any of the prerequisites are already installed.
3. PrerequisiteInstaller.exe installs the programs in the argument file and returns one of the following exit codes:
   - 0 - Success
   - 1 – Another instance of this application is already running
   - 2 – Invalid command line parameter
   - 1001 – A pending restart blocks installation
   - 3010 – A restart is needed

4. If a prerequisite requires a restart, a 3010 code is generated and you are prompted to click **Finish** to restart the system. The behavior of the installer after a 3010 code is different depending on which of the following conditions exist on the computer:
   - If Windows Server 2008 Service Pack 2 (SP2) is already installed on the system, the 3010 code is generated and the remaining prerequisites are installed. After the last prerequisite is installed you are prompted to restart the system.
   - If Windows Server 2008 SP2 is installed on the system by PrerequisiteInstaller.exe, the installer generates the 3010 code, and the installation of the remaining prerequisites is skipped. You are prompted to restart the system.

     After the system restarts, PrerequisiteInstaller.exe starts running again because the startup file that is created before the restart contains a /continue flag.

     After a restart, PrerequisiteInstaller.exe ignores the arguments file and attempts to download and install the remaining prerequisites from the Internet. For more information, see Known issues

Use the following procedure to create an arguments file.

To create an arguments file

1. Using a text editor, create a new text document named PrerequisiteInstaller.Arguments.txt. Save this file to the same location as PrerequisiteInstaller.exe. This file will contain the switches and arguments that are used when you run the Microsoft SharePoint Products Preparation Tool.
2. Using a text editor, edit PrerequisiteInstaller.Arguments.txt and provide file paths to the installation source for each prerequisite switch, using the following syntax:

   /switch: <*path*>

   Where */switch* is a valid switch and <*path*> is a path to the installation source.

   The following example shows a complete arguments file that uses a file share as a common installation point.

   /SQLNCli: "\\o14-sf-admin\SP_prereqs\sqlncli.msi"
   /ChartControl: "\\o14-sf-admin\SP_prereqs\MSChart.exe"
   /W2K8SP2: "\\o14-sf-admin\SP_prereqs\Windows6.0-KB948465-X64.exe"
   /NETFX35SP1: "\\o14-sf-admin\SP_prereqs\dotnetfx35setup.exe"
   /PowerShell: "\\o14-sf-admin\SP_prereqs\Windows6.0-KB968930-x64.msu"
   /KB976394: "\\o14-sf-admin\SP_prereqs\Windows6.0-KB976394-x64.msu"
   /KB976462: "\\o14-sf-admin\SP_prereqs\Windows6.1-KB976462-v2-x64.msu"
   /IDFX: "\\o14-sf-admin\SP_prereqs\Windows6.0-KB974405-x64.msu"
   /Sync: "\\o14-sf-admin\SP_prereqs\Synchronization.msi"
   /FilterPack: "\\o14-sf-admin\SP_prereqs\FilterPackx64.exe"
   /ADOMD: "\\o14-sf-admin\SP_prereqs\SQLSERVER2008_ASADOMD10.msi"

/ReportingServices: "\\o14-sf-admin\SP_prereqs\rsSharePoint.msi"
/Speech: "\\o14-sf-admin\SP_prereqs\SpeechPlatformRuntime.msi"
/SpeechLPK: "\\o14-sf-admin\SP_prereqs\MSSpeech_SR_en-US_TELE.msi"

Important:
For readability, the switches and path statements in the preceding example are displayed on separate lines. When you actually create a PrerequisitesInstaller.Arguments.txt file do not use line breaks, but separate each switch and path statement by a space. For more information, see Known issues.

3. After you finish editing PrerequisiteInstaller.Arguments.txt, save your edits, and verify that this file is in the same directory as PrerequisiteInstaller.exe.

Use the following procedure to install the prerequisites.

To install the prerequisites using an arguments file

1. Run PrerequisiteInstaller.exe from the command prompt to install the prerequisites.

   Caution:
   If you are prompted to click **Finish** to restart the system, do not do so. Click **Cancel**. For more information, see Known issues before proceeding to the next step.
2. Restart the system manually.
3. Run PrerequisiteInstaller.exe from the command prompt.

Known issues

There are two known issues that affect the use of an arguments file:

- Using line breaks in the arguments file

  If you create an arguments file and use line breaks to put each switch and argument on a separate line, the prerequisite installer fails. The workaround is to enter all the switch and argument pairs on a single line.
- After a computer restart, the arguments file is not used

  After a restart, PrerequisiteInstaller.exe executes the startup command file, which contains a /continue flag. The /continue flag forces the installer to ignore the arguments file.

  You must prevent a restart by deleting the startup task in this command file using one of the following options:

  Option 1

  a. Run PrerequisiteInstaller.exe by double-clicking it. The program will display the first screen with the list of prerequisites.

  b. Click **Cancel**. PrerequisiteInstaller.exe deletes the startup task.

Option 2

a. From the **Start** menu, choose **Run** and then type **regedit** to open the registry.

b. Open the key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders.

c. Check the value for "Common Startup". This shows the directory where the startup tasks are listed.

d. Close the registry editor without making any changes.

e. Navigate to the startup directory, which is usually <systemdir>\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup.

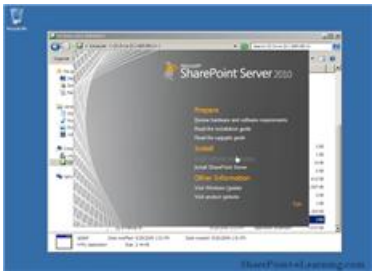
f. Delete the startup task by deleting "SharePointServerPreparationToolStartup_0FF1CE14-0000-0000-0000-000000000000.cmd".

Installing SharePoint Server 2010

This video is authored by Asif Rehmani, SharePoint MVP, MCT, and president of SharePoint-Videos.com.

For more videos from Asif, see http://www.Sharepoint-Videos.com (*http://go.microsoft.com/fwlink/?LinkId=187074*).

This video illustrates how to install and configure Microsoft SharePoint Server 2010 and how to create your first Site Collection.



Watch the video | Right-click the link and click **Save Target As** to download the video.

Related resources

Resource	Description
Deployment for SharePoint Server 2010	Learn about deployment scenarios, step-by-step installation instructions, and post-installation configuration for SharePoint Server 2010.
SharePoint-Videos.com (*http://go.microsoft.com/fwlink/?LinkId=187074*)	Find more videos from Asif Rehmani.
SharePoint Products Tech Center (*http://go.microsoft.com/fwlink/?LinkID=177939*)	Find details about related technologies, downloads, and additional resources.

See Also

Video demos and training for SharePoint Server 2010 (*http://technet.microsoft.com/library/7bd43d63-26e9-45b7-b1bb-f8775a260709(Office.14).aspx*)

Prepare for deployment (SharePoint Server 2010)

This section provides essential steps to take before you deploy Microsoft SharePoint Server 2010.

In this section:

- Hardware and software requirements (SharePoint Server 2010)

  This article provides the minimum hardware and software requirements necessary to install SharePoint Server 2010.

- Administrative and service accounts required for initial deployment (SharePoint Server 2010)

  This article provides information about the administrative and service accounts that are required for an initial SharePoint Server 2010 deployment. Additional accounts and permissions are required to fully implement all aspects of a production farm.

- Harden SQL Server for SharePoint environments (SharePoint Server 2010)

  This article describes how to harden Microsoft SQL Server for Microsoft SharePoint 2010 Products environments.

See Also

Deployment overview (SharePoint Server 2010)

Deployment scenarios (SharePoint Server 2010)

Hardware and software requirements (SharePoint Server 2010)

This article lists the minimum hardware and software requirements to install and run Microsoft SharePoint Server 2010.

Important:
If you contact Microsoft technical support about a production system that does not meet the minimum hardware specifications described in this document, support will be limited until the system is upgraded to the minimum requirements.

In this article:

- Overview
- Hardware requirements—Web servers, application servers, and single server installations
- Hardware requirements—Database servers
- Software requirements
- Access to applicable software

Overview

Microsoft SharePoint Server 2010 provides for a number of installation scenarios. Currently, these installations include single server with built-in database installations and single-server or multiple-server farm installations.

If you plan on installing Microsoft Project Server 2010 with SharePoint Server 2010, see Hardware and software requirements (Project Server 2010) (*http://technet.microsoft.com/library/32d82f51-546a-42a3-ade5-54cc4dfdcb87(Office.14).aspx*). Especially note the supported Web browsers for Project Web App users.

Hardware requirements—Web servers, application servers, and single server installations

The requirements in the following table apply both to installations on a single server with a built-in database and to servers running SharePoint Server 2010 in a multiple server farm installation.

Component	Minimum requirement
Processor	64-bit, four cores
RAM	• 4 GB for developer or evaluation use

Component	Minimum requirement
	• 8 GB for production use in a single server or multiple server farm
Hard disk	80 GB for system drive For production use, you need additional free disk space for day-to-day operations. Maintain twice as much free space as you have RAM for production environments. For more information, see Capacity management and sizing for SharePoint Server 2010 (*http://technet.microsoft.com/library/031b0634-bf99-4c23-8ebf-9d58b6a8e6ce(Office.14).aspx*).

Hardware requirements—Database servers

The requirements in the following table apply to database servers in production environments with multiple servers in the farm.

Note:
Our definitions of small and medium deployments are those described in the "Reference Architectures" section in Capacity management and sizing for SharePoint Server 2010 (*http://technet.microsoft.com/library/031b0634-bf99-4c23-8ebf-9d58b6a8e6ce(Office.14).aspx*).

Component	Minimum requirement
Processor	• 64-bit, four cores for small deployments • 64-bit, eight cores for medium deployments
RAM	• 8 GB for small deployments • 16 GB for medium deployments For large deployments, see the "Estimate memory requirements" section in Storage and SQL Server capacity planning and configuration (SharePoint Server 2010) (*http://technet.microsoft.com/library/a96075c6-d315-40a8-a739-49b91c61978f(Office.14).aspx*). **Note:** These values are higher than those recommended as the minimum values for SQL Server because of the distribution of data required for a SharePoint Products 2010 environment. For more information about SQL Server system requirements, see Hardware and Software Requirements for Installing SQL Server 2008 (*http://go.microsoft.com/fwlink/?LinkId=129377*).
Hard disk	80 GB for system drive Hard disk space is dependent on the size of your SharePoint content. For information about estimating the size of content and other databases for your

Component	Minimum requirement
	deployment, see Storage and SQL Server capacity planning and configuration (SharePoint Server 2010) (*http://technet.microsoft.com/library/a96075c6-d315-40a8-a739-49b91c61978f(Office.14).aspx*).

Software requirements

The requirements in the following tables apply to single server with built-in database installations and server farm installations that include a single server and multiple servers in the farm.

Important:

SharePoint Server 2010 does not support single label domain names. For more information, see Information about configuring Windows for domains with single-label DNS names (*http://support.microsoft.com/kb/300684*).

The Microsoft SharePoint Products Preparation Tool — which you access from the SharePoint Server 2010 Start page — can assist you in the installation of the software prerequisites for SharePoint Server 2010. Ensure that you have an Internet connection, because some of these prerequisites are installed from the Internet. For more information, see Deploy a single server with SQL Server (SharePoint Server 2010), Deploy a single server with a built-in database (SharePoint Server 2010), and Multiple servers for a three-tier farm (SharePoint Server 2010).

Minimum requirements

Environment	Minimum requirement
Database server in a farm	One of the following: • The 64-bit edition of Microsoft SQL Server 2008 R2. • The 64-bit edition of Microsoft SQL Server 2008 with Service Pack 1 (SP1) and Cumulative Update 2. From the Cumulative update package 2 for SQL Server 2008 Service Pack 1 (*http://go.microsoft.com/fwlink/?LinkId=165962*) page, click the **View and request hotfix downloads** link and follow the instructions. On the Hotfix Request page, download the SQL_Server_2008_SP1_Cumulative_Update_2 file. When you install Microsoft SQL Server 2008 SP1 on Windows Server 2008 R2, you might receive a compatibility warning. You can disregard this warning and continue with your installation. **Note:** We do not recommend that you use CU3 or CU4, but instead CU2, CU5, or a later CU than CU5. For more information, see Cumulative update package 5 for SQL Server 2008 (*http://go.microsoft.com/fwlink/?LinkId=196928*). Download the SQL_Server_2008_RTM_CU5_SNAC file.

Environment	Minimum requirement
	• The 64-bit edition of Microsoft SQL Server 2005 with Service Pack 3 (SP3). From the Cumulative update package 3 for SQL Server 2005 Service Pack 3 (*http://go.microsoft.com/fwlink/?LinkId=165748*) page, click the **View and request hotfix downloads** link and follow the instructions. On the Hotfix Request page, download the SQL_Server_2005_SP3_Cumulative_Update_3 file. For more information about choosing a version of SQL Server, see SQL Server 2008 R2 and SharePoint 2010 Products: Better Together (white paper) (SharePoint Server 2010) (*http://technet.microsoft.com/library/665876e1-2706-42ad-bd76-8e4d1da0ce92(Office.14).aspx*).
Single server with built-in database	• The 64-bit edition of Windows Server 2008 Standard, Enterprise, Data Center, or Web Server with SP2, or the 64-bit edition of Windows Server 2008 R2 Standard, Enterprise, Data Center, or Web Server. If you are running Windows Server 2008 without SP2, the Microsoft SharePoint Products Preparation Tool installs Windows Server 2008 SP2 automatically. **Note:** You must download an update for Windows Server 2008 and Windows Server 2008 R2 before you run Setup. The update is a hotfix for the .NET Framework 3.5 SP1 that is installed by the Preparation tool. It provides a method to support token authentication without transport security or message encryption in WCF. For more information and links, see the "Access to Applicable Software" section later in this article. • KB979917 - QFE for Sharepoint issues - Perf Counter fix & User Impersonation (*http://go.microsoft.com/fwlink/?LinkId=192577*) • For Windows Server 2008 SP2, download the Windows6.0-KB979917-x64.msu (Vista) file. • For Windows Server 2008 R2, download the Windows6.1-KB979917-x64.msu (Win7) file. For information, see the related KB article Two issues occur when you deploy an ASP.NET 2.0-based application on a server that is running IIS 7.0 or IIS 7.5 in Integrated mode (*http://go.microsoft.com/fwlink/?LinkId=192578*). **The preparation tool installs the following prerequisites:** • Web Server (IIS) role • Application Server role • Microsoft .NET Framework version 3.5 SP1 • SQL Server 2008 Express with SP1 • Microsoft Sync Framework Runtime v1.0 (x64)

Environment	Minimum requirement
	• Microsoft Filter Pack 2.0 • Microsoft Chart Controls for the Microsoft .NET Framework 3.5 • Windows PowerShell 2.0 • SQL Server 2008 Native Client • Microsoft SQL Server 2008 Analysis Services ADOMD.NET • ADO.NET Data Services Update for .NET Framework 3.5 SP1 • A hotfix for the .NET Framework 3.5 SP1 that provides a method to support token authentication without transport security or message encryption in WCF. • Windows Identity Foundation (WIF) **Note:** If you have Microsoft "Geneva" Framework installed, you must uninstall it before you install the Windows Identity Foundation (WIF).

Front-end Web servers and application servers in a farm	• The 64-bit edition of Windows Server 2008 Standard, Enterprise, Data Center, or Web Server with SP2, or the 64-bit edition of Windows Server 2008 R2 Standard, Enterprise, Data Center, or Web Server. If you are running Windows Server 2008 with SP1, the Microsoft SharePoint Products Preparation Tool installs Windows Server 2008 SP2 automatically.
	
	Note:
	You must download an update for Windows Server 2008 and Windows Server 2008 R2 before you run Setup. The update is a hotfix for the .NET Framework 3.5 SP1 that is installed by the Preparation tool. It provides a method to support token authentication without transport security or message encryption in WCF. For more information and links, see the "Access to Applicable Software" section.
	• KB979917 - QFE for Sharepoint issues - Perf Counter fix & User Impersonation (*http://go.microsoft.com/fwlink/?LinkId=192577*)
	  • For Windows Server 2008 SP2, download the Windows6.0-KB979917-x64.msu (Vista) file.
	  • For Windows Server 2008 R2, download the Windows6.1-KB979917-x64.msu (Win7) file.
	For information, see the related KB article Two issues occur when you deploy an ASP.NET 2.0-based application on a server that is running IIS 7.0 or IIS 7.5 in Integrated mode (*http://go.microsoft.com/fwlink/?LinkId=192578*).
	
	The preparation tool installs the following prerequisites:

Environment	Minimum requirement
	• Web Server (IIS) role • Application Server role • Microsoft .NET Framework version 3.5 SP1 • Microsoft Sync Framework Runtime v1.0 (x64) • Microsoft Filter Pack 2.0 • Microsoft Chart Controls for the Microsoft .NET Framework 3.5 • Windows PowerShell 2.0 • SQL Server 2008 Native Client • Microsoft SQL Server 2008 Analysis Services ADOMD.NET • ADO.NET Data Services Update for .NET Framework 3.5 SP1 • A hotfix for the .NET Framework 3.5 SP1 that provides a method to support token authentication without transport security or message encryption in WCF. • Windows Identity Foundation (WIF) **Note:** If you have Microsoft "Geneva" Framework installed, you must uninstall it before you install the Windows Identity Foundation (WIF).
Client computer	• A supported browser. For more information, see Plan browser support (SharePoint Server 2010) (*http://technet.microsoft.com/library/ff6c5b8c-59bd-4079-8f0b-de4f8b4e0a86(Office.14).aspx*).

Optional software

Environment	Optional software
Single server with built-in database and front-end Web servers and application servers in a farm	• Microsoft SQL Server 2008 R2 to work with PowerPivot workbooks. For more information, see Microsoft SQL Server 2008 R2 (*http://go.microsoft.com/fwlink/?LinkID=179611*). • Windows 7 or Windows Vista. For more information, see Setting Up the Development Environment for SharePoint Server (*http://go.microsoft.com/fwlink/?LinkID=164557*). • SQL Server Remote BLOB Store installation package from the Feature Pack for Microsoft SQL Server 2008 R2. For the download, go to the Download Center (*http://go.microsoft.com/fwlink/?LinkID=177388*). **The preparation tool installs the following optional software:**

Environment	Optional software
	• Microsoft SQL Server 2008 R2 Reporting Services Add-in for Microsoft SharePoint Technologies 2010 (SSRS) to use Access Services for SharePoint Server 2010. For the download, go to the Download Center (*http://go.microsoft.com/fwlink/?LinkID=192588*). • Microsoft Server Speech Platform to make phonetic name matching work correctly for SharePoint Search 2010.
Client computer	• Microsoft Office 2010 client. For more information, see Microsoft Office 2010 (*http://go.microsoft.com/fwlink/?LinkId=195843*). • Microsoft Silverlight 3.

Access to applicable software

To install Windows Server 2008, Microsoft SQL Server, or SharePoint Server, you can go to the Web sites listed in this section. You can install most software prerequisites through the SharePoint Server Start page. The software prerequisites are also available from Web sites listed in this section. The Web Server (IIS) role and the Application Server role can be enabled manually in Server Manager.

In scenarios where installing prerequisites directly from the Internet is not possible or not feasible, you can install the prerequisites from a network share. For more information, see Install prerequisites from a network share (SharePoint Server 2010).

- SharePoint Server 2010 Standard Trial (*http://go.microsoft.com/fwlink/?LinkId=197413*)
- SharePoint Server 2010 Enterprise Trial (*http://go.microsoft.com/fwlink/?LinkId=197414*)
- 2010 Server Language Packs for SharePoint Server 2010, Project Server 2010, Search Server 2010, and Office Web Apps 2010 (*http://go.microsoft.com/fwlink/?LinkId=197415*)
- Windows Server 2008 R2 and SharePoint Server 2010: Better Together (white paper) (*http://technet.microsoft.com/library/849bbd03-0914-4464-b34d-24fb9eed3564(Office.14).aspx*)
- Business Productivity at Its Best: Microsoft Office 2010 and SharePoint Server 2010 Better Together (white paper) (*http://technet.microsoft.com/library/7cb3da78-28e9-47e5-9218-fcf9d4801205(Office.14).aspx*)
- Windows Server 2008 (*http://go.microsoft.com/fwlink/?LinkId=197426*)
- Windows Server 2008 R2 (*http://go.microsoft.com/fwlink/?LinkId=197428*)
- SQL Server 2008 R2 (*http://go.microsoft.com/fwlink/?LinkId=197429*)
- SQL Server 2008 (*http://go.microsoft.com/fwlink/?LinkID=179611*)
- SQL Server 2005 (*http://go.microsoft.com/fwlink/?LinkId=197431*)
- Microsoft SQL Server 2008 SP1 (*http://go.microsoft.com/fwlink/?LinkId=166490*)
- Cumulative update package 2 for SQL Server 2008 Service Pack 1 (*http://go.microsoft.com/fwlink/?LinkId=165962*)

- Cumulative update package 5 for SQL Server 2008 (*http://go.microsoft.com/fwlink/?LinkId=197434*). Download the SQL_Server_2008_RTM_CU5_SNAC file.
- Microsoft SQL Server 2005 SP3 (*http://go.microsoft.com/fwlink/?LinkId=166496*)
- Cumulative update package 3 for SQL Server 2005 Service Pack 3 (*http://go.microsoft.com/fwlink/?LinkId=165748*)
- Microsoft Windows Server 2008 SP2 (*http://go.microsoft.com/fwlink/?LinkId=166500*)
- Windows Server 2008 with SP 2 FIX: A hotfix that provides a method to support the token authentication without transport security or message encryption in WCF is available for the .NET Framework 3.5 SP1 (*http://go.microsoft.com/fwlink/?LinkID=160770*)
- Windows Server 2008 R2 FIX: A hotfix that provides a method to support the token authentication without transport security or message encryption in WCF is available for the .NET Framework 3.5 SP1 (*http://go.microsoft.com/fwlink/?LinkID=166231*)
- Microsoft .NET Framework 3.5 Service Pack 1 (*http://go.microsoft.com/fwlink/?LinkId=131037*)
- Microsoft SQL Server 2008 Express Edition Service Pack 1 (*http://go.microsoft.com/fwlink/?LinkId=166503*)
- Windows Identity Foundation for Windows Server 2008 (*http://go.microsoft.com/fwlink/?LinkID=160381*)
- Windows Identity Foundation for Windows Server 2008 R2 (*http://go.microsoft.com/fwlink/?LinkID=166363*)
- Microsoft Sync Framework v1.0 (*http://go.microsoft.com/fwlink/?LinkID=141237*)
- Microsoft Office 2010 Filter Packs (*http://go.microsoft.com/fwlink/?LinkId=191851*)
- Microsoft Chart Controls for Microsoft .NET Framework 3.5 (*http://go.microsoft.com/fwlink/?LinkID=141512*)
- Windows PowerShell 2.0 (*http://go.microsoft.com/fwlink/?LinkId=161023*)
- Microsoft SQL Server 2008 Native Client (*http://go.microsoft.com/fwlink/?LinkId=166505*)
- Microsoft SQL Server 2008 Analysis Services ADOMD.NET (*http://go.microsoft.com/fwlink/?linkid=160390*)
- KB979917 - QFE for Sharepoint issues - Perf Counter fix & User Impersonation (*http://go.microsoft.com/fwlink/?LinkId=192577*)
  - For Windows Server 2008 SP2, download the Windows6.0-KB979917-x64.msu (Vista) file.
  - For Windows Server 2008 R2, download the Windows6.1-KB979917-x64.msu (Win7) file.
- ADO.NET Data Services Update for .NET Framework 3.5 SP1 (*http://go.microsoft.com/fwlink/?LinkId=163519*) for Windows Server 2008 SP2
- ADO.NET Data Services Update for .NET Framework 3.5 SP1 (*http://go.microsoft.com/fwlink/?LinkId=163524*) for Windows Server 2008 R2 or Windows 7
- Microsoft Silverlight 3 (*http://go.microsoft.com/fwlink/?LinkId=166506*)

- Microsoft Office 2010 (*http://go.microsoft.com/fwlink/?LinkID=195843*)
- SQL Server 2008 R2 Reporting Services Add-in for Microsoft SharePoint Technologies 2010 (*http://go.microsoft.com/fwlink/?LinkId=192588*)
- SQL Server Remote BLOB Store installation package from the Feature Pack for Microsoft SQL Server 2008 R2. For the download, go to the Download Center (*http://go.microsoft.com/fwlink/?LinkID=177388*).
- Microsoft Server Speech Platform (*http://go.microsoft.com/fwlink/?LinkID=179612*)
- Speech recognition language for English (*http://go.microsoft.com/fwlink/?LinkID=179613*)
- Speech recognition language for Spanish (*http://go.microsoft.com/fwlink/?LinkID=179614*)
- Speech recognition language for German (*http://go.microsoft.com/fwlink/?LinkID=179615*)
- Speech recognition language for French (*http://go.microsoft.com/fwlink/?LinkID=179616*)
- Speech recognition language for Japanese (*http://go.microsoft.com/fwlink/?LinkID=179617*)
- Speech recognition language for Chinese (*http://go.microsoft.com/fwlink/?LinkID=179618*)
- Office Communicator 2007 R2 (*http://go.microsoft.com/fwlink/?LinkId=196930*)
- Microsoft SharePoint Designer 2010 (32-bit) (*http://go.microsoft.com/fwlink/?LinkId=196931*)
- Microsoft SharePoint Designer 2010 (64-bit) (*http://go.microsoft.com/fwlink/?LinkId=196932*)

Administrative and service accounts required for initial deployment (SharePoint Server 2010)

This article provides information about the administrative and service accounts that are required for an initial Microsoft SharePoint Server 2010 deployment. Additional accounts and permissions are required to fully implement all aspects of a production farm.

Note:
For a complete list of permissions, see Account permissions and security settings (SharePoint Server 2010) (*http://technet.microsoft.com/library/55b99d80-3fa7-49f0-bdf4-adb5aa959019(Office.14).aspx*).

Required permissions

To deploy SharePoint Server 2010 on a server farm, you must provide credentials for several different accounts. The following table describes the accounts that are used to install and configure SharePoint Server 2010.

Account	Purpose	Requirements
SQL Server service account	The SQL Server service account is used to run SQL Server. It is the service account for the following SQL Server services: • MSSQLSERVER • SQLSERVERAGENT If you do not use the default SQL Server instance, in the Windows Services console, these services will be shown as the following: • MSSQL$InstanceName • SQLAgent$InstanceName	Use either a Local System account or a domain user account. If you plan to back up to or restore from an external resource, permissions to the external resource must be granted to the appropriate account. If you use a domain user account for the SQL Server service account, grant permissions to that domain user account. However, if you use the Network Service or the Local System account, grant permissions to the external resource to the machine account (domain_name\SQL_hostname$). The instance name is arbitrary and was created when Microsoft SQL Server was installed.
Setup user account	The Setup user account is used to run the following: • Setup • SharePoint Products Configuration Wizard	• Domain user account. • Member of the Administrators group on each server on which Setup is run. • SQL Server login on the computer that runs SQL Server.

Account	Purpose	Requirements

		• Member of the following SQL Server security roles:
		  • **securityadmin** fixed server role
		  • **dbcreator** fixed server role
		If you run Windows PowerShell cmdlets that affect a database, this account must be a member of the **db_owner** fixed database role for the database.

Server farm account or database access account	The server farm account is used to perform the following tasks: • Configure and manage the server farm. • Act as the application pool identity for the SharePoint Central Administration Web site. • Run the Microsoft SharePoint Foundation Workflow Timer Service.	• Domain user account. Additional permissions are automatically granted for the server farm account on Web servers and application servers that are joined to a server farm. The server farm account is automatically added as a SQL Server login on the computer that runs SQL Server. The account is added to the following SQL Server security roles: • **dbcreator** fixed server role • **securityadmin** fixed server role • **db_owner** fixed database role for all SharePoint databases in the server farm

Note:
We recommend that you install SharePoint Server 2010 by using least-privilege administration.

Harden SQL Server for SharePoint environments (SharePoint Server 2010)

This article describes how to harden Microsoft SQL Server for Microsoft SharePoint 2010 Products environments.

In this article:

- Summary of hardening recommendations
- Configure a SQL Server instance to listen on a non-default port
- Configure Windows Firewall to block default SQL Server listening ports
- Configure Windows Firewall to open manually assigned ports
- Configure a SQL client alias
- Test the SQL client alias

Summary of hardening recommendations

For secure server farm environments, the recommendation is to do the following:

- Block UDP port 1434.
- Configure named instances of SQL Server to listen on a nonstandard port (other than TCP port 1433 or UDP port 1434).
- For additional security, block TCP port 1433 and reassign the port that is used by the default instance to a different port.
- Configure SQL Server client aliases on all front-end Web servers and application servers in the server farm. After you block TCP port 1433 or UDP port 1434, SQL Server client aliases are necessary on all computers that communicate with the computer running SQL Server.

For more information about these recommendations, see Plan security hardening (SharePoint Server 2010).

Configure a SQL Server instance to listen on a non-default port

Use SQL Server Configuration Manager to change the TCP port that is used by an instance of SQL Server.

1. On the computer running SQL Server, open SQL Server Configuration Manager.
2. In the left pane, expand **SQL Server Network Configuration**.

3. Click the corresponding entry for the instance that you are configuring. The default instance is listed as **Protocols for MSSQLSERVER**. Named instances will appear as **Protocols for named_instance**.
4. In the right pane, right-click **TCP/IP**, and then click **Properties**.
5. Click the **IP Addresses** tab. For every IP address that is assigned to the computer running SQL Server, there is a corresponding entry on this tab. By default, SQL Server listens on all IP addresses that are assigned to the computer.
6. To globally change the port that the default instance is listening on, follow these steps:
   a. For each IP address except **IPAll**, clear all values for both **TCP dynamic ports** and **TCP Port**.
   b. For **IPAll**, clear the value for **TCP dynamic ports**. In the **TCP Port** field, enter the port that you want the instance of SQL Server to listen on. For example, enter 40000.
7. To globally change the port that a named instance is listening on, perform the following steps:
   a. For each IP address including **IPAll**, clear all values for **TCP dynamic ports**. A value of 0 for this field indicates that SQL Server uses a dynamic TCP port for the IP address. A blank entry for this value means that SQL Server will not use a dynamic TCP port for the IP address.
   b. For each IP address except **IPAll**, clear all values for **TCP Port.**
   c. For **IPAll**, clear the value for **TCP dynamic ports**. In the **TCP Port** field, enter the port that you want the instance of SQL Server to listen on. For example, enter 40000.
8. Click **OK**. You will receive a message indicating that the change will not take effect until the SQL Server service is restarted. Click **OK**.
9. Close SQL Server Configuration Manager.
10. Restart the SQL Server service and confirm that the computer running SQL Server is listening on the port that you selected. You can confirm this by looking in the event viewer log after restarting the SQL Server service. Look for an information event similar to the following event:

    Event Type:Information

    Event Source:MSSQL$MSSQLSERVER

    Event Category:(2)

    Event ID:26022

    Date:3/6/2008

    Time:1:46:11 PM

    User:N/A

    Computer:*computer_name*

    Description:

    Server is listening on ['any' <ipv4>50000]

Configure Windows Firewall to block default SQL Server listening ports

1. In **Control Panel**, open **Windows Firewall**. Click **Change settings** to open the **Windows Firewall Settings** dialog box
2. On the **General** tab, click **On**. Ensure that the **Don't allow exceptions** check box is cleared.
3. On the **Exceptions** tab, click **Add Port.**
4. In the **Add a Port** dialog box, enter a name for the port. For example, enter UDP-1434. Then, enter the port number. For example, enter 1434.
5. Click the appropriate option: **UDP** or **TCP**. For example, to block port 1434, click **UDP**. To block port 1433, click **TCP**.
6. Click **Change Scope** and ensure that the scope for this exception is set to **Any computer (including those on the Internet)**.
7. Click **OK**.
8. On the **Exceptions** tab, locate the exception you created. To block the port, clear the check box for this exception. By default, this check box is selected, which means that the port is open.

Configure Windows Firewall to open manually assigned ports

1. Follow steps 1 through 7 in the previous procedure to create an exception for the port you manually assigned to an instance of SQL Server. For example, create an exception for TCP port 40000.
2. On the **Exceptions** tab, locate the exception that you created. Ensure that the check box for the exception is selected. By default, this check box is selected, which means that the port is open.

Note:
For more information about how to use Internet Protocol security (IPsec) to secure communication to and from your computer running SQL Server, see the Microsoft Knowledge Base article 233256: How to Enable IPSec Traffic Through a Firewall (*http://go.microsoft.com/fwlink/?LinkId=76142*).

Configure a SQL client alias

If you block UDP port 1434 or TCP port 1433 on the computer running SQL Server, you must create a SQL Server client alias on all other computers in the server farm. You can use SQL Server client components to create a SQL Server client alias for computers that connect to SQL Server.

1. Run Setup for SQL Server on the target computer, and select the following client components to install:
   a. **Connectivity Components**

b. **Management Tools**

2. Open SQL Server Configuration Manager.
3. In the left pane, click **SQL Native Client Configuration**.
4. In the right pane, right-click **Aliases**, and select **New Alias**.
5. In the **Alias** dialog box, enter a name for the alias and then enter the port number for the database instance. For example, enter SharePoint_*alias*.
6. In the **Port No** field, enter the port number for the database instance. For example, enter 40000. Ensure that the protocol is set to TCP/IP.
7. In the **Server** field, enter the name of the computer running SQL Server.
8. Click **Apply**, and then click **OK**.

Test the SQL client alias

Test connectivity to the computer running SQL Server by using Microsoft SQL Server Management Studio, which is available by installing SQL Server client components.

1. Open SQL Server Management Studio.
2. When you are prompted to enter a server name, enter the name of the alias that you created, and then click **Connect**. If the connection is successful, SQL Server Management Studio is populated with objects that correspond to the remote database.

Note:
To check connectivity to additional database instances from within SQL Server Management Studio, click **Connect**, and then click **Database Engine**.

Deployment scenarios (SharePoint Server 2010)

This section describes how to deploy Microsoft SharePoint Server 2010 on one or more servers to create different topologies that you can use for testing and implementing Microsoft SharePoint Server 2010 solutions at different stages of the deployment life cycle.

In this section:

- Deploy a single server with SQL Server (SharePoint Server 2010)

  This article describes how to install SharePoint Server 2010 on a single server. This deployment uses Microsoft SQL Server and can easily be scaled out to create two- and three-tier farm topologies.

- Deploy a single server with a built-in database (SharePoint Server 2010)

  This article describes how to install SharePoint Server 2010 on a single server. This deployment uses SQL Server Express and is typically used for evaluating SharePoint Server 2010.

- Multiple servers for a three-tier farm (SharePoint Server 2010)

  This article describes how to install SharePoint Server 2010 on multiple servers. This deployment uses Microsoft SQL Server and the resulting three-tier topology provides the foundation for implementing any solution.

- Quick start: Deploy single server in an isolated Hyper-V environment (SharePoint Server 2010)

  This article describes how to use Windows PowerShell to install SharePoint Server 2010 on a single server that uses either SQL Server Express or Microsoft SQL Server. Use the included Windows PowerShell code to quickly install SharePoint Server 2010 in an isolated Hyper-V environment that you can use for to evaluate SharePoint Server 2010.

- Deploy by using DBA-created databases (SharePoint Server 2010)

  This article describes how to deploy Microsoft SharePoint Server 2010 in a farm environment that uses DBA-created databases.

- Deploy in a virtual environment (SharePoint Server 2010)

  This article describes guidance for deploying a virtual environment.

- Install SharePoint Server 2010 by using Windows PowerShell

  This article explains how to use a scripted module to deploy SharePoint Server 2010.

Deploy a single server with SQL Server (SharePoint Server 2010)

This article describes how to perform a clean installation of Microsoft SharePoint Server 2010 on a single server farm.

In this article:

- Overview
- Before you begin
- Install SharePoint Server 2010
- Post-installation steps

Overview

When you install SharePoint Server 2010 on a single server farm, you can configure SharePoint Server 2010 to meet your specific needs. After Setup and the SharePoint Products Configuration Wizard have been completed, you will have installed binaries, configured security permissions, registry settings, the configuration database, and the content database, and installed the SharePoint Central Administration Web site. Next, you can choose to run the Farm Configuration Wizard to configure the farm, select the services that you want to use in the farm, and create the first site collection, or you can manually perform the farm configuration at your own pace.

A single server farm typically consists of one server that runs both Microsoft SQL Server and SharePoint Server 2010. You can deploy SharePoint Server 2010 in a single server farm environment if you are hosting only a few sites for a limited number of users. This configuration is also useful if you want to configure a farm to meet your needs first, and then add servers to the farm at a later stage.

Note:
This guide does not explain how to install SharePoint Server 2010 in a multiple server farm environment or how to upgrade from previous releases of SharePoint Server. For more information, see Multiple servers for a three-tier farm (SharePoint Server 2010). For more information about upgrade, see Upgrading to SharePoint Server 2010 (*http://technet.microsoft.com/library/396c85d9-4b86-484e-9cc5-f6c4d725c578(Office.14).aspx*).

Before you begin

Before you begin deployment, do the following:

- Ensure that you are familiar with the operating-system guidelines described in Performance Tuning Guidelines for Windows Server 2008 (*http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv.mspx*) and Performance Tuning Guidelines for Windows Server 2008 R2 (*http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv-R2.mspx*).
- Ensure that you have met all hardware and software requirements. For more information, see Hardware and software requirements (SharePoint Server 2010).
- Ensure that you perform a clean installation of SharePoint Server 2010. You cannot install the RTM version of SharePoint Server 2010 without first removing the beta version of SharePoint Server 2010.
- Ensure that you are prepared to set up the required accounts with appropriate permissions, as described in Administrative and service accounts required for initial deployment (SharePoint Server 2010).

  Note:
  As a security best practice, we recommend that you install SharePoint Server 2010 by using least-privilege administration.

- Ensure that you have decided which services to use for your Web application, as described in Configure services (SharePoint Server 2010).

  Note:
  If you want to use User Profile Synchronization for your My Sites, then you must configure your permissions correctly in directory services and business systems before you install SharePoint Server 2010.

  For more information, see Plan for profile synchronization (SharePoint Server 2010) (*http://technet.microsoft.com/library/8451dde9-bbd1-4285-bc24-71bd795fb912(Office.14).aspx*) and Configure profile synchronization (SharePoint Server 2010) (*http://technet.microsoft.com/library/144e5f6e-0c9c-4f01-9b1f-26190d527e85(Office.14).aspx*).

Install SharePoint Server 2010

To install and configure SharePoint Server 2010, follow these steps:

1. Run the Microsoft SharePoint Products Preparation Tool, which installs all required prerequisites to use SharePoint Server.

2. Run Setup, which installs binaries, configures security permissions, and edits registry settings for SharePoint Server 2010.
3. Run SharePoint Products Configuration Wizard, which installs and configures the configuration database, the content database, and installs the SharePoint Central Administration Web site.
4. Configure browser settings.
5. Run the Farm Configuration Wizard, which configures the farm, creates the first site collection, and selects the services that you want to use in the farm.
6. Perform post-installation steps.

Important:
To complete the following procedures, you must be a member of the Administrators group on the local computer.

Run the Microsoft SharePoint Products Preparation Tool

Use the following procedure to install software prerequisites for SharePoint Server 2010.

To run the Microsoft SharePoint Products Preparation Tool

1. Insert your SharePoint Server 2010 installation disc.
2. On the SharePoint Server 2010 Start page, click **Install software prerequisites**.

   Note:
   Because the prerequisite installer downloads components from the Microsoft Download Center, you must have Internet access on the computer on which you are installing SharePoint Server.
3. On the Welcome to the Microsoft SharePoint Products Preparation Tool page, click **Next**.
4. On the License Terms for software product page, review the terms, select the **I accept the terms of the License Agreement(s)** check box, and then click **Next**.
5. On the Installation Complete page, click **Finish**.

Run Setup

The following procedure installs binaries, configures security permissions, and edits registry settings for SharePoint Server 2010. At the end of Setup, you can choose to start the SharePoint Products Configuration Wizard, which is described later in this section.

To run Setup

1. On the SharePoint Server 2010 Start page, click **Install SharePoint Server**.
2. On the Enter Your Product Key page, enter your product key, and then click **Continue**.
3. On the Read the Microsoft Software License Terms page, review the terms, select the **I accept**

the terms of this agreement check box, and then click **Continue**.

4. On the Choose the installation you want page, click **Server farm**.
5. On the **Server Type** tab, click **Complete**.
6. Optional: To install SharePoint Server 2010 at a custom location, click the **File Location** tab, and then either type the location or click **Browse** to find the location.
7. Click **Install Now**.
8. When Setup finishes, a dialog box prompts you to complete the configuration of your server. Ensure that the **Run the SharePoint Products and Technologies Configuration Wizard now** check box is selected.
9. Click **Close** to start the configuration wizard.

Note:

If Setup fails, check log files in the Temp folder of the user who ran Setup. Ensure that you are logged in as the user who ran Setup, and then type **%temp%** in the location bar in Windows Explorer. If the path in Windows Explorer resolves to a location that ends in a "1" or "2", you will need to navigate up one level to view the log files. The log file name is SharePoint Server Setup (<*timestamp*>).

Tip:

If you have closed the SharePoint Products Configuration Wizard, you can access it by clicking **Start**, pointing to **All Programs**, and then clicking **Microsoft SharePoint 2010 Products**. If the **User Account Control** dialog box appears, click **Continue**.

Run the SharePoint Products Configuration Wizard

Use the following procedure to install and configure the configuration database and the content database, and install the SharePoint Central Administration Web site.

To run the SharePoint Products Configuration Wizard

1. On the Welcome to SharePoint Products page, click **Next**.
2. In the dialog box that notifies you that some services might need to be restarted during configuration, click **Yes**.
3. On the Connect to a server farm page, click **Create a new server farm**, and then click **Next**.
4. On the Specify Configuration Database Settings page, do the following:
    a. In the **Database server** box, type the name of the computer that is running SQL Server.
    b. In the **Database name** box, type a name for your configuration database or use the default database name. The default name is SharePoint_Config.
    c. In the **Username** box, type the user name of the server farm account. Ensure that you type the user name in the format DOMAIN\user name.

Important:

The server farm account is used to create and access your configuration database. It also acts as the application pool identity account for the SharePoint Central Administration application pool, and it is the account under which the Microsoft SharePoint Foundation Workflow Timer service runs. The SharePoint Products Configuration Wizard adds this account to the SQL Server Login accounts, the SQL Server **dbcreator** server role, and the SQL Server **securityadmin** server role. The user account that you specify as the service account must be a domain user account, but it does not need to be a member of any specific security group on your front-end Web servers or your database servers. We recommend that you follow the principle of least privilege and specify a user account that is not a member of the Administrators group on your front-end Web servers or your database servers.

d. In the **Password** box, type the user password.

5. Click **Next**.
6. On the Specify Farm Security Settings page, type a passphrase, and then click **Next**.

   Ensure that the passphrase meets the following criteria:

   - Contains at least eight characters
   - Contains at least three of the following four character groups:
   - English uppercase characters (from A through Z)
   - English lowercase characters (from a through z)
   - Numerals (from 0 through 9)
   - Nonalphabetic characters (such as !, $, #, %)

   Note:

   Although a passphrase is similar to a password, it is usually longer to enhance security. It is used to encrypt credentials of accounts that are registered in SharePoint Server; for example, the SharePoint Server system account that you provide when you run the SharePoint Products Configuration Wizard. Ensure that you remember the passphrase, because you must use it each time you add a server to the farm.

7. On the Configure SharePoint Central Administration Web Application page, do the following:

   a. Either select the **Specify port number** check box and type the port number you want the SharePoint Central Administration Web application to use, or leave the **Specify port number** check box cleared if you want to use the default port number.

   b. Click either **NTLM** or **Negotiate (Kerberos)**.

8. Click **Next**.
9. On the Completing the SharePoint Products Configuration Wizard page, review your

configuration settings to verify that they are correct, and then click **Next**.

Note:
The **Advanced Settings** option is not available in SharePoint Server 2010.

10. On the Configuration Successful page, click **Finish**.

Note:
If the SharePoint Products Configuration Wizard fails, check the PSCDiagnostics log files, which are located on the drive on which SharePoint Server is installed, in the %COMMONPROGRAMFILES%\Microsoft Shared\Web Server Extensions\14\LOGS folder.

Note:
If you are prompted for your user name and password, you might need to add the SharePoint Central Administration Web site to the list of trusted sites and configure user authentication settings in Internet Explorer. You might also want to disable the Internet Explorer Enhanced Security settings. Instructions for how to configure or disable these settings are provided in the following section.

Tip:
If you see a proxy server error message, you might need to configure your proxy server settings so that local addresses bypass the proxy server. Instructions for configuring proxy server settings are provided in the following section.

Configure browser settings

After you run the SharePoint Products Configuration Wizard, you should ensure that SharePoint Server works properly for local administrators in your environment by configuring additional settings in Internet Explorer.

Note:
If local administrators are not using Internet Explorer, you might need to configure additional settings. For information about supported browsers, see Plan browser support (SharePoint Server 2010) (*http://technet.microsoft.com/library/ff6c5b8c-59bd-4079-8f0b-de4f8b4e0a86(Office.14).aspx*).

If you are prompted for your user name and password, perform the following procedures:

- Add the SharePoint Central Administration Web site to the list of trusted sites
- Disable Internet Explorer Enhanced Security settings

If you receive a proxy server error message, perform the following procedure:

- Configure proxy server settings to bypass the proxy server for local addresses

For more information, see Getting Started with IEAK 8 (*http://go.microsoft.com/fwlink/?LinkId=151359*).

To add the Central Administration Web site to the list of trusted sites

1. In Internet Explorer, on the **Tools** menu, click **Internet Options**.
2. On the **Security** tab, in the **Select a zone to view or change security settings** area, click **Trusted Sites**, and then click **Sites**.
3. Clear the **Require server verification (https:) for all sites in this zone** check box.
4. In the **Add this Web site to the zone** box, type the URL to your site, and then click **Add**.
5. Click **Close** to close the **Trusted Sites** dialog box.
6. Click **OK** to close the **Internet Options** dialog box.

To disable Internet Explorer Enhanced Security settings

1. Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Server Manager**.
2. In **Server Manager**, select the root of **Server Manager**.
3. In the **Security Information** section, click **Configure IE ESC**.

   The **Internet Explorer Enhanced Security Configuration** dialog box opens.
4. In the **Administrators** section, click **Off** to disable the Internet Explorer Enhanced Security settings, and then click **OK**.

To configure proxy server settings to bypass the proxy server for local addresses

1. In Internet Explorer, on the **Tools** menu, click **Internet Options**.
2. On the **Connections** tab, in the **Local Area Network (LAN) settings** area, click **LAN Settings**.
3. In the **Automatic configuration** area, clear the **Automatically detect settings** check box.
4. In the **Proxy Server** area, select the **Use a proxy server for your LAN** check box.
5. Type the address of the proxy server in the **Address** box.
6. Type the port number of the proxy server in the **Port** box.
7. Select the **Bypass proxy server for local addresses** check box.
8. Click **OK** to close the **Local Area Network (LAN) Settings** dialog box.
9. Click **OK** to close the **Internet Options** dialog box.

Run the Farm Configuration Wizard

You have now completed Setup and the initial configuration of SharePoint Server 2010. You have created the SharePoint Central Administration Web site. You can now create your farm and sites, and you can select services by using the Farm Configuration Wizard.

To run the Farm Configuration Wizard

1. On the SharePoint Central Administration Home page, under **Quick Launch**, click **Configuration Wizards**, and then click **Launch the Farm Configuration Wizard**.
2. On the Help Make SharePoint Better page, click one of the following options, and then click **OK**:
   - **Yes, I am willing to participate (Recommended.)**
   - **No, I don't want to participate.**
3. On the Configure your SharePoint farm page, click **Walk me through the settings using this wizard**, and then click **Next**.
4. On the Configure your SharePoint farm page, in the **Service Account** section, click the service account option that you want to use to configure your services.

   Note

   For security reasons, we recommend that you use a different account from the farm administrator account to configure services in the farm.

   If you decide to use an existing managed account — that is, an account that SharePoint Server is aware of — ensure that you click that option before you continue.
5. Select the services that you want to use in the farm, and then click **Next**.

   Note:
   For more information, see Configure services (SharePoint Server 2010). If you are using Microsoft Office Web Apps, see Office Web Apps (Installed on SharePoint 2010 Products) (*http://technet.microsoft.com/library/8a58e6c2-9a0e-4355-ae41-4df25e5e6eee(Office.14).aspx*).
6. On the Create Site Collection page, do the following:
   a. In the **Title and Description** section, in the **Title** box, type the name of your new site.
   b. Optional: In the **Description** box, type a description of what the site contains.
   c. In the **Web Site Address** section, select a URL path for the site.
   d. In the **Template Selection** section, in the **Select a template** list, select the template that you want to use for the top-level site in the site collection.

      Note:
      To view a template or a description of a template, click any template in the **Select a template** list.
7. Click **OK**.
8. On the Configure your SharePoint farm page, review the summary of the farm configuration, and then click **Finish**.

Post-installation steps

After you install and configure SharePoint Server 2010, your browser window opens to the Central Administration Web site of your new SharePoint site. Although you can start adding content to the site or customizing the site, we recommend that you first perform the following administrative tasks by using the SharePoint Central Administration Web site.

- **Configure usage and health data collection** You can configure usage and health data collection in your server farm. The system writes usage and health data to the logging folder and to the logging database. For more information, see Configure usage and health data collection (SharePoint Server 2010).
- **Configure diagnostic logging** You can configure diagnostic logging that might be required after initial deployment or upgrade. The default settings are sufficient for most situations, but depending upon the business needs and lifecycle of the farm, you might want to change these settings. For more information, see Configure diagnostic logging (SharePoint Server 2010).
- **Configure incoming e-mail** You can configure incoming e-mail so that SharePoint sites accept and archive incoming e-mail. You can also configure incoming e-mail so that SharePoint sites can archive e-mail discussions as they happen, save e-mailed documents, and show e-mailed meetings on site calendars. In addition, you can configure the SharePoint Directory Management Service to provide support for e-mail distribution list creation and management. For more information, see Configure incoming e-mail (SharePoint Server 2010).
- **Configure outgoing e-mail** You can configure outgoing e-mail so that your Simple Mail Transfer Protocol (SMTP) server sends e-mail alerts to site users and notifications to site administrators. You can configure both the "From" e-mail address and the "Reply" e-mail address that appear in outgoing alerts. For more information, see Configure outgoing e-mail (SharePoint Server 2010).
- **Configure a mobile account** You can configure a mobile account so that SharePoint sends text message (SMS) alerts to your, or site users', mobile phones. For more information, see Configure a mobile account (SharePoint Server 2010).
- **Install and configure Remote BLOB Storage** You can install and configure Remote BLOB Storage (RBS) for an instance of SQL Server 2008 that supports a SharePoint farm. For more information, see Install and configure Remote BLOB Storage (RBS) with the FILESTREAM provider (SharePoint Server 2010).
- **Configure Search settings** You can configure Search settings to crawl the content in SharePoint Server 2010. For more information, see Post-installation steps for search (SharePoint Server 2010).
- **Configure and use the Exchange connector** You can create a crawl rule and add a content source for crawling Microsoft Exchange Server public folders. For more information, see Configure and use the Exchange connector (SharePoint Server 2010).

Deploy a single server with a built-in database (SharePoint Server 2010)

This article describes how to perform a clean installation of Microsoft SharePoint Server 2010 on a single server with a built-in database.

In this article:

- Overview
- Before you begin
- Install SharePoint Server 2010
- Post-installation steps

Overview

You can quickly publish a SharePoint site by deploying SharePoint Server 2010 on a single server with a built-in database. This configuration is useful if you want to evaluate SharePoint Server 2010 features and capabilities, such as collaboration, document management, and search. This configuration is also useful if you are deploying a small number of Web sites and you want to minimize administrative overhead. When you deploy SharePoint Server 2010 on a single server with a built-in database by using the default settings, Setup installs Microsoft SQL Server 2008 Express and the SharePoint product, and then the SharePoint Products Configuration Wizard creates the configuration database and content database for your SharePoint sites. Additionally, the SharePoint Products Configuration Wizard installs the SharePoint Central Administration Web site and creates your first SharePoint site collection.

Note:
This article does not describe how to install SharePoint Server 2010 in a farm environment, or how to upgrade from previous releases of SharePoint Server. For more information about installing SharePoint Server 2010 on a single server farm, see Deploy a single server with SQL Server (SharePoint Server 2010). For more information about installing SharePoint Server 2010 on a multiple server farm, see Multiple servers for a three-tier farm (SharePoint Server 2010). For more information about upgrade, see Upgrading to SharePoint Server 2010 (*http://technet.microsoft.com/library/396c85d9-4b86-484e-9cc5-f6c4d725c578(Office.14).aspx*).

Consider the following restrictions of this method of installation:

- You cannot install the single server with built-in database version of SharePoint Server on a domain controller.
- A SQL Server 2008 Express database cannot be larger than 4 GB.

- You cannot use User Profile Synchronization in a single server with built-in database installation. If you want to use User Profile Synchronization, you must use a server farm installation of SharePoint. For more information, see Deploy a single server with SQL Server (SharePoint Server 2010) or Multiple servers for a three-tier farm (SharePoint Server 2010), and Configure profile synchronization (SharePoint Server 2010) (*http://technet.microsoft.com/library/144e5f6e-0c9c-4f01-9b1f-26190d527e85(Office.14).aspx*).

Before you begin

Before you begin deployment, ensure that you have met all hardware and software requirements. For more information, see Hardware and software requirements (SharePoint Server 2010). Also, ensure that you perform a clean installation of SharePoint Server 2010. You cannot install the released version of SharePoint Server 2010 without first removing the beta version of SharePoint Server 2010.

Install SharePoint Server 2010

To install and configure SharePoint Server 2010, follow these steps:

1. Run the Microsoft SharePoint Products Preparation Tool.
2. Run Setup, which installs SQL Server 2008 Express and the SharePoint product.
3. Run SharePoint Products Configuration Wizard, which installs and configures the configuration database, the content database, and installs the SharePoint Central Administration Web site. This wizard also creates your first SharePoint site collection.
4. Configure browser settings.
5. Perform post-installation steps.

Important:
To complete the following procedures, you must be a member of the Administrators group on the local computer.

Run the Microsoft SharePoint Products Preparation Tool

Use the following procedure to install software prerequisites for SharePoint Server 2010.

To run the Microsoft SharePoint Products Preparation Tool

1. Insert your SharePoint Server 2010 installation disc.
2. On the SharePoint Server 2010 Start page, click **Install software prerequisites**.

   Note:
   Because the prerequisite installer downloads components from the Microsoft Download Center, you must have Internet access on the computer on which you are installing SharePoint Server.

3. On the Welcome to the Microsoft SharePoint Products Preparation Tool page, click **Next**.
4. On the Installation Complete page, click **Finish**.

Run Setup

The following procedure installs SQL Server 2008 Express and the SharePoint product. At the end of Setup, you can choose to start the SharePoint Products Configuration Wizard, which is described later in this section.

To run Setup

1. On the SharePoint Server 2010 Start page, click **Install SharePoint Server**.
2. On the Read the Microsoft Software License Terms page, review the terms, select the **I accept the terms of this agreement** check box, and then click **Continue**.
3. On the Choose the installation you want page, click **Standalone**.
4. When Setup finishes, a dialog box prompts you to complete the configuration of your server. Ensure that the **Run the SharePoint Products Configuration Wizard now** check box is selected.
5. Click **Close** to start the configuration wizard.

Note:
If Setup fails, check log files in the Temp folder of the user who ran Setup. Ensure that you are logged in as the user who ran Setup, and then type **%temp%** in the location bar in Windows Explorer. If the path in Windows Explorer resolves to a location that ends in a "1" or "2", you will need to navigate up one level to view the log files. The log file name is SharePoint Server Setup (<*timestamp*>).

Tip:
If you have closed the SharePoint Products Configuration Wizard, you can access it by clicking **Start**, pointing to **All Programs**, and then clicking **Microsoft SharePoint 2010 Products**. If the **User Account Control** dialog box appears, click **Continue**.

Run the SharePoint Products Configuration Wizard

Use the following procedure to install and configure the configuration database and the content database, and install the SharePoint Central Administration Web site. This also creates your first SharePoint site collection.

To run the SharePoint Products Configuration Wizard

1. On the Welcome to SharePoint Products page, click **Next**.
2. In the dialog box that notifies you that some services might need to be restarted during

configuration, click **Yes**.

3. On the Configuration Successful page, click **Finish**.

   Note:
   If the SharePoint Products Configuration Wizard fails, check the PSCDiagnostics log files, which are located on the drive on which SharePoint Server is installed, in the %COMMONPROGRAMFILES%\Microsoft Shared\Web Server Extensions\14\LOGS folder.

4. On the Template Selection page, select one of the following options, and then click **OK**:
   - In the **Template Selection** section, click a predefined template.
   - In the **Solutions Gallery** section, click **Solutions Gallery**, and customize your own site template.
5. On the Set Up Groups for this Site page, specify who should have access to your site, and then either create a new group or use an existing group for these users by doing one of the following:
   - To create a new group, click **Create a new group**, and then type the name of the group and the members you want to be part of this group.
   - To use an existing group, click **Use an existing group**, and then select the user group in the **Item** list.
6. Click **OK**.

Note:

If you are prompted for your user name and password, you might need to add the SharePoint Central Administration Web site to the list of trusted sites and configure user authentication settings in Internet Explorer. You might also want to disable the Internet Explorer Enhanced Security settings. Instructions for how to configure or disable these settings are provided in the following section.

Tip:
If you see a proxy server error message, you might need to configure your proxy server settings so that local addresses bypass the proxy server. Instructions for configuring proxy server settings are provided in the following section.

Configure browser settings

After you run the SharePoint Products Configuration Wizard, you should ensure SharePoint Server 2010 works properly for local administrators in your environment by configuring additional settings in Internet Explorer.

Note:
If local administrators are not using Internet Explorer, you might need to configure additional settings. For information about supported browsers, see Plan browser support (SharePoint Server 2010) (*http://technet.microsoft.com/library/ff6c5b8c-59bd-4079-8f0b-de4f8b4e0a86(Office.14).aspx*).

If you are prompted for your user name and password, perform the following procedures:

- Add the SharePoint Central Administration Web site to the list of trusted sites
- Disable Internet Explorer Enhanced Security settings

If you receive a proxy server error message, perform the following procedure:

- Configure proxy server settings to bypass the proxy server for local addresses

For more information, see Getting Started with IEAK 8 (*http://go.microsoft.com/fwlink/?LinkId=151359*).

To add the Central Administration Web site to the list of trusted sites

1. In Internet Explorer, on the **Tools** menu, click **Internet Options**.
2. On the **Security** tab, in the **Select a zone to view or change security settings** area, click **Trusted Sites**, and then click **Sites**.
3. Clear the **Require server verification (https:) for all sites in this zone** check box.
4. In the **Add this Web site to the zone** box, type the URL to your site, and then click **Add**.
5. Click **Close** to close the **Trusted Sites** dialog box.
6. Click **OK** to close the **Internet Options** dialog box.

To disable Internet Explorer Enhanced Security settings

1. Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Server Manager**.
2. In **Server Manager**, select the root of **Server Manager**.
3. In the **Security Information** section, click **Configure IE ESC**.

   The **Internet Explorer Enhanced Security Configuration** dialog box opens.
4. In the **Administrators** section, click **Off** to disable the Internet Explorer Enhanced Security settings, and then click **OK**.

To configure proxy server settings to bypass the proxy server for local addresses

1. In Internet Explorer, on the **Tools** menu, click **Internet Options**.
2. On the **Connections** tab, in the **Local Area Network (LAN) settings** area, click **LAN Settings**.
3. In the **Automatic configuration** area, clear the **Automatically detect settings** check box.
4. In the **Proxy Server** area, select the **Use a proxy server for your LAN** check box.

5. Type the address of the proxy server in the **Address** box.
6. Type the port number of the proxy server in the **Port** box.
7. Select the **Bypass proxy server for local addresses** check box.
8. Click **OK** to close the **Local Area Network (LAN) Settings** dialog box.
9. Click **OK** to close the **Internet Options** dialog box.

Post-installation steps

After you install SharePoint Server 2010, your browser window opens to the Central Administration Web site of your new SharePoint site. Although you can start adding content to the site or you can start customizing the site, we recommend that you first perform the following administrative tasks by using the SharePoint Central Administration Web site:

- **Configure usage and health data collection** You can configure usage and health data collection in your server farm. The system writes usage and health data to the logging folder and to the logging database. For more information, see Configure usage and health data collection (SharePoint Server 2010).
- **Configure diagnostic logging** You can configure diagnostic logging that might be required after initial deployment or upgrade. The default settings are sufficient for most situations, but depending upon the business needs and lifecycle of the farm, you might want to change these settings. For more information, see Configure diagnostic logging (SharePoint Server 2010).
- **Configure incoming e-mail** You can configure incoming e-mail so that SharePoint sites accept and archive incoming e-mail. You can also configure incoming e-mail so that SharePoint sites can archive e-mail discussions as they happen, save e-mailed documents, and show e-mailed meetings on site calendars. In addition, you can configure the SharePoint Directory Management Service to provide support for e-mail distribution list creation and management. For more information, see Configure incoming e-mail (SharePoint Server 2010).
- **Configure outgoing e-mail** You can configure outgoing e-mail so that your Simple Mail Transfer Protocol (SMTP) server sends e-mail alerts to site users and notifications to site administrators. You can configure both the "From" e-mail address and the "Reply" e-mail address that appear in outgoing alerts. For more information, see Configure outgoing e-mail (SharePoint Server 2010).
- **Configure a mobile account** You can configure a mobile account so that SharePoint sends text message (SMS) alerts to your, or site users', mobile phones. For more information, see Configure a mobile account (SharePoint Server 2010).
- **Install and configure Remote BLOB Storage** You can install and configure Remote BLOB Storage (RBS) for an instance of SQL Server 2008 that supports a SharePoint server farm. For more information, see Install and configure Remote BLOB Storage (RBS) with the FILESTREAM provider (SharePoint Server 2010).

- **Configure search settings** You can configure search settings to crawl the content in SharePoint Server 2010. For more information, see Post-installation steps for search (SharePoint Server 2010).
- **Configure and use the Exchange connector** You can create a crawl rule and add a content source for crawling Microsoft Exchange Server public folders. For more information, see Configure and use the Exchange connector (SharePoint Server 2010).

Multiple servers for a three-tier farm (SharePoint Server 2010)

This article describes how to install Microsoft SharePoint Server 2010 on multiple servers to create a Microsoft SharePoint Server farm deployed across three tiers. The farm consists of two front-end Web servers, an application server, and a database server. The deployment sequence and configurations that are described in this article are based on recommended best practices. The resulting farm configuration is not complex, but provides a fundamental infrastructure for implementing a SharePoint Server solution on similar — or more complex — farms.

The farm is provisioned with Search Service Application; Search is configured to crawl the content that is created as part of this deployment.

In this article:

- Overview
- Prepare the farm servers
- Install SharePoint Server 2010 on the farm servers
- Create and configure the farm
- Add Web servers to the farm
- Configure diagnostic logging and usage and health data collection
- Create and configure a new Search Service Application
- Create a site
- Add content sources and configure a search crawl schedule
- Post-installation steps

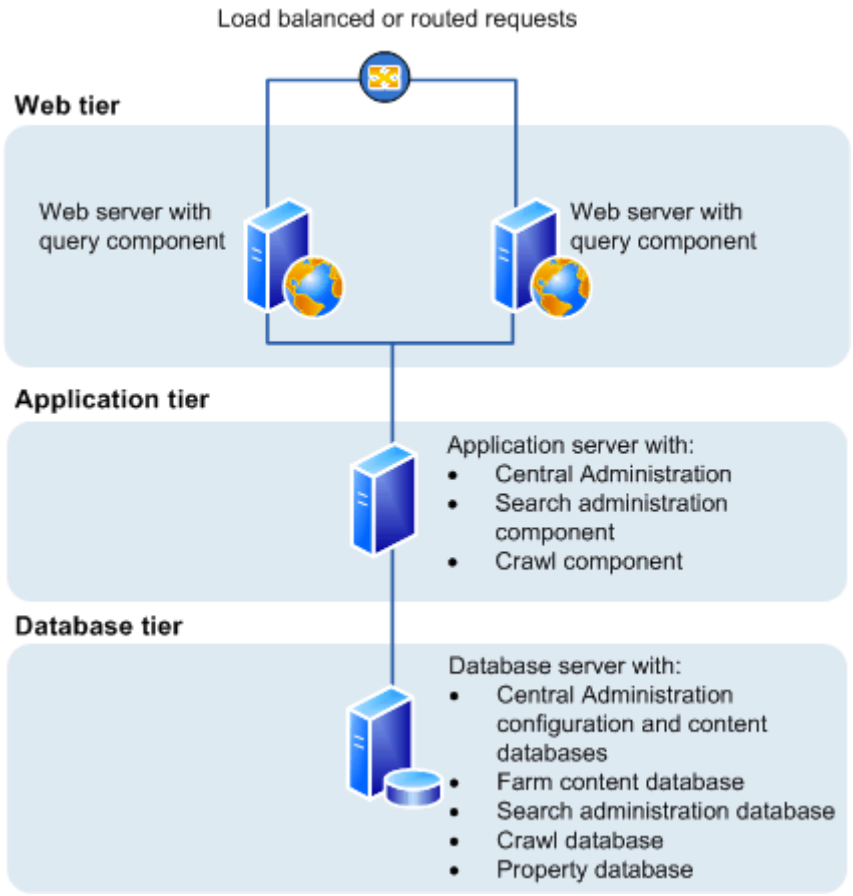
Overview

The basic steps in this deployment are as follows:

- Ensure that you are familiar with the concept of a three-tier topology.
- Ensure that you have done all the planning and preparatory work, such as verifying hardware and software requirements.
- Install the required software updates on all servers that will be part of the farm.
- Install the SharePoint Server prerequisites on servers in the application and Web tiers.
- Install SharePoint Server on the application server and the Web servers.
- Create and configure the SharePoint farm.
- Provision services.
- Complete post-deployment tasks as required.

Topology overview

This topology is typically used for the medium and large farms described in Deployment overview (SharePoint Server 2010). In terms of performance, capacity, and scalability, a three-tier topology is recommended over a two-tier topology. A three-tier topology provides the most efficient physical and logical layout to support scaling out or scaling up, and provides better distribution of services across the member servers of the farm. The following illustration shows the three-tier deployment that is described in this article.



Referring to the preceding illustration, note the following:

- You can add Web servers to the Web tier. These servers can be configured as conventional Web servers to handle user requests, or they can be configured to host dedicated query components or other service components.
- You can add farm servers to the application tier and configure them as dedicated servers that will host the Central Administration Web site or other services on the farm that require dedicated resources or isolation from the Web tier — for example, crawl components, query components, and profile pages.

- You can add database servers to the database tier to implement a stand-alone instance, database mirroring, or a failover cluster. If you are configuring the farm for high availability, database mirroring or a failover cluster is required on the database tier.

Before you begin

Before you begin deployment, do the following:

- Ensure that you are familiar with the operating-system guidelines described in Performance Tuning Guidelines for Windows Server 2008 (*http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv.mspx*) and Performance Tuning Guidelines for Windows Server 2008 R2 (*http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv-R2.mspx*).
- Ensure that you have met all hardware and software requirements. For more information, see Hardware and software requirements (SharePoint Server 2010)
- Ensure that you are prepared to set up the required accounts with appropriate permissions, as described in Administrative and service accounts required for initial deployment (SharePoint Server 2010)

  Note:
  As a security best practice, we recommend that you install SharePoint Server by using least-privilege administration.

Using the Microsoft SharePoint 2010 Products Preparation Tool

The Microsoft SharePoint Products Preparation Tool checks for the presence of prerequisites, and installs and configures any programs that are required. The Microsoft SharePoint Products Preparation Tool requires an Internet connection to download and configure SharePoint Server prerequisites. If you do not have an Internet connection for the farm servers, you will have to obtain installable images for the required software. For installation locations, see Software requirements in "Determine hardware and software requirements (SharePoint Server 2010)."

If you decide to obtain installable images, we recommend that you create an installation point that you can use for storing the images. You can use this installation point for installing future software updates.

Database server

Ensure that SQL Server 2005 or SQL Server 2008 is updated to the required level and configured as follows:

- SQL Server 2005: Local and remote connections are enabled and are configured to use the TCP/IP protocol.
- SQL Server 2008: The TCP/IP protocol is enabled for the network configuration.

In organizations whose database administrators operate independently from SharePoint administrators, you will need to ensure that the correct version of SQL Server is available and updated to the required level. In addition, you will have to request a DBA-created database that is configured for your farm.

Note:
The procedures in this article do not use a DBA-created database; these procedures will be different in a DBA-created database deployment. For more information, see Deploy by using DBA-created databases (SharePoint Server 2010).

Public updates and hotfix packages

Ensure that public updates and the required hotfix packages are installed for the operating system, SQL Server, and SharePoint Server. We recommend that all servers be updated to the same software version before you apply the public updates. The following hotfixes must be applied for this release of SharePoint Server:

- SQL Server 2008: Install this update before deploying the farm. Cumulative update package 2 for SQL Server 2008 Service Pack 1 (*http://support.microsoft.com/kb/970315*)
- After you install the prerequisites on each farm server, you will need to install Error message when you use the "IMSAdminBase::CopyKey" method as a part of the IIS 6.0 Compatibility components in IIS 7.0: "Exception from HRESULT: 0x80070003" (*http://support.microsoft.com/kb/949516*) on the application server first, and then on the two Web servers.

Prepare the farm servers

Before you install SharePoint Server , you must check for and install all the prerequisites on the application server and the Web servers by using the Microsoft SharePoint Products Preparation Tool.

Tip:
If you decide to install prerequisites manually, you can still run the Microsoft SharePoint Products Preparation Tool to verify which prerequisites are required on each server.

Use the following procedure to install prerequisites on each of the farm servers.

To run the preparation tool

1. Download SharePoint Server 2010 (*http://go.microsoft.com/fwlink/?LinkId=168780*) from the product installation location, and then double-click the appropriate executable file.
2. If you install from a bootable image, click **Install software prerequisites** on the splash screen.

   Note:
   You can download all of the prerequisites and install them from a network share. For more information, see Install prerequisites from a network share (SharePoint Server 2010).
3. On the Welcome to the Microsoft SharePoint Products Preparation Tool page, click **Next**.

Note:

The preparation tool may have to restart the local server to complete the installation of some of the prerequisites. The installer will continue to run after the server is restarted, and no manual intervention is required. However, you will have to log back on to the server.

4. On the Installation Complete page, click **Finish**.

Note:

After you complete the Microsoft SharePoint Products Preparation Tool, you must install KB 949516 (*http://go.microsoft.com/fwlink/?LinkId=148917*) and KB 971831 (*http://go.microsoft.com/fwlink/?LinkID=165750*). You might also need to restart the server after installing this hotfix.

Note:

If the error message "Loading this assembly would produce a different grant set from other instances. (Exception from HRESULT: 0x80131401)" is displayed when you start the IIS worker process (w3wp.exe), another service, or a managed application on a server that is also running SharePoint Server 2010, you must install KB963676 (*http://go.microsoft.com/fwlink/?LinkId=151358*). You must restart the computer after you apply this hotfix.

Install SharePoint Server 2010 on the farm servers

After the prerequisites are installed, use the following procedure to install SharePoint Server on each of the farm servers.

To run Setup

1. On the Start page, click **Install Office SharePoint ServerInstall SharePoint Foundation**.
2. On the Read the Microsoft Software License Terms page, review the terms, select the **I accept the terms of this agreement** check box, and then click **Continue**.
3. On the Choose the installation you want page, click **Server Farm**.
4. On the **Server Type** tab, click **Complete**.
5. On the File Location tab, accept the default location or change the installation path, and then click **Install Now**.

Note:

As a best practice, we recommend that you install SharePoint Server on a non-system drive.

6. When Setup finishes, a dialog box prompts you to complete the configuration of your server. Clear the **Run the SharePoint Products and Technologies Configuration Wizard now** check box.

7. Click **Close** to finish Setup.

Create and configure the farm

To create and configure the farm, you run the SharePoint Products Configuration Wizard. This wizard automates several configuration tasks, including creating the configuration database, installing services, and creating the Central Administration Web site. It is recommended that you run the SharePoint Products Configuration Wizard on the server that will host the Central Administration Web site before you run the wizard on the other servers in the farm.

To run the configuration wizard and configure the farm

1. On the server that will host Central Administration (the application server), click **Start**, point to **All Programs**, and then click **Microsoft SharePoint 2010 Products**.
2. In the list of available options, click **SharePoint Products and Technologies Configuration Wizard**.
3. On the Welcome to SharePoint Products and Technologies page, click **Next**.
4. In the dialog box that notifies you that some services might need to be restarted during configuration, click **Yes**.
5. On the Connect to a server farm page, click **Create a new server farm**, and then click **Next**.
6. On the Specify Configuration Database Settings page, do the following:
   a. In the **Database server** box, type the name of the computer that is running SQL Server.
   b. In the **Database name** box, type a name for your configuration database, or use the default database name. The default name is SharePoint_Config.
   c. In the **Username** box, type the user name of the server farm account in DOMAIN\username format.

      Important:
      The server farm account is used to create and access your configuration database. It also acts as the application pool identity account for the SharePoint Central Administration application pool, and it is the account under which the Windows SharePoint Services Timer service runs. The SharePoint Products Configuration Wizard adds this account to the SQL Server Login accounts, the SQL Server **dbcreator** server role, and the SQL Server **securityadmin** server role. The user account that you specify as the service account must be a domain user account, but it does not need to be a member of any specific security group on your Web servers or your database servers. We recommend that you follow the principle of least privilege, and specify a user account that is not a member of the Administrators group on your Web servers or your database servers.

   d. In the **Password** box, type the user password.

7. Click **Next**.
8. On the Specify Farm Security Settings page, type a passphrase, and then click **Next**.

   Ensure that the passphrase meets the following criteria:

   - Contains at least eight characters
   - Contains at least three of the following four character groups:
   - English uppercase characters (from A through Z)
   - English lowercase characters (from a through z)
   - Numerals (from 0 through 9)
   - Nonalphabetic characters (such as !, $, #, %)

   Note:
   Although a passphrase is similar to a password, it is usually longer to enhance security. It is used to encrypt credentials of accounts that are registered in SharePoint Server 2010. For example, the SharePoint Server 2010 system account that you provide when you run the SharePoint Products Configuration Wizard wizard. Ensure that you remember the passphrase, because you must use it each time you add a server to the farm.
9. On the Configure SharePoint Central Administration Web Application page, do the following:
   a. Either select the **Specify port number** check box and type a port number if you want the SharePoint Central Administration Web application to use a specific port number, or leave the **Specify port number** check box cleared if you want to use the default port number.

      Note:
      If you want to access the SharePoint Central Administration Web site from a remote computer, ensure that you allow access to the port number that you configure in this step. You do this by configuring the inbound rule for **SharePoint Central Administration v4** in Windows Firewall with Advanced Security.
   b. Click either **NTLM** or **Negotiate (Kerberos)**.
10. Click **Next**.
11. On the Configuration Successful page, click **Finish**.

    Note:
    If the SharePoint Products Configuration Wizard fails, check the log files on the drive on which SharePoint Server 2010 is installed, which are located in the %COMMONPROGRAMFILES%\Microsoft Shared\Web Server Extensions\14\LOGS folder.
12. The Central Administration Web site will open in a new browser window.

    On the Help Make SharePoint Better page, click one of the following options and then click **OK**.
    a. **Yes, I am willing to participate (Recommended).**

b. **No, I don't wish to participate.**

13. On the Configure your SharePoint farm page, you have the option to use a wizard to configure services or you can decide to configure services manually. For the purpose of this article, we use the manual option. Click **Cancel**.

    The choice you make here is a matter of personal preference. The Farm Configuration Wizard will configure some services automatically when it is run; however, if you configure services manually you have greater flexibility in designing your logical architecture.

    For information about using the wizard to configure services, see Configure services (SharePoint Server 2010). If you are using Microsoft Office Web Apps, see Office Web Apps (Installed on SharePoint 2010 Products) (*http://technet.microsoft.com/library/8a58e6c2-9a0e-4355-ae41-4df25e5e6eee(Office.14).aspx*).

    Important:
    If you are using a DBA-created database you cannot use the Farm Configuration Wizard, you must use SharePoint Products Configuration Wizard.

Add Web servers to the farm

After you create the farm on the application server, you can add the servers for the Web tier by following the same process described earlier in this topic for installing SharePoint Server on the server that hosts Central Administration. The only difference is that during Setup, you will be prompted to join and existing farm. Follow the wizard steps to join the farm.

For additional information about adding servers to a farm, see Add a Web or application server to the farm (SharePoint Server 2010) (*http://technet.microsoft.com/library/0926f63d-8dae-44c0-9e91-51209aa4c3ef(Office.14).aspx*). This article also provides detailed information for the steps in the following procedure.

Configure diagnostic logging and usage and health data collection

After you add the front-end Web servers, configure initial diagnostic logging and usage and health data collection for the farm.

Diagnostic logging can help identify and isolate issues as they occur in your server farm. Accept the default settings when you configure diagnostic logging on new installations. Then, when issues occur in your server farm, you can revisit these settings and adjust the levels accordingly. This will help to identify the cause and isolate the issues. Usage and health reporting can be used to show where diagnostic logging settings deviate from the default values.

For more information about diagnostic and health usage, see:

- Configure diagnostic logging (SharePoint Server 2010)
- Configure usage and health data collection (SharePoint Server 2010)

Use the following procedures to complete the initial configuration of diagnostic logging and usage and health data collection.

Note:
Because this is an initial farm deployment without any benchmark data, default settings are accepted unless otherwise noted.

To configure diagnostic logging

1. On the Central Administration Home page, click **Monitoring**.
2. In the **Reporting** section, click **Configure diagnostic logging**.
3. On the Diagnostic Logging page, verify that **Enable Event Log Flood Protection** is selected. If not, click the corresponding check box to enable this feature.
4. The default location for the trace log is on the drive where you installed SharePoint Server. As a best practice, we recommend that the trace log be stored on a non-system drive.

   Important:
   If you change the trace log path to a non-system drive, this location must exist on all the servers in the farm. Existing or new servers cannot log data if the location does not exist. In addition, you will not be able to add new servers unless the path you specify exists on the new server. You cannot use a network share for logging purposes.
5. Click **OK** to save your changes.

After you finish configuring diagnostic logging, configure usage and health data collection.

To configure usage and health data collection

- On the Central Administration Monitoring page, click **Configure usage and health data collection**.
- Click the check box to enable **Usage Data Collection**.
- Click the check box to enable **Health Data Collection**.
- Click **OK**.

Create and configure a new Search Service Application

As a best practice, we recommend that you install search components on the Web and application tiers in order to optimize performance by distributing the load placed on the servers in the farm. To distribute search components across the farm tiers, you must follow these steps:

- Create a new Search Service Application
- Move the query components to the Web tier

For more information about service applications, see:

- About service applications and services (SharePoint Server 2010) (*http://technet.microsoft.com/library/ba00a200-ea96-45f6-a423-3cfcffba77f8(Office.14).aspx*)
- Service application and service management (SharePoint Server 2010)

To create a Search Service Application

1. On the server that hosts the Central Administration Web site, open Central Administration.
2. Under the **Application Management** section, click **Manage service applications**.
3. On the ribbon, click **New**, and then click **Search Service Application**.
4. On the Create New Search Service Application page, accept the default value for **Name**, or type in a new name for the search service application.
5.
6. For the **Search Service Account**, accept the default for the farm, or create a new account for the Search Service. For more information about farm accounts, see Administrative and service accounts required for initial deployment (SharePoint Server 2010).

   noteDXDOC112778PADS Security Note

   We recommend that you create a new account for the Search Service. If you accept the default account, the Search Service will use the highly privileged farm administrator account. This is not recommended because the content access account, which defaults to the Search Service account, would crawl all the drafts and Help files in your corpus, thus exposing all content in the search results.
7. For the **Application Pool for Search Admin Web Service**, accept the default (**Create new application pool**), and then type a name for the application pool.
8. For the **Application Pool for Search Query and Site Settings Web Service**, use the same application pool that you created for the Search Admin Web Service. Click **Use existing application pool**, and then select the pool name from the drop-down list.
9. Click **OK**.

After you create a Search Service Application, the result is a topology where the search administration and crawl components are configured on the server that hosts the Central Administration Web site. The next step is to move the query components to the servers on the Web tier, the crawl and property store (query) databases are provisioned on the SQL server instance that was specified for the SharePoint Server farm.

Important:

The search administration component cannot be moved to another server: it resides on the server where the Search Service Application was created. There can only be one search administration component per farm.

To move the query component to a Web server

1. On the Central Administration Home page, click **Manage service applications**.
2. On the Services Applications page, click **Search Service Application**.
3. On the Search Administration page, in the **Search Application Topology** section, click **Modify**.
4. On the Topology for Search Service Application: Search Service Application page, in the **Index Partition** section, click **Query component 0**, and then click **Edit Properties**.
5. On the Edit Query Component page, select one of the front-end Web servers from the **Server** drop-down list, and then click **OK**.
6. On the Topology for Search Service Application: Search Service Application page, in the **Index Partition** section, click **Query component 0**, and then click **Add mirror**.

   Note:
   A query component mirror is a replica of the index partition on another server — in this case, on one of the front-end Web servers.
7. On the Add mirror query component page, in the **Server** drop-down list, select the front-end Web server that you identified on the Edit Query Component page, and then click **OK**.
8. Repeat the preceding steps and configure the second front-end Web server as a host for a query component and its mirror.

Create a site

To create a site during this phase of the deployment, you must create a Web application and a site collection. Use the procedures in the following articles to create a Web application by using Central Administration, and then create a top-level Web site that is associated with the Web application.

- Create a Web application

  Important:
  When creating a new Web application or extending the existing Web application into a new zone initially, ensure that the public URL is the URL that end users will use to browse to the Web application. If you are using reverse proxy servers or load balancers, you may also have to add internal URLs for alternate access mapping (AAM). We recommend that you configure AAM before creating a site collection. For more information, see Logical architecture components (SharePoint Server 2010) (*http://technet.microsoft.com/library/aaed3a01-f4dc-4353-abda-0beced2080b6(Office.14).aspx*).
- Create a site collection (SharePoint Server 2010)

Add content sources and configure a search crawl schedule

The final step is to identify a content source for search, and then configure the initial and incremental crawl schedule.

Use the following procedure to add a content source and configure the crawl schedule. In this procedure, the default content source is used for configuring crawl, but you can choose a different content source in your environment.

To add a content source and configure a crawl schedule

1. On the Central Administration Home page, click **Manage service applications** in the Application Management section.
2. On the **Service Applications** tab, click **Search Service Application**
3. On the Search Administration page, in the **Crawling** section, click **Content Sources**.
4. On the Manage Content Sources page, Local SharePoint sites are displayed as the available content source. You can configure crawl for the existing source, or click **New Content Source** to add a content source.

   Note:
   For the purpose of this article, the default source is used for configuring crawl.
5. Click **Local SharePoint sites**.
6. On the Edit Content Source page, click **Create schedule** in the **Full Crawl** section.

   On the Manage Schedules page, configure the type of schedule and the schedule settings that you want to use.
7. On the Edit Content Source page, click **Create schedule** in the **Incremental Crawl** section.

   On the Manage Schedules page, configure the type of schedule and the schedule settings that you want to use.
8. Click **Start full crawl of this content source**.
9. Click **OK**.

Post-installation steps

After you install and configure SharePoint Server 2010, your browser window opens to the Central Administration Web site of your new SharePoint site. Although you can start adding content to the site or customizing the site, we recommend that you first perform the following administrative tasks by using the SharePoint Central Administration Web site.

- **Configure outgoing e-mail** You can configure outgoing e-mail so that your Simple Mail Transfer Protocol (SMTP) server sends e-mail alerts to site users and notifications to site administrators.

You can configure both the "From" e-mail address and the "Reply" e-mail address that appear in outgoing alerts. For more information, see Configure outgoing e-mail (SharePoint Server 2010).

Note:
You can configure incoming e-mail so that SharePoint sites accept and archive incoming e-mail. However, we recommend that you undertake this task after you complete the initial farm deployment and configuration. For more information, see Configure incoming e-mail (SharePoint Server 2010).

- **Configure a mobile account** You can configure a mobile account so that SharePoint sends text message (SMS) alerts to your, or site users', mobile phones. For more information, see Configure a mobile account (SharePoint Server 2010).

Quick start: Deploy single server in an isolated Hyper-V environment (SharePoint Server 2010)

You can use an isolated and secure Hyper-V virtual machine to test the features and behavior of SharePoint Server 2010. This approach uses minimal hardware resources and enables you to isolate the SharePoint Server 2010 test system from a production environment. This isolation is recommended in order to eliminate potential security threats to a corporate network and server environment.

By using the manual steps or the Windows PowerShell 2.0 commands that are provided in this article, you can quickly deploy SharePoint Server 2010 on a single server that uses one of the following databases:

- The built-in SQL Server 2008 Express database that is provided with SharePoint Server
- Microsoft SQL Server 2005 with Service Pack 3 (SP3) and Cumulative Update 3 installed
- Microsoft SQL Server 2008 with Service Pack 1 (SP1) and Cumulative Update 2

Important:
The single server SharePoint Server deployment described in this article is only intended to be used for evaluation and testing purposes, and should not be used in a production environment.

In this article:

- Requirements and recommendations
- Required permissions
- Pre-deployment tasks
- Deploy SharePoint Server 2010 manually
- Deploy SharePoint Server 2010 by using Windows PowerShell scripts

Requirements and recommendations

The following requirements and recommendations for the Hyper-V virtualization server, virtual machine, and the deployment environment only apply to the single server deployment scenario described in this article.

Virtualization server and virtual machine configuration

The following table provides the minimum and recommended configurations for the virtualization server and the virtual machines. These configurations will support the database options that are available for a single server deployment.

Resource	Minimum	Recommended
CPU	Dual processor, 2 gigahertz (GHz)	Dual processor, 2 GHz
Memory	4 gigabytes (GB)	8 GB
Hard drive	Fixed-size virtual hard disk that has a capacity of 40 GB **Tip:** To speed up the creation of a fixed-size virtual hard disk, initially configure the hard disk as dynamically expanding. After you install all the required software (including SharePoint Server), convert the virtual hard disk to a fixed-size hard disk.	Fixed-size virtual hard disk that has a capacity of 80 GB
Network adapter type	Synthetic	Synthetic
Network type	Internal to ensure virtual machine isolation and enable virtualization server-virtual machine communications **Tip:** For ease of access to—and installation of—required and recommended software, use an External network. When you are ready to install SharePoint Server, configure the virtual machines to use an Internal network.	Internal to ensure virtual machine isolation and enable virtualization server-virtual machine communications

The following configuration guidance is provided for the virtualization server:

- The logical-to-virtual processor (core) ratio should be as low as possible, with 1:1 being optimal.
- Using the 1:1 logical-to-virtual processor ratio, you should configure the virtualization server so the total number of processors on the virtual machines is less than the total number of physical cores. For example, if you are using a four-core virtualization server, the best practice is to create three virtual machines that use a single processor, or one virtual machine that has two processors and one virtual machine that uses one processor. Either of these configurations would leave one core free for virtualization server processes.

In addition to the preceding requirements for the virtual environment, review the Hardware and software requirements (SharePoint Server 2010) article before you start deploying SharePoint Server 2010 on the virtual machine.

Deployment environment

A domain is required to deploy SharePoint Server 2010.

If you do not have an isolated virtual domain available to deploy SharePoint Server 2010, you must create a virtual domain on a Hyper-V that is configured to use the following:

- A domain controller with Active Directory Domain Services (AD DS)
- A domain controller with a DNS server

You can deploy SharePoint Server on a domain controller. However, some configuration is required. Start Windows PowerShell with the Run as administrator option and run the following commands to enable deployment on a domain controller:

```
$acl = Get-Acl HKLM:\System\CurrentControlSet\Control\ComputerName

$person = [System.Security.Principal.NTAccount]"Users"

$access = [System.Security.AccessControl.RegistryRights]::FullControl

$inheritance = [System.Security.AccessControl.InheritanceFlags]"ContainerInherit,
ObjectInherit"

$propagation = [System.Security.AccessControl.PropagationFlags]::None

$type = [System.Security.AccessControl.AccessControlType]::Allow

$rule = New-Object System.Security.AccessControl.RegistryAccessRule($person, $access,
$inheritance, $propagation, $type)

$acl.AddAccessRule($rule)

Set-Acl HKLM:\System\CurrentControlSet\Control\ComputerName $acl
```

Required permissions

In order to install SharePoint Server 2010, the logon account that you are using on the virtual machine must be a member of:

- The local Administrators group on the virtual machine
- The SQL Server **dbcreator** fixed server role
- The SQL Server **securityadmin** server role

For more information, see Administrative and service accounts required for initial deployment (SharePoint Server 2010).

Pre-deployment tasks

Complete the following tasks before you deploy SharePoint Server 2010:

- On the virtualization server, create an installation point that contains the SharePoint Server software or provide media, such as an ISO image, that can be accessed from the virtual machine.

- Create a virtual machine that meets the minimum requirements described in the “Requirements and recommendations” section earlier in this article.
- On the virtual machine:
  - Install the operating system and the mandatory and recommended security updates.
  - Install the edition of SQL Server that you want to use if you are not using the built-in version that is provided with SharePoint Server.
  - Install the mandatory and recommended updates for the edition of SQL Server that you install.
  - Configure the Windows Server firewall to enable SQL Server access. For more information, see Configuring the Windows Firewall to Allow SQL Server Access (*http://go.microsoft.com/fwlink/?LinkID=134724*).
  - Review the Hardware and software requirements (SharePoint Server 2010) article to determine the programs and hotfixes that must be obtained and installed before you install SharePoint Server 2010.

Deploy SharePoint Server 2010 manually

For information about how to manually deploy SharePoint Server 2010 on a single server, see Deploy a single server with a built-in database (SharePoint Server 2010) or Deploy a single server with SQL Server (SharePoint Server 2010).

Deploy SharePoint Server 2010 by using Windows PowerShell scripts

You can use Windows PowerShell scripts to deploy SharePoint Server 2010 on a single server.

noteDXDOC112778PADS Security Note

- As a best practice, you should not run unsigned scripts.
- For more information about signing Windows PowerShell scripts, see Windows PowerShell: Sign Here Please (*http://go.microsoft.com/fwlink/?linkid=160357*) in TechNet Magazine. For more information about code signing in general, see Introduction to Code Signing (*http://go.microsoft.com/fwlink/?linkid=59273*) on MSDN. For more information about setting up your own certification authority (CA), see Active Directory Certificate Services (*http://go.microsoft.com/fwlink/?linkid=136444*) in the TechNet Library.

Create and use one of the following Windows PowerShell script files to deploy SharePoint Server on a single server.

- simplesingleserver.ps1: Installs SharePoint Server 2010 using the built-in database to store configuration information and documents.
- simplefarm.ps1: Installs SharePoint Server 2010 using either SQL Server 2005 or SQL Server 2008 to store configuration information and documents.

simplesingleserver.ps1

This script deploys SharePoint Server 2010 on a single server that uses the built-in database.

Copy the following code to a text editor and save it as simplesingleserver.ps1 in the directory of your choice:

```
$SetupPath         = Read-Host -Prompt "Please specify the path to the install media (D:)"


## Here is the script to install SharePoint Server 2010 with SQL Express and create Central
Admin ##
& $SetupPath\PrerequisiteInstaller.exe /unattended | Write-Host
if( $lastexitcode -eq 0 ) {
    & $SetupPath\setup.exe /config $SetupPath\Files\SetupSilent\config.xml | Write-Host
    if( $lastexitcode -eq 0 ) {
        Write-Host "Install successful..."
    } else { Write-Error "ERROR: $lastexitcode" }
} else { Write-Error "ERROR: $lastexitcode" }
```

To run simplesingleserver.ps1

1. Navigate to the Files\SetupSilent directory on the installation source and open config.xml in a text editor.
2. Locate the `<!--<PIDKEY Value="Enter Product Key Here" />-->` line.
3. Replace "Enter Product Key Here" with one of the following PIDKEYs: PKXTJ-DCM9D-6MM3V-G86P8-MJ8CY or BV7VC-RMR6B-26P6Y-BTQDG-DX2KQ
4. Remove the comment markers for the line of code you edited and then save the config.xml file.
5. Start Windows PowerShell 2.0 using the Run as administrator option.
6. Navigate to the directory where you saved simplesingleserver.ps1.
7. Run .\simplesingleserver.ps1 from the Windows PowerShell command prompt.

simplefarm.ps1

This script deploys SharePoint Server 2010 on a single server that uses a SQL Server database. Copy the following code to a text editor and save it as simplefarm.ps1 in the directory of your choice:

```
## Settings you may want to change ##
$err = $null
$SetupPath         = Read-Host -Prompt "Please specify the path to the install media (D:)"
Write-Host "Please specify the Farm Administrator credentials"
```

```
$FarmCredential    = Get-Credential "DOMAIN\<Farm Admin>"
$DBServer          = Read-Host -Prompt "Please enter the name of your database server"
$Passphrase        = Read-Host -Prompt "Please enter the farm passphrase (optional)" -
AsSecureString
$FarmName          = Read-Host -Prompt "Please enter a farm name (optional)"
$CAPort            = Read-Host -Prompt "Please enter the Central Administration port number
(optional)"

if ([String]::IsNullOrEmpty($SetupPath))
{
    Write-Error "You must enter the install media path"
    return
}
if ([String]::IsNullOrEmpty($FarmCredential))
{
    Write-Error "You must enter a Farm Administrator's user name and password"
    return
}
if ([String]::IsNullOrEmpty($DBServer))
{
    Write-Error "You must enter a database server"
    return
}
if ($Passphrase.Length -eq 0)
{
    Write-Warning "You didn't enter a farm passphrase, using the Farm Administrator's password
instead"
    $Passphrase = $FarmCredential.Password
}
if ([String]::IsNullOrEmpty($FarmName))
{
    Write-Warning "You didn't enter a farm name, using the machine name instead"
    $FarmName = $env:COMPUTERNAME
```

```
}
if ([String]::IsNullOrEmpty($CAPort))
{
    Write-Warning "You didn't enter a Central Administration port number, using 5000 instead"
    $CAPort = 5000
}

## Here is the script to install SharePoint Server 2010 and create Central Admin ##
Write-Host "[1/15] Running prerequisite installer..."
& $SetupPath\PrerequisiteInstaller.exe /unattended | Write-Host
if( $lastexitcode -eq 0 ) {
    Write-Host "[2/15] Running silent farm binary installation... (this will take some time)"
    & $SetupPath\setup.exe /config $SetupPath\Files\SetupFarmSilent\config.xml | Write-Host
    Write-Host "[3/15] Completed silent farm binary installation."
    if( $lastexitcode -eq 0 ) {
        Add-PSSnapin Microsoft.SharePoint.PowerShell -erroraction SilentlyContinue
        Write-Host "[4/15] Creating new configuration database..."
        New-SPConfigurationDatabase -DatabaseName ("{0}_SharePoint_Configuration_DB" -f
$FarmName) -DatabaseServer $DBServer -AdministrationContentDatabaseName ("{0}_AdminContent_DB"
-f $FarmName) -FarmCredentials $FarmCredential -Passphrase $Passphrase -ErrorVariable err
        Write-Host "[5/15] Verifying farm creation..."
        $spfarm = get-spfarm
            if ($spfarm -ne $null) {
            Write-Host "[6/15] ACLing SharePoint Resources..."
            Initialize-SPResourceSecurity -ErrorVariable err
            if ([String]::IsNullOrEmpty($err) -eq $true) {
                Write-Host "[7/15] Installing Services..."
                Install-SPService -ErrorVariable err
                    if ([String]::IsNullOrEmpty($err) -eq $true) {
                    Write-Host "[8/15] Installing Features..."
                    Install-SPFeature -AllExistingFeatures -ErrorVariable err
                    if ([String]::IsNullOrEmpty($err) -eq $true) {
                        Write-Host "[9/15] Provisioning Central Administration..."
```

```
                        New-SPCentralAdministration -Port $CAPort -WindowsAuthProvider "NTLM" -
ErrorVariable err
                        if ([String]::IsNullOrEmpty($err) -eq $true) {
                            Write-Host "[10/15] Installing Help..."
                            Install-SPHelpCollection -All -ErrorVariable err
                            if ([String]::IsNullOrEmpty($err) -eq $true) {
                                Write-Host "[11/15] Installing Application Content..."
                                Install-SPApplicationContent -ErrorVariable err
                                if ([String]::IsNullOrEmpty($err) -eq $true) {
                                    Write-Host "[12/15] Basic configuration completed."
                                    $SPManagedAccount = Get-SPManagedAccount
$FarmCredential.UserName
                                    Write-Host "[13/15] Creating Web Application and
Application Pool..."
                                    $webApp = New-SPWebApplication -Name "SharePoint - 80" -
ApplicationPool "SharePoint - 80" -ApplicationPoolAccount $SPManagedAccount -DatabaseServer
$DBServer -DatabaseName ("{0}_Content_DB" -f $FarmName) -Url "http://localhost" -ErrorVariable
err
                                    if ([String]::IsNullOrEmpty($err) -eq $true) {
                                        Write-Host "[14/15] Creating site collection..."
                                        $SPSite = New-SPSite -url "http://localhost" -
OwnerAlias $FarmCredential.UserName -ErrorVariable err
                                        if ([String]::IsNullOrEmpty($err) -eq $true) {
                                            Write-Host "[15/15] Port 80 Web application and
site collect created.  Browse to http://localhost to pick a template for the Web..."
                                        } else { Write-Error "ERROR: $err" }
                                    } else { Write-Error "ERROR: $err" }
                                } else { Write-Error "ERROR: $err" }
                            } else { Write-Error "ERROR: $err" }
                        } else { Write-Error "ERROR: $err" }
                    } else { Write-Error "ERROR: $err" }
                } else { Write-Error "ERROR: $err" }
            } else { Write-Error "ERROR: $err" }
        } else { Write-Error "ERROR: $err" }
```

```
    } else { Write-Error "ERROR: $lastexitcode" }
} else { Write-Error "ERROR: $lastexitcode" }


$exitprompt = Read-Host -Prompt "Press Enter to exit..."
```

To run simplefarm.ps1

1. Navigate to the Files\SetupFarmSilent directory on the installation source and open config.xml in a text editor.
2. Locate the `<!--<PIDKEY Value="Enter Product Key Here" />-->` line.
3. Replace "Enter Product Key Here" with one of the following PIDKEYs: PKXTJ-DCM9D-6MM3V-G86P8-MJ8CY or BV7VC-RMR6B-26P6Y-BTQDG-DX2KQ
4. Remove the comment markers for the line of code you edited and then save the config.xml file.
5. Start Windows PowerShell 2.0 using the Run as administrator option.
6. Navigate to the directory where you saved simplefarm.ps1.
7. Run .\simplefarm.ps1 from the Windows PowerShell command prompt.

Deploy by using DBA-created databases (SharePoint Server 2010)

This article describes how to deploy Microsoft SharePoint Server 2010 in a farm environment that uses DBA-created databases. In organizations where database administrators (DBAs) operate independently from SharePoint administrators, the DBAs create and manage all the databases. This is typical in IT environments where security requirements and company policies require a separation of administrator roles. The farm administrator provides Microsoft SharePoint Server 2010 database requirements to the database administrator, who in turn, creates the necessary Microsoft SharePoint Server databases and sets up the logins that are required for the farm.

In this article:

- Before you begin
- About configuring DBA-created databases
- Create and configure databases for Central Administration
- Create and configure additional databases

Before you begin

Before you start this deployment, ensure that you have all the information that you require in order to successfully deploy and configure SharePoint Server on all of the farm servers. The following sections provide the information that you will need to ensure a successful SharePoint Server deployment.

Farm server requirements

Ensure that all the farm servers and the database server meet the requirements that are documented in the following articles.

- Hardware and software requirements: Hardware and software requirements (SharePoint Server 2010)
- Administrative and service accounts: Administrative and service accounts required for initial deployment (SharePoint Server 2010)

Database requirements

Deploying SharePoint Server 2010 on DBA-created databases involves working with the DBA to ensure that all the SharePoint Server databases that you need are created and correctly configured before you create and configure the farm.

The following list shows some, but not necessarily all, of the information that a DBA needs in order to create databases for the farm. Additional information may be required by the DBA in your organization:

- SQL Server version information as well as service pack and cumulative update level. For more information, see Hardware and software requirements (SharePoint Server 2010).
- The required login accounts with associated roles and permissions. For more information, see Administrative and service accounts required for initial deployment (SharePoint Server 2010).
- The number of databases that are required as well as SharePoint configuration specifics. This information can be obtained by deploying SharePoint Server.
- SharePoint data storage requirements, such as data type, data volume, type of database activity (read or write) and Input/Output operations per second (IOPS).
- The DBA must configure surface area settings so that local and remote connections use TCP/IP or named pipes.
- All of the databases required by SharePoint Server use the Latin1_General_CI_AS_KS_WS collation.
- All of the SharePoint Server databases require that the farm Setup user account is assigned to them as the database owner (**dbo**).
- SharePoint user Service Level Agreement considerations.

About configuring DBA-created databases

Use the procedures in this article as a guide for deploying a farm that uses DBA-created databases. This deployment includes all the databases that are required for the farm.

Note:
This article only applies to the SQL Server database versions supported by SharePoint Server 2010.

For each procedure you must use Windows PowerShell 2.0 or SharePoint Server command-line tools to configure the farm.

We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

Note:
Psconfig is located in the following folder: Program Files\Common Files\Microsoft Shared\web server extensions\14\BIN.

In order to use Windows PowerShell to configure the farm:

1. Verify that the user account has access to one of the servers on which Windows PowerShell 2.0 is running, and that the user account is a Farm Administrator and is a member of the **SharePoint_Shell_Access** role for the SQL Server-based source content database, the administration content database, the destination content database, and the configuration database.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.

4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell prompt, type the appropriate command, and then press ENTER.

For the purpose of illustrating the required procedures, the basic farm that needs to be configured consists of:

1. Central Administration
2. A Web portal
3. Diagnostic logging and usage and health data collection
4. Enterprise search

The following databases are required and are typically used by the farm administrator in the following sequence as the farm is created. The databases in the following list use the default names that are provided when you use the SharePoint Products Configuration Wizard to set up a farm. You can, of course, use database names that you choose.

- The configuration database (SharePoint_Config)
- The Central Administration content database (SharePoint_AdminContent_GUID)
- The Web site content database, which is created automatically by the SharePoint Server Setup program (WSS_Content_GUID)
- The diagnostic logging database (WSS_Logging_GUID)
-
- Databases for the SharePoint Server search service application that is required for enterprise search. For more information, see Service application and service management (SharePoint Foundation 2010) (*http://technet.microsoft.com/library/aaed3a01-f4dc-4353-abda-0beced2080b6(Office.14).aspx*), Service application and service management (SharePoint Server 2010).
  - The search crawl database (Search_Service_1_CrawlStoreDB_GUID
  - The search administration database (Search_Service_1_DB_GUID)
  - The search property database (Search_Service_1_PropertyStoreDB_GUID

Create and configure databases for Central Administration

Use the procedures in this section to create the required databases and give the accounts membership in the database Users security group and database roles.

The procedures require action by the DBA and the Setup user account. The labels [DBA] or [Setup] respectively are used for each step to indicate which role performs the action.

The following procedure only has to be performed once for the farm, on the server that you want to run the Central Administration Web site. The farm has one configuration database and one content database for Central Administration.

To create and configure the configuration database, the Central Administration content database, and the Central Administration Web application

1. [DBA] Create the configuration database and the Central Administration content database using the LATIN1_General_CI_AS_KS_WS collation sequence and set the database owner (**dbo**) to be the Setup user account.
2. [Setup] Run Setup on each server computer in the farm. You must run Setup on at least one of these computers by using the **Complete** installation option. The steps for this option are described in Deploy a single server with SQL Server (SharePoint Server 2010).
3. [Setup] Do not run the SharePoint Products Configuration Wizard after Setup finishes.

   From the SharePoint 2010 Management Shell, use the **New-SPConfigurationDatabase** command to create a new configuration database, for example:

   ```
   New-SPConfigurationDatabase -DatabaseName "SharePointConfigDB1" -DatabaseServer
   "SQL-01" -Passphrase (ConvertTo-SecureString "MyPassword" -AsPlainText -force) -
   FarmCredentials (Get-Credential)
   ```

   For more information, see New-SPConfigurationDatabase (*http://technet.microsoft.com/library/b04f1577-1985-41b8-b555-2f5145a00241(Office.14).aspx*).
4. [Setup] After the command has finished, run the SharePoint Products Configuration Wizard and complete the rest of the configuration for the server. This creates the Central Administration Web application and performs other setup and configuration tasks.
5. [DBA] After the SharePoint Products Configuration Wizard has finished, perform the following actions for both the configuration database and the Central Administration content database:
   - Add the SharePoint Server search account, default content access account, and the services account to the Users group.
   - Add the SharePoint Server search service account, default content access account, and the services account to the WSS_Content_Application_Pools role.
6. [Setup] To confirm that the databases were created and configured correctly, verify that the home page of the Central Administration Web site can be accessed. However, do not configure anything by using Central Administration at this point. If the Central Administration page does not render, verify the accounts that are used in this procedure and ensure that they are properly assigned.

The rest of the farm servers will be configured after the procedures in the article are finished and the farm is established. You will run the SharePoint Products Configuration Wizard on these servers by selecting the **Yes, I want to connect to an existing server farm** option, instead of by using the commands that are used in this procedure.

The following procedure will only have to be performed once for the farm. The farm has only one SharePoint Foundation search database.

Create and configure the SharePoint Foundation search database and start the search service

1. [DBA] Create the SharePoint Foundation search database using the LATIN1_General_CI_AS_KS_WS collation sequence and set the database owner (**dbo**) to be the Setup user account.
2. [Setup] Open the command line, and then use the **Get-SPSearchService**, **Set-SPSearchService**, **Get-SPSearchServiceInstance**, and **Set-SPSearchServiceInstance** cmdlets to configure the database and start the search service. Use the following example as a guide.

```
$searchService = Get-SPSearchService

Set-SPSearchService -MaxBackupDuration 120

Get-SPSearchServiceInstance -Local

Get-SPSearchServiceInstance -Local | Set-SPSearchServiceInstance -ProxyType proxy
```

For more information, see the following topics:

- Get-SPSearchService (*http://technet.microsoft.com/library/90160cc4-60c3-4983-8b4a-674cbf4c4f9c(Office.14).aspx*)
- Set-SPSearchService (*http://technet.microsoft.com/library/664d55c9-c436-4096-a385-446c920f4df1(Office.14).aspx*)
- Get-SPSearchServiceInstance (*http://technet.microsoft.com/library/d0fcee38-4403-4ef6-b3ed-c28cec050557(Office.14).aspx*)
- Set-SPSearchServiceInstance (*http://technet.microsoft.com/library/85dce2d2-1b01-4f7f-86d0-5523c432efe6(Office.14).aspx*)

Create and configure additional databases

After you finish configuring the databases required for Central administration, complete your farm deployment by creating and configuring the databases that are required for Web content and any service applications that you want to use.

The following procedure will have to be performed once for each portal site in the farm.

The procedures described here are used for the following SharePoint Server 2010 operations:

- Start the search service on servers that are hosting search query components or search crawl components. For more information, see Manage search topology (*http://technet.microsoft.com/library/ff30cd9b-a827-4bee-b38a-ed2c3b6b3b47(Office.14).aspx*).
- Create and configure the portal site Web application content database.

- Create and configure the search service application databases.

The following procedure must be performed once for each server that runs search queries or database crawls in the farm.

To start the SharePoint Server Search service on each server that will run search queries or crawls

1. [Setup] From the SharePoint 2010 Management Shell, use the **Get-SPEnterpriseSearchService, Set- SPEnterpriseSearchService**, and **Get-SPEnterpriseSearchServiceInstance** cmdlets. Use the following example as a guide.

```
$searchService = Get-SPEnterpriseSearchService

$password = Read-Host -AsSecureString

**********

Set-SPEnterpriseSearchService -IgnoreSSLWarnings $true -ServiceAccount
contoso\adminAccount -ServicePassword $password

$ssInstance = Get-SPEnterpriseSearchServiceInstance -Local

Get-SPEnterpriseSearchServiceInstance -Local | Start-
SPEnterpriseSearchServiceInstance
```

   For more information, see the following topics:

   - Get-SPEnterpriseSearchService (*http://technet.microsoft.com/library/fcf865a8-b46b-44fb-89ba-f52e20876f6c(Office.14).aspx)*
   - Set-SPEnterpriseSearchService (*http://technet.microsoft.com/library/f8baccd0-21d2-40aa-b700-997ec7ca7011(Office.14).aspx*)
   - Get-SPEnterpriseSearchServiceInstance (*http://technet.microsoft.com/library/309d98e8-a5fa-4cb5-b6e1-bf94380a8212(Office.14).aspx*)

The following procedure must be performed once for each service application in the farm. In the scenario used for this article, a search service application is used for the farm. The search service application requires the following databases:

1. Search administration database (Search_Service_DB)
2. Search crawl database (Search_Service_CrawlStoreDB)
3. Search property database (Search_Service_PropertyStoreDB)

All the command-line instructions in the following procedure use Windows PowerShell.

To create and configure databases for search

1. [DBA] Add the SharePoint 2010 Management Shell administration account to the **dbcreator** and **securityadmin** roles as temporary SQL Server logins.

2. [Setup] Create the search service application on the temporary server, as follows:
   a. Create the necessary application pools for the search administration and search query Web services, as follows:

```
# Create managed account for DOMAIN\QryPool
$SearchCR = Get-Credential;$managedAccount = New-SPManagedAccount -Credential
$SearchCR
# Create managed account for DOMAIN\AdminPool
$SearchCR = Get-Credential;$managedAccount = New-SPManagedAccount -Credential
$SearchCR
```

   b. Create a new search service application using a temporary Microsoft SQL Server database, as follows:

```
Write-Host -ForegroundColor white Creating the Search service application
$searchAppName="SearchAppTest"
$appQueryPoolName=$searchAppName+"_AppQueryPool"
$appAdminPoolName=$searchAppName+"_AppAdminPool"
$appQueryPool = New-SPServiceApplicationPool -name $appQueryPoolName -account
DOMAIN\QryPool
$appAdminPool = New-SPServiceApplicationPool -name $appAdminPoolName -account
DOMAIN\AdminPool

$searchApp = New-SPEnterpriseSearchServiceApplication -Name $searchAppName -
ApplicationPool $appQueryPool -AdminApplicationPool $appAdminPool -DatabaseServer
<Temp_SQL_Server>

Write-Host -ForegroundColor white Initializing the administration component
$searchInstance = Get-SPEnterpriseSearchServiceInstance -Local
$searchApp | get-SPEnterpriseSearchAdministrationComponent | set-
SPEnterpriseSearchAdministrationComponent -SearchServiceInstance $searchInstance
$admin = ($searchApp | get-SPEnterpriseSearchAdministrationComponent)
Write-Host -ForegroundColor white Waiting for the admin component to be initialized
do {write-host -NoNewline .;Start-Sleep 10;} while (-not $admin.Initialized)

Write-Host -ForegroundColor white Creating a new crawl topology
$InitialCrawlTopology = $searchApp | Get-SPEnterpriseSearchCrawlTopology -Active
```

```
$InitialCrawlTopology

$searchInstance0 = Get-SPEnterpriseSearchServiceInstance -Local

$CrawlTopology = $searchApp | New-SPEnterpriseSearchCrawlTopology

$CrawlDatabase0 = ([array]($searchApp | Get-SPEnterpriseSearchCrawlDatabase))[0]

$CrawlComponent0 = New-SPEnterpriseSearchCrawlComponent -CrawlTopology
$CrawlTopology -CrawlDatabase $CrawlDatabase0 -SearchServiceInstance
$searchInstance0 -IndexLocation $searchInstance0.DefaultIndexLocation

$CrawlTopology | Set-SPEnterpriseSearchCrawlTopology -Active

Write-Host -ForegroundColor white Waiting for the old crawl topology to become
inactive

do {write-host -NoNewline .;Start-Sleep 10;} while ($InitialCrawlTopology.State -ne
"Inactive")

$InitialCrawlTopology | Remove-SPEnterpriseSearchCrawlTopology


Write-Host -ForegroundColor white Creating a new query topology

$InitialQueryTopology = $searchApp | Get-SPEnterpriseSearchQueryTopology -Active

$InitialQueryTopology

$searchInstance0 = Get-SPEnterpriseSearchServiceInstance -Local

# Create a new query topology

$QueryTopology = $searchApp | New-SPEnterpriseSearchQueryTopology -Partitions 1

$IndexPartition0= (Get-SPEnterpriseSearchIndexPartition -QueryTopology
$QueryTopology)

$QueryComponent0 = New-SPEnterpriseSearchQuerycomponent -QueryTopology
$QueryTopology -IndexPartition $IndexPartition0 -SearchServiceInstance
$searchInstance0 -IndexLocation $searchInstance0.DefaultIndexLocation

# Get the default property store

$PropertyDatabase0 = ([array]($searchApp | Get-
SPEnterpriseSearchPropertyDatabase))[0]

$IndexPartition0 | Set-SPEnterpriseSearchIndexPartition -PropertyDatabase
$PropertyDatabase0

$QueryTopology | Set-SPEnterpriseSearchQueryTopology -Active

Write-Host -ForegroundColor white Waiting for the old query topology to become
inactive

do {write-host -NoNewline .;Start-Sleep 10;} while ($InitialQueryTopology.State -ne
```

```
"Inactive")

# Delete the old query topology

$InitialQueryTopology | Remove-SPEnterpriseSearchQueryTopology

Write-Host -ForegroundColor white Creating the Search application proxy

$searchAppProxy = new-spenterprisesearchserviceapplicationproxy -name
($searchApp.Name+"_proxy") -Uri $searchApp.Uri

Write-Host -ForegroundColor white Search provisioning finished
```

c. Pause the search service application

```
$searchApp.Pause()
```

3. [DBA] Back up all the search databases: SearchAppTest, SearchAppTest_CrawlStore, and SearchAppTest_PropertyStore.
4. [DBA] Restore all the search databases to the production SQL Server database server.
5. [DBA] Add the following accounts to the SQL Server**Logins** on the production SQL Server database server using the default settings (leave the Public role checked for Server roles):
   - Search service account
   - Search Query and Site Settings Web Service application pool account
   - Search Admin Web Service application pool account (if different from the search service application pool)
6. [DBA] If the SQL Server database has not been used to provision farm databases, add the following accounts to the SQL Server**Logins**
   - The SharePoint_Shell_Access role for the SQL Server-based source content database, the administration content database, the destination content database, and the configuration database
   - The farm administrator account
7. [Setup] Switch over to the new production database environment.
   a. Switch the search administration database to the new search administration production database.

```
$searchapp | Set-SPEnterpriseSearchServiceApplication -DatabaseServer
<Production_SQL_server>
```

   b. Wait for the search service instances to finish re-provisioning.

```
do {write-host -NoNewline .;Sleep 10; $searchInstance = Get-
SPEnterpriseSearchServiceInstance –Local} while ($searchInstance.Status -ne
"Online")
```

   c. Switch the search crawl database to the new search crawl production database.

```
$CrawlDatabase0 | Set-SPEnterpriseSearchCrawlDatabase -DatabaseServer
<Production_SQL_server>
```

d. Wait for the search service instances to finish re-provisioning.

```
do {write-host -NoNewline .;Sleep 10; $searchInstance = Get-
SPEnterpriseSearchServiceInstance –Local} while ($searchInstance.Status -ne
"Online")
```

e. Switch the search property database to the new search property production database.

```
$PropertyDatabase0 | Set-SPEnterpriseSearchPropertyDatabase -DatabaseServer
<Production_SQL_server>
```

f. Wait for the search service instances to finish re-provisioning.

```
do {write-host -NoNewline .;Sleep 10; $searchInstance = Get-
SPEnterpriseSearchServiceInstance –Local} while ($searchInstance.Status -ne
"Online")
```

g. Restart the search service instances.

```
get-SPEnterpriseSearchServiceInstance | start-SPEnterpriseSearchServiceInstance
```

h. Resume the search service application.

```
$searchapp.Resume()
```

Use the following procedure to create and configure the portal site Web application and its content database.

To create and configure the portal site Web application

1. [DBA] Create the portal site Web application content database using the LATIN1_General_CI_AS_KS_WS collation sequence and set the database owner (**dbo**) to be the Setup user account.
2. [DBA] Using SQL ServerManagement Studio, add the service application service account to the Users group and then to the **db_owner** role for the portal site Web application content database.
3. [Setup] From the SharePoint 2010 Management Shell, use the **New-SPWebApplication** and **Get-SPWebApplication** cmdlets to configure the portal site Web application content database. Use the following example as a guide.

```
New-SPWebApplication -Name "Contoso Internet Site" -Port 80 -HostHeader
"http://sharepoint.contoso.com" -URL "https://www.contoso.com" -ApplicationPool
"ContosoAppPool" -ApplicationPoolAccount (Get-SPManagedAccount "DOMAIN\jdoe")

Get-SPWebApplication http://somesite | Set-SPWebApplication -Zone "Extranet" -
HostHeader "http://www.contoso.com" - AllowAnonymousAccess
```

For more information, see New-SPWebApplication (*http://technet.microsoft.com/library/eaeb5bed-81e7-4275-b005-aa7fc465e6d5(Office.14).aspx*) and Get-SPWebApplication (*http://technet.microsoft.com/library/11d6521f-f99c-433e-9ab5-7cf9e953457a(Office.14).aspx*).

Important:

This command must be run on the same computer that is running the Web application. The host name and port combination must not describe a Web application that already exists or an error results and the Web application is not created.

4. [Setup] From a command prompt, run the following command to restart IIS: **iisreset /noforce**

Deploy in a virtual environment (SharePoint Server 2010)

This section contains articles that provide guidance for configuring virtual machines for Microsoft SharePoint Server 2010 servers in a virtual environment.

In this section:

- Virtual machine guidance (SharePoint Server 2010)

Virtual machine guidance (SharePoint Server 2010)

This article provides guidance for configuring a virtual machine (VM) that uses Windows Server 2008 Hyper-V technology and that is used in a Microsoft SharePoint Server 2010 farm. This includes farm servers on the Web server tier, application server tier, and database server tier.

Before you configure a virtual machine for a SharePoint farm, we recommend that you read the Hyper-V Getting Started Guide (*http://go.microsoft.com/fwlink/?LinkId=187754*). Because every configuration decision you make for a virtual machine or its infrastructure has an impact on performance or functionality (SharePoint Server 2010 and Hyper-V)—understanding each configuration option is important.

In a Hyper-V environment, you have the option of specifying the configuration of virtual networking and the configuration for each virtual machine. Additionally, you can configure how the VM interacts with the virtualization host, as well as VM stop and restart behavior if the running state of the virtual machine is interrupted.

In this article:

- Networking
- Network adapters
- Virtual machine configuration
- Integration services
- Using snapshots

Networking

You can configure Hyper-V networking before you create any virtual machines or after you create one or more VMs. You can also create more than one virtual network for a Hyper-V host.

Using Virtual Network Manager (accessed from Hyper-V Manager), you have three different types of virtual networks to choose from. The following table provides a summary of the network types and their characteristics.

Type	Description
External	Use this type when you want to allow VMs to communicate with externally located servers and the management operating system (sometimes referred to as the parent partition). This type also allows VMs on the same physical server to communicate with each other.

Type	Description
Internal	Use this type when you want to allow communication between VMs on the same physical server and VMs and the management operating system. An internal virtual network is a virtual network that is not bound to a physical network adapter. It is commonly used to build a test environment where you need to connect to the VMs from the management operating system.
Private	Use this type when you want to allow communication only between VMs on the same physical server. A private virtual network is a virtual network without a virtual network adapter in the management operating system. Private virtual networks are commonly used when you want to isolate VMs from network traffic in the management operating system and in the external networks.

Use SharePoint Server farm requirements to determine which of the three networking configurations that you want to use on a virtualization host. For example, in the Quick start: Deploy single server in an isolated Hyper-V environment (SharePoint Server 2010) article, we used an Internal network to isolate the test environment from a production environment.

After determining the type of network that you want to use, you can specify the range of media access control (MAC) addresses that are automatically assigned to virtual adapters. The R2 release of Microsoft Hyper-V Server 2008 enables you to provide static MAC addresses to an adapter in order to avoid collisions on a network.

From a networking performance perspective, the ability to create virtual local area networks (VLANs) can provide performance gains. Virtual machines on the same VLAN can communicate through the virtual switch, which means that network traffic is faster because it does not have to use the physical network adapter. Another benefit of a VLAN configuration is the fact that is software-based, computers can easily be moved and still maintain their network configurations.

The following links provide more information about virtual networking concepts and how to configure virtual networks.

- How does basic networking work in Hyper-V? (*http://go.microsoft.com/fwlink/?LinkId=128228*)
- Hyper-V: What are the uses for different types of virtual networks? (*http://go.microsoft.com/fwlink/?LinkId=128085*)
- Understanding Hyper-V VLANs (*http://go.microsoft.com/fwlink/?LinkId=180709*)
- Hyper-V VLANs Part II (*http://go.microsoft.com/fwlink/?LinkId=18775*)
- Configuring Virtual Networks (*http://go.microsoft.com/fwlink/?LinkId=158767*)

Network adapters

Two types of network adapters can be attached to a virtual machine: a network adapter and a legacy adapter. A network adapter provides better performance than the legacy network adapter. The legacy adapter emulates an Intel 21140-based PCI Fast Ethernet Adapter, which results in a lower data

transfer than the network adapter. A legacy network adapter also supports network-based installations because it includes the ability to boot to the Pre-Boot Execution Environment (PXE).

Unless you need to use a legacy adapter until you can install the virtual machine driver, or need to do a network boot, we recommend that you configure the VM with a network adapter.

Note:
You can use the legacy adapter initially to do a network boot, and when that is no longer required, add a network adapter and delete the legacy adapter.

As is the case with physical servers, you can install multiple adapters on a virtual machine.

Virtual machine configuration

Hyper-V provides numerous options for configuring a virtual machine. These options can be changed after you have started the virtual machine and installed Microsoft SharePoint Server. You will have to shut the virtual machine down in order to the change the configuration. Based on the capacity management and high availability planning that you did for Capacity management and high availability in a virtual environment (SharePoint Server 2010) (*http://technet.microsoft.com/library/5c23b49c-c4da-4bd3-9554-285700ff80e5(Office.14).aspx*), configure the following for each virtual machine:

- The boot sequence (legacy network adapter, CD, IDE, or floppy disk)
- The amount of memory
- The number of virtual processors, up to a limit of four
- The type and number of controllers
- The type and number of disks
- The type and number of network adapters

In addition to the preceding configurations, you also have the option to configure COM ports and to configure a virtual floppy disk.

From a SharePoint Server perspective, the primary configuration considerations are memory, processor, and the type and number of controllers and hard disks.

Memory

After factoring in a 2 GB RAM reserve for the virtualization host, you can configure a virtual machine to have any amount of the remaining memory. You will, of course have to take into account the amount of memory that you provide to other virtual machines running on the same virtualization host.

Note:
The 2 GB of RAM reserved for the virtualization host is used as a guide and not a required or enforced amount of memory. Typically the actual amount of RAM committed to the physical server is less.

The actual memory overhead on Hyper-V is fairly small. This is well-illustrated if you download the Hyper-V RAM Calculator.xls (*http://go.microsoft.com/fwlink/?LinkId=187756*) and use it to calculate RAM use on a host of a given size with a several virtual machines.

Processor

You can configure multiple virtual processors for a virtual machine, up to a limit of four processors. You cannot configure more processors per VM that there are logical (cores) processors on the virtualization host. For example, given a dual core physical server, you are limited to configuring two virtual processors for a VM. Although Hyper-V supports up to eight virtual processors per core, a configuration that uses this ratio (1 logical: 8 virtual) is referred to as being oversubscribed. For any virtual machine used in a SharePoint farm, we recommend a ratio of 1:1. Oversubscribing the CPU on the virtualization host will work, but performance will degrade depending on the amount of oversubscription.

Controller and hard disk

You can select either integrated device electronics (IDE) or SCSI devices on virtual machines, as follows:

- IDE devices: Hyper-V uses emulated devices with IDE controllers. You can have up to two IDE controllers with two disks on each controller. The startup disk (sometimes referred to as the boot disk) must be attached to one of the IDE devices. The startup disk can be either a virtual hard disk or a physical disk. Although a virtual machine must use an IDE device as the startup disk to start the guest operating system, you have many options to choose from when selecting the physical device that will provide the storage for the IDE device.
- SCSI devices: Each virtual machine supports up to 256 SCSI disks (four SCSI controllers with each controller supporting up to 64 disks). SCSI controllers use a type of device developed specifically for use with virtual machines and use the virtual machine bus to communicate. The virtual machine bus must be available when the guest operating system is started. Therefore, virtual hard disks attached to SCSI controllers cannot be used as startup disks.

  Note:
  Although the I/O performance of physical SCSI and IDE devices can differ significantly, this is not true for the virtualized SCSI and IDE devices in Hyper-V. IDE and SCSI devices both offer equally fast I/O performance when integration services are installed in the guest operating system. Support for hot swappable hard drives, which is supported by the Hyper-V implementation of SCSI, is a better reason for selecting SCSI drives than performance gains.

The version of Hyper-V released with Windows Server 2008 R2 provides dramatic improvements in virtual hard disk performance. For more information, see Virtual Hard Disk Performance: Windows Server 2008 / Windows Server 2008 R2 / Windows 7 (*http://go.microsoft.com/fwlink/?LinkId=186519*). For a summary of virtual machine drive options, see the "How to choose your Hyper-V and VHD Storage Container Format" section of this white paper. Also, heed the authors' advice: When choosing

the right VHD for your environment, consider both the access performance and storage needs. When using Windows Server 2008 R2, the choice has less to do with the access speed and more to do with the amount of memory used due to advanced caching.

Important:
There is not a generic storage solution for every virtual environment. Selecting the optimal virtual machine drive option for your SharePoint Server servers requires research and extensive testing to implement the best storage solution for your virtual environment.

Integration services

Hyper-V includes a software package for supported guest operating systems that improves integration between the physical computer and the virtual machine. This package is referred to as integration services. You should verify that the management operating system (which runs the Hyper-V role) and virtual machines are running the same version of integration services. For more information, see Version Compatibility for Integration Services (*http://go.microsoft.com/fwlink/?LinkId=188011*).

For each virtual machine you can configure the following integration items between the VM and the virtualization host:

- Operating system shutdown
- Time synchronization
- Data exchange
- Heartbeat
- Backup (volume snapshot)

Important:
Disable the time synchronization for each SharePoint Server virtual machine. SharePoint Server 2010 implements timer jobs extensively and the latency during time synchronization will cause unpredictable results in the SharePoint Server environment.

Automatic stop and start

For each virtual machine you can configure automatic stop and start behavior if a physical machine shuts down. The options for stop are:

- Save the virtual machine state. The current state of the virtual machine is saved and when the VM is started, Hyper-V attempts to restore the VM to the state it was in.
- Turn off the virtual machine. This is the equivalent of pulling the power plug on a server.
- Shut down the guest (virtual machine) operating system. This is the equivalent of shutting down the machine using the Windows Shut down option.

For a SharePoint Server virtual machine, do not configure the virtual machine to save state. Virtual machines that come up from saved state will be out of synchronization with the other servers in the farm. We recommend that you configure the virtual machine to use a shutdown because it provides the

cleanest method to minimize virtual machine corruption. When a shutdown occurs any timer jobs that are running are allowed to finish and there will not be any synchronization issues when the VM restarts.

The opposite of an automatic stop is an automatic start. Hyper-V provides the following startup options when the physical server restarts:

- Do nothing. You will have to start the VM manually regardless of its state when the physical server shut down.
- Automatically start if the machine was running when the service stopped.
- Always start this virtual machine automatically. Hyper-V will start the machine regardless of its state when the physical server shut down.

We recommend that you either of the first two options. Either option is valid, but the decision is ultimately up to the IT team that is managing and maintaining the virtual environment.

In addition to the preceding start options, you can configure a startup time delay for a virtual machine. We recommend that you do so in order to reduce resource contention on a virtualization host. However, if your start option is to do nothing, this is not an issue.

Using snapshots

Snapshots provide a very useful tool for capturing the current state of a running, paused, or stopped virtual machine. The snapshot feature enables you to quickly and easily revert to a previous virtual machine configuration. This capability is particularly well-suited to a development or test environment.

As a best practice, we recommend that you do not use the snapshot feature on virtual machines in a production environment for the following reasons:

- Clock synchronization: When you take a snapshot of a running virtual machine, there is latency between the time the snapshot is started and the time the snapshot is finished. This latency affects SharePoint Server timer jobs and, as a result, time synchronization between farm servers.

  Important:
  If you choose to take a snapshot of a virtual machine, shut down the machine to allow running jobs to finish before taking the snapshot. We recommend that you closely monitor the virtual machine and other farm servers after the virtual machine is restarted to ensure that there are no time synchronization issues.

- Performance: When you create a snapshot for a virtual machine you have, in effect, created a differencing disk. There is a continuous exchange of configuration data between the virtual machine and the snapshot, which affects performance.

Install SharePoint Server 2010 by using Windows PowerShell

This article discusses how to do a clean installation of Microsoft SharePoint Server 2010 on a stand-alone server or on a server farm by using Windows PowerShell.

Installation process overview:

- Farm server requirements
- Prepare SPModule
- Install Microsoft SharePoint Server 2010 by running Install-SharePoint
- Configure the farm by using New-SharePointFarm
- Create a Web Application using Windows PowerShell
- Deploy services using the SharePoint 2010 Farm Configuration Wizard
- Create a site collection by using Windows PowerShell
- Perform additional configuration tasks
- Add servers to the farm using Join-SharePointFarm
- Configure the trace log

You can streamline deployment by using Windows PowerShell to install Microsoft SharePoint Server 2010 in combination with other administrator tools to automate unattended installations and configure the farm.

Farm server requirements

Before you install SharePoint Server 2010, review the following information about permissions, hardware requirements, and software requirements and steps to perform before beginning the process:

- Ensure that you have met all hardware and software requirements. You must have a 64-bit version of Windows Server 2008 or Windows Server 2008 R2. For server farms, you must also have a 64-bit version of SQL Server 2005 or SQL Server 2008. For more information about these requirements (such as specific updates that you must install), see Determine hardware and software requirements (SharePoint Server 2010).
- Ensure that you are prepared to set up the required accounts by using appropriate permissions. For detailed information, see Administrative and service accounts required for initial deployment (SharePoint Server 2010).

Prepare SPModule

The SPModule.zip file is a Windows PowerShell module written by members of the SharePoint Product Group that will install a SharePoint farm.

To use the SPModule.zip file, use the steps below:

1. Download the SPModule.zip and text file from the following Download Center page (*http://go.microsoft.com/fwlink/?LinkId=187924*).
2. Extract the files to a folder named SPModule and then add that folder to your environment path. The `PSModulePath` environment variable is used to store paths to the locations of the modules that are installed on disk. To view paths specified in the `PSModulePath` variable, from a Windows PowerShell command prompt, type `$env:PSModulePath`. For more information, see PSModule Environment Variable (*http://go.microsoft.com/fwlink/?LinkId=187757*). After extraction, the SPModule folder should contain two files: SPModule.misc and SPModule.setup.

   Note:
   If the SPModule folder is not added to the `PSModulePath` variable, you must specify a full path to load the SPModule.
3. Decide on the type of signing to provide.

   Note:
   The downloaded files are unsigned. To provide self-signed scripts, see AllSigned: Signing Your PowerShell Scripts (*http://go.microsoft.com/fwlink/?LinkId=187758*).

   By default, Windows PowerShell execution policy is set to Restricted so that no scripts can be run. To change an execution policy, run the **Set-ExecutionPolicy** cmdlet. For additional information about execution polices, see About_Signing (*http://go.microsoft.com/fwlink/?LinkId=187759*).

1. Click **Start**, point to **All Programs**, and then click **Windows PowerShell V2**. Once the Windows PowerShell Command Prompt window opens, the first thing we need to do is add the path to the module to your Windows PowerShell module path (presuming you created a folder called "SPModule" on your server):
2. Right-click **Windows PowerShell V2**, and then click **Run as administrator**.

   Note:
   If you already have Microsoft SharePoint Server 2010 installed, you could open the SharePoint 2010 Management Shell instead.
3. Once the Command Prompt window is displayed, we need to import the module into the current Windows PowerShell session and add it to a path by typing the following syntax from a Windows PowerShell command prompt:

```
Import-Module SPModule.misc

Import-Module SPModule.setup
```

When you import the SPModule.misc module, you will invoke an update check. In version 1.0, this will check a text file to see if there is a newer version available. If you are notified that there is, you can go and download the newer version. Once the Import-Module commands have completed successfully, you are ready to use SPModule.

Important:
The following commands should only be used in the context of the SPModule and the process in this article.

1. Install-SharePoint: Installs all of the binary files for SharePoint Server 2010. For more information, see Scripted deployment reference (SharePoint Server 2010) (*http://technet.microsoft.com/library/7e1bffa2-bc8c-4522-b36f-7060722f9278(Office.14).aspx*).
2. New-SharePointFarm: Creates a SharePoint farm and performs the following related tasks:
   - Configures security
   - Creates a shared service Web application that you can populate with service applications.
   - Creates and configures the Central Administration Web site.
   - Installs all of the product Help files.
   - Installs all farm features.

   For more information, see Scripted deployment reference (SharePoint Server 2010) (*http://technet.microsoft.com/library/7e1bffa2-bc8c-4522-b36f-7060722f9278(Office.14).aspx*).
3. Join-SharePointFarm: Adds servers to the farm, and then configures them. For more information, see Scripted deployment reference (SharePoint Server 2010) (*http://technet.microsoft.com/library/7e1bffa2-bc8c-4522-b36f-7060722f9278(Office.14).aspx*).

After Windows PowerShell version 2.0 is installed, you can use a new feature of Windows PowerShell called Remoting. By using the remoting feature and a couple lines of Windows PowerShell code, an administrator can remotely install multiple servers in a farm. For information about Remoting and SPModule, see Remote Install with SPModule (*http://go.microsoft.com/fwlink/?LinkId=187923*).

Install Microsoft SharePoint Server 2010 by running Install-SharePoint

After you have determined the required accounts for the installation, you can installSharePoint Server 2010. The product DVD contains examples of configuration (Config.xml) files. These example files are stored under the \Files folder in the root directory of the DVD, in folders that correspond to different scenarios. These example files are described in the following table.

Configuration file	Description
Setup\Config.xml	Stand-alone server installation, using Microsoft SQL Server 2005 Express Edition
SetupFarm\Config.xml	Server farm installation

SetupFarmSilent\Config.xml	Server farm installation in silent mode
SetupFarmUpgrade\Config.xml	In-place upgrade of an existing farm
SetupSilent\Config.xml	Stand-alone server installation, usingSQL Server 2005 Express Edition, in silent mode
SetupSingleUpgrade\Config.xml	In-place upgrade of an existing single-server installation

To run Install-SharePoint with a Config.xml file at a Windows PowerShell command prompt

1. On the drive on which the SharePoint Server 2010 product DVD is located, change to the root directory to locate the setup.exe file.
2. Run SPModule.Setup Install-SharePoint with the selected Config.xml file, as follows:

   Install-SharePoint -SetupExePath<*path and file name*>**-ConfigXml**<*path and file name*>

   Note:
   You can select one of the example files, or customize your own configuration file.
3. Press ENTER.

Setup is now finished.

The following example shows the configuration file for setting up a single server in silent mode (SetupSilent).

```
<Configuration>
<Package Id="sts">
  <Setting Id="LAUNCHEDFROMSETUPSTS" Value="Yes" />
  </Package>
<Package Id="spswfe">
  <Setting Id="SETUPCALLED" Value="1" />
  </Package>
  <Logging Type="verbose" Path="%temp%" Template="SharePoint Server Setup(*).log" />
- <!--
<PIDKEY Value="Enter Product Key Here" />
  -->
  <Display Level="none" CompletionNotice="no" />
  <Setting Id="SERVERROLE" Value="APPLICATION" />
```

```
<Setting Id="USINGUIINSTALLMODE" Value="0" />
<Setting Id="SETUP_REBOOT" Value="Never" />
<Setting Id="SETUPTYPE" Value="CLEAN_INSTALL" />
</Configuration>
```

You can also customize your own configuration file. To control the installation, first edit the Config.xml file in a text editor to include the elements that you want with the appropriate settings for those elements. Next, run **Install-SharePoint -SetupExePath -ConfigXML** to specify that Setup runs and uses the options that you set in the Config.xml file.

Some typical configuration options include the following:

- Bypassing the prompt for the product key by providing the key as a value, `<PIDKEY Value="Enter PID Key Here" />`, in the Config.xml file.
- Adding a location for a log file, `<Logging Type="off" | "standard"(default) | "verbose" Path="path" Template="file name.log"/>`, which you can view if command-line installation fails.

Important:
Use a text editor, such as Notepad, to edit Config.xml. Do not use a general-purpose XML editor, such as Microsoft Office Word 2007. To validate that your XML file is well-formed, use any supported browser.

Run Install-SharePoint using a PID key

To run Setup in silent mode without using a configuration XML file, type one of the following commands at a Windows PowerShell command prompt:

- For stand-alone server: `Install-SharePoint -SetupExePath "<drive letter>:\SharePoint 2010\Setup\setup.exe" -PIDKey "PVRYW-RRP8H-FG6G2-YGMQW-3V9VX" -ServerRole "SINGLESERVER"`
- For a farm deployment: `Install-SharePoint -SetupExePath "<drive letter>:\SharePoint 2010\Setup\setup.exe" -PIDKey "PVRYW-RRP8H-FG6G2-YGMQW-3V9VX"`

To run Setup in silent mode using a configuration XML file, type one of the following commands at a Windows PowerShell command prompt:

- For a stand-alone server: `Install-SharePoint -SetupExePath "G:\SharePoint 2010\Setup\setup.exe" -ConfigXML "G:\SharePoint 2010\Setup\Config\singleserver_config.xml"-PIDKey "PVRYW-RRP8H-FG6G2-YGMQW-3V9VX" -ServerRole "SINGLESERVER"`
- For a farm deployment: `Install-SharePoint -SetupExePath "G:\SharePoint 2010\Setup\setup.exe" -ConfigXML "G:\SharePoint 2010\Setup\Config\appserver_config.xml"-PIDKey "PVRYW-RRP8H-FG6G2-YGMQW-3V9VX"`

Configure the farm by using New-SharePointFarm

You use the New-SharePointFarm command to configure SharePoint Server 2010 after the Install-SharePoint command has finished. The configuration options are different depending on whether you install SharePoint Server 2010 on a stand-alone server or on a server farm.

Configure SharePoint Server 2010 on a stand-alone server

In stand-alone server deployments, you can run New-SharePointFarm. After you have logged on by using the Setup user account that you previously created and configured, you configure SharePoint Server 2010.

To configure SharePoint Server 2010 on a stand-alone server by using New-SharePointFarm

- At the Windows PowerShell command prompt, type the following command:

  ```
  New-SharePointFarm -DatabaseServer <String> -DatabaseAccessAccount <(Get-
  Credential domain\account)> -FarmName <string>
  ```

  Where:

  - <*String*> is name database server. For example, "SQL01"
  - <*PSCredential*> is the DOMAIN\password of the user account that is performing the installation.
  - <*String*> is the name of the farm. For example, "Farm1"

  The New-SharePointFarm command describes the configuration steps as they occur, and notes the successful completion of configuration. For a stand-alone server installation, this is the final step in a command-line installation.

We strongly recommend that you install and configure SharePoint Server 2010 on all of the farm servers before you create sites.

Caution:
Do not run the New-SharePointFarm and Join-SharePointFarm commands simultaneously on multiple computers as contention issues and unpredictable behavior can occur.

Note:
If any of these commands fail, look in the post-setup configuration log files. The log files are available at %COMMONPROGRAMFILES%\Microsoft shared\Web server extensions\14\Logs, and can be identified by a file name that begins with PowerShell_ConfigurationDiagnostics and the .log file name extension.

Note:
The Windows PowerShell cmdlets mentioned in this section must be run from the SharePoint 2010 Management Shell.

To connect to an existing configuration database and join the server to an existing server farm, the **Connect–SPConfigurationDatabase** cmdlet must be run along with the following Windows PowerShell cmdlets, in this order:

- **Install-SPHelpCollection**
- **Initialize-SPResourceSecurity**
- **Install-SPService**
- **Install-SPFeature -AllExistingFeatures**
- **Install-SPApplicationContent**

Create a Web Application using Windows PowerShell

Use the **New-SPWebApplication** cmdlet to create the Web application and a new content database. Use the following example as a guide.

To create a Web application

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

```
New-SPWebApplication -Name <InternetSite> -ApplicationPoolAccount (Get-
SPManagedAccount <DOMAIN\UserName>
```

Where:

- *<InternetSite>* is name of the Web application.
- *Domain\UserName* is the name of the application pool account.

For more information, see New-SPWebApplication (*http://technet.microsoft.com/library/eaeb5bed-81e7-4275-b005-aa7fc465e6d5(Office.14).aspx*).

Note:
We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

Deploy services using the SharePoint 2010 Farm Configuration Wizard

Use the SharePoint Products Configuration Wizard to deploy services on your installation. For information about services and service applications, see Service application and service management (SharePoint Server 2010).

Create a site collection by using Windows PowerShell

You create the top-level site collection by using the **New-SPSite** cmdlet. The **New-SPSite** cmdlets creates a site collection at a specific URL with a specified user as a site owner.

To create a site collection

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

   ```
   New-SPSite <SiteURL> -OwnerAlias <DOMAIN\UserName>
   ```

   Where:

   - *<SiteURL>* is the URL of the new site.
   - *<DOMAIN\UserName>* is the user login name of the site owner.

   For more information, see New-SPSite (*http://technet.microsoft.com/library/ebdadc86-0cda-49b7-a84a-5cfc6b4506b3(Office.14).aspx*).

   Note:
   We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

If you do not specify the site template to use, site owners can choose the site template when they first browse to the site. You can use the **Get-SPWebTemplate** cmdlet to display a list of templates.

For a complete list of common templates in SharePoint Server 2010, see Scripted Deployment Reference (SharePoint Server).

If you want to create additional site collections by using Windows PowerShell, you can use the **New-SPSite** cmdlet.

Note:

If you want to create a new content database with the new site, use the **New-SPContentDatabase** cmdlet or the **New-SPSite** with the **ContentDatabase** parameter.

After creating sites, you might want to configure alternate access mappings. Alternate access mappings direct users to the correct URLs during their interaction with SharePoint Server 2010 (while browsing to the home page of a SharePoint site, for example). Alternate access mappings enable SharePoint Server 2010 to map Web requests to the correct Web applications and sites, and they enable SharePoint Server 2010 to serve the correct content back to the user. For more information, see Set-SPAlternateUrl (*http://technet.microsoft.com/library/846b5eb0-f235-4970-837b-f8f2657722a9(Office.14).aspx*).

Perform additional configuration tasks

After you have installedSharePoint Server 2010, we recommend that you perform the following administrative tasks:

- Configure outgoing e-mail settings.
- Configure workflow settings.
- Configure diagnostic logging settings.
- Configure antivirus settings.

Add servers to the farm using Join-SharePointFarm

You must run the Join-SharePointFarm command on all servers you want to add to the farm. To connect to an existing configuration database and join the server to an existing server farm, type the following command on the server (after installing SharePoint Server 2010):

```
Join-SharePointFarm -DatabaseServer <String> -ConfigurationDatabaseName <String> -Passphrase <SecureString>
```

Where:

- <*String*> is the name of the database server, for example, SQL01.
- <*SecureString*> is the password of the user account in the form DOMAIN\password.

Configure the trace log

The trace log can be useful for analyzing problems that might occur. You can use events that are written to the trace log to determine what configuration changes were made in SharePoint Server 2010 before the problem occurred.

By default, SharePoint Server 2010 saves 14 days of events in the trace log files. This means that trace log files that contain events that are older than 14 days are deleted. You can use the **Set-SPLogLevel** cmdlet to configure all diagnostic logging.

You can use the Diagnostic Logging page in Central Administration to configure the maximum number of trace log files to maintain, and how long (in minutes) to capture events to each log file.

You can also specify where the log files are written or accept the default path by using the **Set-SPLogLevel** cmdlet.

Trace log files can help you troubleshoot issues related to configuration changes to the Microsoft SharePoint Foundation Search service. Because problems related to configuration changes are not always immediately discovered, we recommend that you save all trace log files that the system creates on any day that you make any configuration changes. Store these log files for some time in a safe location that will not be overwritten. We recommend that you store log files on a hard disk drive partition that is used to store log files only.

For additional information about diagnostic logging, see Configure diagnostic logging (SharePoint Server 2010).

Initial configuration (SharePoint Server 2010)

After the installation of Microsoft SharePoint Server 2010, you must perform an initial configuration. If you are using different languages in the server farm, ensure that you install the correct language packs on your Web servers. Next, you can start to configure server farm settings. The configuration of additional settings is optional, but many key features are not available unless these settings are configured. When you have created a Web application and configured the services that you want to use for this Web application, you can start to create site collections.

The articles in this section help you perform the initial configuration of SharePoint Server 2010.

- Deploy language packs (SharePoint Server 2010)

  Language packs enable site owners and site collection administrators to create SharePoint sites and site collections in multiple languages without requiring separate installations of SharePoint Server 2010. This article describes how you install language packs on Web servers.

- Configure farm settings (SharePoint Server 2010)

  This article describes how to configure additional settings in the server farm, for example outgoing and incoming e-mail, mobile account, and diagnostic logging.

- Configure services (SharePoint Server 2010)

  Individual services can be configured independently, and you can implement only the services that your organization needs. Services that are deployed are named service applications. A service application provides a resource that can be shared across sites within a farm or sometimes across multiple farms, and can be accessed by users through a hosting Web application. This article covers how to start, stop, and configure services, and how to manage and publish service applications.

- Configure a managed metadata service application (SharePoint Server 2010)

  A managed metadata service application is used to manage metadata and share content across site collections and Web applications. The service application can be configured manually or by using Windows PowerShell. This article provides links to articles that provide detailed information about how to configure the key elements of a managed metadata service application.

- Prepare to host sites (SharePoint Server 2010)

  After you have installed SharePoint Server 2010 and performed the initial configuration, you can begin to create SharePoint sites. This article describes how you create a Web application and a site collection which are the basis for creating SharePoint sites.

Deploy language packs (SharePoint Server 2010)

In this article:

- About language IDs and language packs
- Downloading language packs
- Preparing the Web servers for language packs
- Installing language packs on the Web servers
- Uninstalling language packs

Language packs enable site owners and site collection administrators to create SharePoint sites and site collections in multiple languages without requiring separate installations of Microsoft SharePoint Server 2010. You install language packs, which contain language-specific site templates, on Web servers. When an administrator creates a site or a site collection that is based on a language-specific site template, the text that appears on the site or the site collection is displayed in the site template's language. Language packs are typically used in multinational deployments where a single server farm supports people in different locations, or when sites and Web pages must be duplicated in one or more languages.

If your users are accessing Microsoft Project Server 2010 in your SharePoint Server 2010 farm and need to view their project data in another language, they will also need to install a corresponding Project Server 2010 language pack. For more information about Project Server 2010 language packs, see Deploy language packs (Project Server 2010) (*http://technet.microsoft.com/library/7ce9823a-c6f0-4bc9-986d-a59f985e43a6(Office.14).aspx*).

Note:
You cannot change an existing site, site collection, or Web page from one language to another by applying different language-specific site templates. After you use a language-specific site template for a site or a site collection, the site or site collection will always display content in the language of the original site template.

Word breakers and stemmers enable you to efficiently and effectively search across content on SharePoint sites and site collections in multiple languages without requiring separate installations of SharePoint Server 2010. Word breakers and stemmers are automatically installed on Web servers by Setup.

Important:
If you are uninstalling SharePoint Server 2010, you must uninstall all language packs before you uninstall SharePoint Server 2010.

About language IDs and language packs

When site owners or site collection administrators create sites or site collections, they can choose a language for each site or site collection.

The language that they choose has a language identifier (ID). The language ID determines the language that is used to display and interpret text that is put on the site or site collection. For example, when a site owner creates a site in French, the site's toolbars, navigation bars, lists, and column headings appear in French. Similarly, if a site owner creates a site in Arabic, the site's toolbars, navigation bars, lists, and column headings appear in Arabic. In addition, the default left-to-right orientation of the site changes to a right-to-left orientation to correctly display Arabic text.

The list of available languages that people can use to create a site or site collection is generated by the language packs that are installed on the Web servers. By default, sites and site collections are created in the language in which SharePoint Server 2010 was installed. For example, if you install the Spanish version of SharePoint Server 2010, the default language for sites, site collections, and Web pages is Spanish. If someone has to create sites, site collections, or Web pages in a language other than the default SharePoint Server 2010 language, you must install the language pack for that language on the Web servers. For example, if you are running the French version of SharePoint Server 2010, and a site owner wants to create sites in French, English, and Spanish, you must install the English and Spanish language packs on the Web servers.

Note:
By default, when a site owner creates a new Web page in a site, the site displays text in the language that is specified by the language ID.

Language packs are not bundled into multilingual installation packages. You must install a specific language pack for each language that you want to support. Also, language packs must be installed on each Web server to ensure that each Web server can render content in the specified language.

Important:
You cannot change an existing site, site collection, or Web page from one language to another by applying different language-specific site templates. After you use a language-specific site template for a site or a site collection, the site or site collection will always display content in the language of the original site template.

For a list of all the language packs available, see Language packs (Search Server 2010) (*http://technet.microsoft.com/library/584cd2bf-46ec-496b-abd5-a0d380f479c4(Office.14).aspx*)

Although a site owner specifies a language ID for a site, some user interface elements such as error messages, notifications, and dialog boxes do not display in the language that was specified. This is because SharePoint Server 2010 relies on several supporting technologies — for example, the Microsoft .NET Framework, Microsoft Windows Workflow Foundation, Microsoft ASP.NET, and Microsoft SQL Server 2005 — some of which are localized into only a limited number of languages. If a user interface element is generated by any of the supporting technologies that is not localized into the language that the site owner specified for the site, the user interface element appears in English. For example, if a site owner creates a site in Hebrew, and the .NET Framework component displays a

notification message, the notification message will not display in Hebrew because the .NET Framework is not localized into Hebrew. This situation can occur when sites are created in any language except the following: Chinese, French, German, Italian, Japanese, Korean, and Spanish.

Important:
Each language pack that you install creates a folder at %COMMONPROGRAMFILES%\Microsoft Shared\Web server extensions\14\LAYOUTS\Locale_ID that contains culture-specific data. In each locale_ID folder. you must have only one HTML error file that contains the error information that is used when a file cannot be found. Anytime a file cannot be found for any site in that culture, this file will be used. You can specify which file to use by setting the SPWebApplication.FileNotFoundPage property for each Web application. For more information see, SPWebApplication.FileNotFoundPage Property (*http://go.microsoft.com/fwlink/?LinkId=169319*)

In some cases, some text might originate from the original installation language, which can create a mixed-language experience. This kind of mixed-language experience is typically seen only by content creators or site owners and is not seen by site users.

Downloading language packs

You must perform the following steps for each language that you want to support. If you decide to download more than one language, please be aware that a unique file that has a common name is downloaded for each language. Therefore, make sure that you download each language pack to a separate folder on the hard disk so that you do not overwrite a language pack of a different language.

Important:
The Windows PowerShell help files are installed in English (en-us) by default. To view these files in the same language as the operating system, you must install the language pack for the same language that the operating system was installed in.

The language packs can be downloaded at 2010 Server Language Packs for SharePoint Server 2010, Project Server 2010, Search Server 2010, and Office Web Apps 2010 (*http://go.microsoft.com/fwlink/?LinkID=192105&clcid=0x409*).

Important

- If you are upgrading from a previous version of Microsoft SharePoint Server and you are using the **Group Approval (eApproval)** features, you must install all the following language packs before you run the SharePoint Products Configuration Wizard:
- After installing the language packs, run the following command in the %COMMONPROGRAMFILES%\Microsoft Shared\Web Server Extensions\14 folder:
- **psconfig.exe -cmd upgrade -inplace v2v**

Download the language pack

1. Download the 64-bit version of the language pack by using one of the download links.
2. On the download page, select the language that you want from the **Change Language** list, and then click **Change**.
3. Click **Download** on the Web page.
4. In the dialog box that appears, click **Save** to download a copy of the file to the local computer.

Note:

If you are uninstalling SharePoint Server 2010, you must uninstall all language packs before you uninstall SharePoint Server 2010.

Preparing the Web servers for language packs

Before you install language packs on the Web servers, you must do the following:

- Install the necessary language files on the Web servers.
- Install SharePoint Server 2010 on each of the Web servers.
- Run the SharePoint Products Configuration Wizard on each of the Web servers.

Language files are used by the operating system and provide support for displaying and entering text in multiple languages. Language files include the following:

- Keyboard files
- Input Method Editors (IMEs)
- TrueType font files
- Bitmap font files
- Code page conversion tables
- National Language Support (.nls) files
- Script engines for rendering complex scripts

By default, most language files are installed on the Windows Server 2008 operating system. However, you must install supplemental language files for East Asian languages and languages that use complex script or require right-to-left orientations. The East Asian languages include Chinese, Japanese, and Korean. The complex script and right-to-left oriented languages include Arabic, Armenian, Georgian, Hebrew, the Indic languages, Thai, and Vietnamese. Instructions for installing these supplemental language files are provided in the following procedure.

You can download the language files at Windows Server 2008 R2 Multilingual User Interface Language Packs (*http://go.microsoft.com/fwlink/?LinkId=207678*).

We recommend that you install these language files only if you must have them. The East Asian files require about 230 megabytes of hard disk space. The complex script and right-to-left languages do not use much disk space, but installing either set of files might decrease performance when you enter text.

Note

- You will need your Windows Server 2008 product disc to perform this procedure, or you will have to know the location of a shared folder that contains the operating system installation files.
- You must restart the computer after you install supplemental language files.

Install additional language files on Windows Server 2008

1. You must be a member of the Administrators group on the computer to install these language files. After the language files are installed, the languages are available to all users of the computer.
2. On the Web server, click **Start** and then **Control Panel**, and then click **Regional and Language Options**.
3. In the **Regional and Language Options** dialog box, on the **Keyboards and Languages** tab, in the **Display Language** section, click **Install/Uninstall languages**.
4. In the **Install or Uninstall Languages** dialog box, click **Install languages**.
5. On the Select the Languages to Install page, select the language to install from the list of available languages. If the language does not appear, click **Browse folder** to navigate to where you downloaded the language file. The language file is a .cab file.
6. Select all the languages that you want to install, and then click **Next**.
7. Accept the terms, and then click **Next**.
8. Click **Install**.

After you install the necessary language files on the Web servers, you have to install SharePoint Server 2010 and run the SharePoint Products Configuration Wizard. The wizard creates and configures the configuration database and performs other configuration tasks that must be done before you install language packs. For more information about how to install SharePoint Server 2010 and running the SharePoint Products Configuration Wizard, see Deployment overview (SharePoint Server 2010).

Installing language packs on the Web servers

After you install the necessary language files on the Web servers, you can install the language packs. Language packs are available as individual downloads (one download for each supported language). If you have a server farm environment and you are installing language packs to support multiple languages, you must install the language packs on each of the Web servers.

Important

- The language pack is installed in its native language. For example, the Russian language pack executable file is in Russian. The procedure that follows is for the English language pack.
- SharePoint Foundation 2010 language packs are not required for SharePoint Server 2010.

Install a language pack

1. Run setup.exe.
2. On the Read the Microsoft Software License Terms page, review the terms, select the **I accept the terms of this agreement** check box, and then click **Continue**.
3. The Setup wizard runs and installs the language pack.
4. Rerun the SharePoint Products Configuration Wizard, using the default settings. If you do not run the SharePoint Products Configuration Wizard after you install a language pack, the language pack will not be installed correctly.

Rerun the SharePoint 2010 Products Configuration Wizard

1. Click **Start**, point to **All Programs**, click **Microsoft SharePoint 2010 Products**, and then click **SharePoint 2010 Products Configuration Wizard**.
2. On the Welcome to SharePoint Products page, click **Next**.
3. Click **Yes** in the dialog box that alerts you that some services might have to be restarted during configuration.
4. On the Modify Server Farm Settings page, click **Do not disconnect from this server farm**, and then click **Next**.
5. If the Modify SharePoint Central Administration Web Administration Settings page appears, do not change any of the default settings, and then click **Next**.
6. On the Completing the SharePoint Products and Technologies Configuration Wizard page, click **Next**.
7. On the Configuration Successful page, click **Finish**.

When you install language packs, the language-specific site templates are installed in the %COMMONPROGRAMFILES%\Microsoft Shared\Web server extensions\14\template*number* directory, where *number* is the Language ID for the language that you are installing. For example, the U.S. English language pack installs to the %COMMONPROGRAMFILES%\Microsoft Shared\Web Server Extensions\14\template\1033 directory. After you install a language pack, site owners and site collection administrators can create sites and site collections based on the language-specific site templates by specifying a language when they are creating a new SharePoint site or site collection.

Important:
After you install a new language pack, you must deactivate and then reactivate any language-specific features before you use the new language pack.

Uninstalling language packs

If you no longer have to support a language for which you have installed a language pack, you can remove the language pack by using the Control Panel. Removing a language pack removes the language-specific site templates from the computer. All sites that were created that have those language-specific site templates will no longer work (the URL will produce a HTTP 500 - Internal server error page). Reinstalling the language pack will make the site functional.

Note:

You cannot remove the language pack for the version of SharePoint Server 2010 that you have installed on the server. For example, if you are running the Japanese version of SharePoint Server 2010, you cannot uninstall the Japanese language support for SharePoint Server 2010.

Configure farm settings (SharePoint Server 2010)

After the initial installation of Microsoft SharePoint Server 2010, you can configure several additional settings. Some of these settings include configuring usage and health data collection to ensure that you collect relevant data to analyze, configuring several diagnostic logging settings to help with troubleshooting, and configuring a mobile account so that users can receive alerts by means of Short Message Service (SMS) when changes have been made in a SharePoint list or item. The configuration of additional settings is optional, but many key features are not available unless these settings are configured.

The articles in this section describe how you configure the server farm.

- Configure usage and health data collection (SharePoint Server 2010)

  This article describes how to configure usage and health data collection in SharePoint Server 2010.
- Configure diagnostic logging (SharePoint Server 2010)

  This article describes how to configure diagnostic logging that might be required after initial deployment or upgrade and possibly throughout the system's life cycle.
- E-mail integration (SharePoint Server 2010)

  This article describes how to configure incoming and outgoing e-mail in the server farm.
- Configure a mobile account (SharePoint Server 2010)

  This article discusses how to configure and manage a mobile account for SharePoint Server 2010 to enable users to subscribe to alerts that are sent by using SMS.
- Install and configure Remote BLOB Storage (RBS) with the FILESTREAM provider (SharePoint Server 2010)

  This article describes how to install and configure Remote BLOB Storage (RBS) for a Microsoft SQL Server 2008 database server that supports a Microsoft SharePoint Server 2010 farm.
- Configure and use the Exchange connector (SharePoint Server 2010)

  This article describes how to create a crawl rule and add a content source for crawling Microsoft Exchange Server public folders.
- Deploy Microsoft Business Intelligence Indexing Connector (SharePoint Server 2010)

  These articles describe deployment steps for Microsoft Business Intelligence Indexing Connector.

Configure usage and health data collection (SharePoint Server 2010)

This article provides information about configuring usage and health data collection in Microsoft SharePoint Server 2010.

The system writes usage and health data to the logging folder and to the logging database. To configure settings for the logging database, you must use Windows PowerShell.

In this article:

- Configure usage and health data collection by using Central Administration
- Configure usage data collection by using Windows PowerShell
- To configure usage data collection for a specific event type by using Windows PowerShell
- Log usage data in a different logging database by using Windows PowerShell

Note:
You cannot configure health data collection settings by using Windows PowerShell.

Configure usage and health data collection by using Central Administration

You can use only Central Administration to configure usage and health data collection.

To configure usage and health data collection by using Central Administration

1. Verify that the user account performing this procedure is a member of the Farm Administrators group.

   Note:
   The usage and health data settings are farm-wide and cannot be set for individual servers in the farm.
2. In Central Administration, on the Home page, click **Monitoring**.
3. On the Monitoring page, in the **Reporting** section, click **Configure usage and health data collection**.
4. On the Configure usage and health data collection page, in the **Usage data collection** section, enable usage data collection by selecting the **Enable usage data collection** text box.
5. In the **Event Selection** section, select the events to log by selecting the check box next to the events in the **Events to log** list.

Note:

Logging uses system resources and can affect performance and disk usage. Only log those events for which you want regular reports. For ad hoc reports or investigations, enable logging for specific events, and then disable logging for the events after the report or investigation is complete.

6. In the **Usage data collection settings** section, type the path of the folder you want usage and health information to be written to in the **Log file location** box. The path that you specify must exist on all farm servers.

   Note:
   These settings are applied to all events. To set event collection settings for individual event types, you must use Windows PowerShell.

7. Type the maximum disk space for the logs in gigabytes (between 1 and 20 GB) in the **Maximum log file size** box.
8. In the **Health data collection** section, select the **Enable health data collection** check box. To change the collection schedules, click **Health Logging Schedule**. A list of timer jobs that collect health data is listed. Click any of the timer jobs to change its schedule, or disable that timer job.
9. In the **Logging Database Server** section, to change the authentication used, select either the **Windows authentication** or **SQL authentication** option.

   Note:
   To change the **Database Server** and **Database Name** values, you must use Windows PowerShell.

Configure usage data collection by using Windows PowerShell

Note:
You can configure usage data collection by using Windows PowerShell, but you cannot configure health data collection by using Windows PowerShell.

To configure usage data collection by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt (that is, PS C:\>), type the following command, and then press ENTER:

```
Set-SPUsageService [-LoggingEnabled {1 | 0}] [-UsageLogLocation <Path>] [-
UsageLogMaxSpaceGB <1-20>] [-Verbose]
```

Important:

You must specify a path for `UsageLogLocation` that exists on all farm servers.

Enable usage data logging by typing `-LoggingEnabled 1`. Specify the maximum amount of drive space used for logging with the `UsageLogMaxSpaceGB` parameter.

For more information, see Set-SPUsageService (*http://technet.microsoft.com/library/c758e682-3a57-4d47-a932-56a96b56614d(Office.14).aspx*).

To configure usage data collection for a specific event type by using Windows PowerShell

The event types listed on the Configure usage and health data collection page in Central Administration are the same as Usage Definitions in Windows PowerShell. You can use only Windows PowerShell to configure usage definitions individually. Moreover, you can configure only the `DaysRetained` setting.

To configure usage data logging for a specific event type using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt (that is, PS C:\>), type the following command, and then press ENTER:

   ```
   Set-SPUsageDefinition -Identity <GUID> [-Enable] [-DaysRetained <1-30>] [-Verbose]
   ```

   Use the `Enabled` switch to enable usage logging for this usage definition. Use `DaysRetained` to specify how long the usage data is retained in the log before being deleted. The range is 1 to 30 days. To view the progress of the command, use the `Verbose` parameter.

For more information, see Set-SPUsageDefinition (*http://technet.microsoft.com/library/05ff2fea-1955-4537-8cfb-1b0e3890e1be(Office.14).aspx*).

Log usage data in a different logging database by using Windows PowerShell

Note:

You can use only Windows PowerShell to change this setting.

To log usage data in a different logging database by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt (that is, PS C:\>), type the following command, and then press ENTER:

   ```
   Set-SPUsageApplication -DatabaseServer <Database server name> -DatabaseName <Database name> [-DatabaseUsername <User name>] [-DatabasePassword <Password>] [-Verbose]
   ```

   You must specify the value for the `DatabaseServer` parameter, even if the new database is on the same database server as the old one. You must use both the `DatabaseUsername` and the `DatabasePassword` parameters if the database owner is a different user account that the one you with which you are logged on. To view the progress of the command, use the `Verbose` parameter.

For more information, see Set-SPUsageApplication (*http://technet.microsoft.com/library/4b918524-5af9-4265-9dcc-470f70fbaaba(Office.14).aspx*).

See Also

Monitoring overview (SharePoint Server 2010) (*http://technet.microsoft.com/library/6483e4c3-dc7e-49fd-92a7-15e1bb63d432(Office.14).aspx*)

Configure diagnostic logging (SharePoint Server 2010)

This article provides information about configuring diagnostic logging in Microsoft SharePoint Server 2010.

In this article:

- Best practices
- Configure diagnostic logging by using Central Administration
- Configure diagnostic logging by using Windows PowerShell

Best practices

The SharePoint Server 2010 environment might require configuration of the diagnostic loggings settings after initial deployment or upgrade and possibly throughout the system's life cycle. The guidelines in the following list can help you form best practices for the specific environment.

- **Change the drive that logging writes to.** By default, diagnostic logging is configured to write logs to the same drive and partition that SharePoint Server 2010 was installed on. Because diagnostic logging can use lots of drive space and writing to the logs can affect drive performance, you should configure logging to write to a drive that is different from the drive on whichSharePoint Server 2010 was installed. You should also consider the connection speed to the drive that logs are written to. If verbose-level logging is configured, lots of log data is recorded. Therefore, a slow connection might result in poor log performance.
- **Restrict log disk space usage.** By default, the amount of disk space that diagnostic logging can use is not limited. Therefore, limit the disk space that logging uses to make sure that the disk does not fill up, especially if you configure logging to write verbose-level events. When the disk restriction is used up, the oldest logs are removed and new logging data information is recorded.
- **Use the Verbose setting sparingly.** You can configure diagnostic logging to record verbose-level events. This means that the system will log every action that SharePoint Server 2010 takes. Verbose-level logging can quickly use drive space and affect drive and server performance. You can use verbose-level logging to record a greater level of detail when you are making critical changes and then re-configure logging to record only higher-level events after you make the change.
- **Regularly back up logs.** The diagnostic logs contain important data. Therefore, back them up regularly to make sure that this data is preserved. When you restrict log drive space usage, or if you keep logs for only a few days, log files are automatically deleted, starting with the oldest files first, when the threshold is met.

- **Enable event log flooding protection.** Enabling this setting configures the system to detect repeating events in the Windows event log. When the same event is logged repeatedly, the repeating events are detected and suppressed until conditions return to a typical state.

You can set the level of diagnostic logging for the event log and for the trace log. This will limit the types and amount of information that will be written to each log. The following tables define the levels of logging available for the event log and trace log:

Event log levels

Level	Definition
None	No logging occurs.
Critical	This message type indicates a serious error that has caused a major failure in the solution.
Error	This message type indicates an urgent condition. All error events should be investigated.
Warning	This message type indicates a potential problem or issue that might require attention. Warning messages should be reviewed and tracked for patterns over time.
Information	Information messages do not require any action, but they can provide valuable data for monitoring the state of your solution.
Verbose	This event log level corresponds to lengthy events or messages.

Trace log levels

Level	Definition
None	No trace logs are written.
Unexpected	This level is used to log messages about events that cause solutions to stop processing. When set to log at this level, the log will only include events at this level.
Monitorable	This level is used to log messages about any unrecoverable events that limit the solution's functionality but do not stop the application. When set to log at this level, the log will also include critical errors (Unexpected level).
High	This level is used to log any events that are unexpected but which do not stall the processing of a solution. When set to log at this level, the log will include warnings, errors (Monitorable level) and critical errors (Unexpected level).
Medium	When set to this level, the trace log includes everything except Verbose messages. This level is used to log all high-level information about operations that were performed. At this level, there is enough detail logged to construct the data flow and sequence of operations. This level of logging could be used by administrators or support professionals to troubleshoot issues.

Level	Definition
Verbose	When set to log at this level, the log includes messages at all other levels. Almost all actions that are performed are logged when you use this level. Verbose tracing produces many log messages. This level is typically used only for debugging in a development environment.

Configure diagnostic logging by using Central Administration

You can use Central Administration to configure diagnostic logging.

To configure diagnostic logging by using Central Administration

1. Verify that the user account that is performing this procedure is a member of the Farm Administrators SharePoint group.
2. In Central Administration, on the Home page, click **Monitoring**.
3. On the Monitoring page, in the **Reporting** section, click **Configure diagnostic logging**.
4. On the Diagnostic Logging page, in the **Event Throttling** section, you can configure event throttling as follows:

   To configure event throttling for all categories:

   a. Select the **All Categories** check box.
   b. Select the event log level from the **Least critical event to report to the event log** list.
   c. Select the trace log level from the **Least critical event to report to the trace log** list.

   To configure event throttling for one or more categories:

   a. Select the check boxes next to the categories that you want.
   b. Select the event log level from the **Least critical event to report to the event log** list.
   c. Select the trace log level from the **Least critical event to report to the trace log** list.

   To configure event throttling for one or more sub-categories (you can expand one or more categories and select any sub-category):

   a. Click **(+)** next to the category to expand the category.
   b. Select the check box next to the sub-category.
   c. Select the event log level from the **Least critical event to report to the event log** list.
   d. Select the trace log level from the **Least critical event to report to the trace log** list.

   To configure event throttling for all categories back to default settings:

   a. Select the **All Categories** check box.
   b. Select **Reset to default** from the **Least critical event to report to the event log** list.
   c. Select **Reset to default** from the **Least critical event to report to the trace log** list.

5. In the **Event Log Flood Protection** section, select the **Enable Event Log Flood Protection** check box.
6. In the **Trace Log** section, in the **Path** box, type the path of the folder to which you want logs to be written.
7. In the **Number of days to store log files** box, type the number of days (1-366) that you want logs to be kept. After this time, logs will automatically be deleted.
8. To restrict how much disk space the logs can use, select the **Restrict Trace Log disk space usage** check box, and then type the number gigabytes (GB) you want to restrict log files to. When logs reach this disk size, older logs will automatically be deleted.
9. After you have made the changes that you want on the Diagnostic Logging page, click **OK**.

Configure diagnostic logging by using Windows PowerShell

You can use Windows PowerShell to configure diagnostic logging.

To configure diagnostic logging by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt (that is, PS C:\>), type the following command, and then press ENTER:

   ```
   Set-SPLogLevel -TraceSeverity {None | Unexpected | Monitorable | Medium | High | Verbose} -EventSeverity {None | Information | Warning | Error | Critical | Verbose} [-Identity <Category name…>]  -Verbose
   ```

   You can use the `Identity` parameter to specify one or more categories to change — for example, Administration. If you do not specify the value for the `Identity` parameter, all categories are changed.

   To view the current settings, type `Get-SPLogLevel`, and then press ENTER.

   To set all categories back to default levels, type `Clear-SPLogLevel`, and then press ENTER.

For more information, see Set-SPLogLevel (*http://technet.microsoft.com/library/c8ede92a-f685-4140-8587-96700d1a45de(Office.14).aspx*).

Note:
We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

See Also

Monitoring overview (SharePoint Server 2010) (*http://technet.microsoft.com/library/6483e4c3-dc7e-49fd-92a7-15e1bb63d432(Office.14).aspx*)

E-mail integration (SharePoint Server 2010)

After the initial installation of Microsoft SharePoint Server 2010, you can configure incoming and outgoing e-mail. These additional settings are optional, but might be useful if you want to work with e-mail in the server farm.

- Configure incoming e-mail (SharePoint Server 2010)

  This article describes how to configure incoming e-mail so that SharePoint sites accept and archive incoming e-mail. It also describes how to configure incoming e-mail so that SharePoint sites can archive e-mail discussions as they happen, save e-mailed documents, and show e-mailed meetings on site calendars. In addition, this article describes how to configure the SharePoint Directory Management Service to provide support for e-mail distribution list creation and management.

- Configure outgoing e-mail (SharePoint Server 2010)

  This article describes how to configure outgoing e-mail so that your Simple Mail Transfer Protocol (SMTP) server sends e-mail alerts to site users and notifications to site administrators.

Configure incoming e-mail (SharePoint Server 2010)

This article describes how to configure incoming e-mail for a server farm for Microsoft SharePoint Server 2010. This article also describes how to install and configure the SMTP service that you must use to enable incoming e-mail.

In this article:

- Overview
- Install and configure the SMTP service
- Configure incoming e-mail in a basic scenario
- Configure incoming e-mail in an advanced scenario
- Prepare your environment for incoming e-mail in an advanced scenario
- Are attachments missing from e-mail messages that are sent to a SharePoint document library?

Overview

When incoming e-mail is enabled, SharePoint sites can receive and store e-mail messages and attachments in lists and libraries. This article describes two scenarios, one basic and one advanced. The basic scenario applies to a single-server farm environment and is recommended if you want to use default settings, whereas the advanced scenario applies to a single-server farm or a multiple-server farm and contains several advanced options from which to choose. For more information, see Plan incoming e-mail (SharePoint Server 2010) (*http://technet.microsoft.com/library/ca092ed2-4aa2-4c2e-b273-661ca6a76e01(Office.14).aspx).*

Before you perform these procedures, confirm that:

- Your system is running SharePoint Server 2010.
- You have read Plan incoming e-mail (SharePoint Server 2010) (*http://technet.microsoft.com/library/ca092ed2-4aa2-4c2e-b273-661ca6a76e01(Office.14).aspx*).
- If you are using the basic scenario, each SharePoint front-end Web server must be running the Simple Mail Transfer Protocol (SMTP) service and the Microsoft SharePoint Foundation Web Application service.
- If you are using the advanced scenario, you can use one or more servers in the server farm to run the SMTP service and to have a valid SMTP server address. Alternatively, you must know the name of a server outside the farm that is running the SMTP service and the location of the e-mail drop folder.

If you have not installed and configured the SMTP service and do not choose to use an e-mail drop folder, you must perform the following procedures before you configure incoming e-mail:

- Install and configure the SMTP service.

Install and configure the SMTP service

Incoming e-mail for SharePoint Server 2010 uses the SMTP service. You can use the SMTP service in one of two ways. You can install the SMTP service on one or more servers in the farm, or administrators can provide an e-mail drop folder for e-mail that is forwarded from the service on another server. For more information about the e-mail drop folder option, see Plan incoming e-mail (SharePoint Server 2010) (*http://technet.microsoft.com/library/ca092ed2-4aa2-4c2e-b273-661ca6a76e01(Office.14).aspx*).

Install the SMTP service

If you are not using a drop folder for e-mail, the SMTP service must be installed on every front-end Web server in the farm that you want to configure for incoming e-mail. To install the SMTP service, use the Add Features Wizard in Server Manager. After the procedure is complete, a default SMTP configuration has been created. You can customize this default SMTP configuration to meet the requirements of your environment.

To install the SMTP service

1. Verify that you have the following administrative credentials:
   - You must be a member of the Administrators group on the local computer.
2. Click **Start**, point to **Administrative Tools**, and then click **Server Manager**.
3. In Server Manager, click **Features**.
4. In **Features Summary**, click **Add Features** to open the Add Features Wizard.
5. On the Select Features page, select **SMTP Server**.
6. In the **Add Features Wizard** dialog box, click **Add Required Features**, and then click **Next**.
7. On the Confirm Installation Selections page, click **Install**.
8. On the Installation Results page, ensure that the installation finished successfully, and then click **Close**.

Install IIS 6.0 Management tools

To manage the SMTP service on Windows Server 2008 and Windows Server 2008 R2, you must use Internet Information Services (IIS) 6.0 Manager.

To install IIS 6.0 Manager

1. Verify that you have the following administrative credentials:
   - You must be a member of the Administrators group on the local computer.

2. Click **Start**, point to **Administrative Tools**, and then click **Server Manager**.
3. In Server Manager, click **Roles**.
4. In Role Services, click **Add Role Services**.
5. On the Select Role Services page, select **Management Tools** and **IIS 6 Management compatibility**, and then click **Install**.

Configure the SMTP service

After you install the SMTP service, you configure it to accept e-mail from the mail server for the domain. You can decide to accept relayed e-mail from all servers except those that you specifically exclude. Alternatively, you can block e-mail from all servers except those that you specifically include. You can include servers individually, or in groups by subnet or domain.

After you configure the service, set it to start automatically.

To configure the SMTP service

1. Verify that you have the following administrative credentials:
   - You must be a member of the Administrators group on the local computer.
2. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) 6.0 Manager**.
3. In IIS Manager, expand the server name that contains the SMTP server that you want to configure.
4. Right-click the SMTP virtual server that you want to configure, and then click **Start**.
5. Right-click the SMTP virtual server that you want to configure, and then click **Properties**.
6. On the **Access** tab, in the **Access control** area, click **Authentication**.
7. In the **Authentication** dialog box, verify that **Anonymous access** is selected.
8. Click **OK**.
9. On the **Access** tab, in the **Relay restrictions** area, click **Relay**.
10. To enable relaying from any server, click **All except the list below**.
11. To accept relaying from one or more specific servers, follow these steps:
    a. Click **Only the list below**.
    b. Click **Add**, and then add servers one at a time by IP address, or in groups by using a subnet or domain.
    c. Click **OK** to close the **Computer** dialog box.
12. Click **OK** to close the **Relay Restrictions** dialog box.
13. Click **OK** to close the **Properties** dialog box.

To set the SMTP service to start automatically

1. Click **Start**, point to **Administrative Tools**, and then click **Services**.
2. In Services, right-click **Simple Mail Transfer Protocol (SMTP)**, and then select **Properties**.
3. In the **Simple Mail Transfer Protocol (SMTP) Properties** dialog box, on the **General** tab, in the **Startup type** list, select **Automatic**.
4. Click **OK**.

Configure incoming e-mail in a basic scenario

You can use the following procedure to configure incoming e-mail in a basic scenario by selecting the **Automatic** settings mode and using the default settings. After the procedure is complete, users can send e-mail to lists and libraries.

To configure incoming e-mail in a basic scenario

1. Verify that you have the following administrative credentials:
   - You must be a member of the Administrators group on the computer that is running the SharePoint Central Administration Web site.
2. In Central Administration, click **System Settings**.
3. On the System Settings page, in the **E-Mail and Text Messages (SMS)** section, click **Configure incoming e-mail settings**.
4. If you want to enable sites on this server to receive e-mail, on the Configure Incoming E-Mail Settings page, in the **Enable Incoming E-Mail** section, click **Yes**.
5. Select the **Automatic** settings mode.
6. In the **Incoming E-Mail Server Display Address** section, in the **E-mail server display address** box, type a display name for the e-mail server, for example, mail.fabrikam.com.
7. Use the default settings for all other sections, and then click **OK**.

After you configure incoming e-mail, users who have Manage Lists permissions can configure e-mail–enabled lists and document libraries.

Configure incoming e-mail in an advanced scenario

You can use the following procedure to configure incoming e-mail in an advanced scenario by selecting the **Advanced** settings mode and additional options that you want to use for your incoming e-mail environment. After the procedure is complete, users can send e-mail to lists and libraries.

Note:

You can also use the **Automatic** settings mode in an advanced scenario. The main difference is that in the **Automatic** settings mode, you can select to receive e-mail that has been routed through a safe-e-mail server application, whereas in the **Advanced** settings mode, you can instead specify a drop folder. For more information, see Plan incoming e-mail (SharePoint Server 2010) (*http://technet.microsoft.com/library/ca092ed2-4aa2-4c2e-b273-661ca6a76e01(Office.14).aspx*).

Several of these steps mention pre-requisite procedures that are documented in the Prepare your environment for incoming e-mail in an advanced scenario (*http://technet.microsoft.com/library/3ff64de4-b32d-41a5-887e-f5356358e628.aspx#section5*) section of this article.

To configure incoming e-mail in an advanced scenario

1. Verify that you have the following administrative credentials:
   - You must be a member of the Administrators group on the computer that is running the SharePoint Central Administration Web site.
2. In Central Administration, click **System Settings**.
3. On the System Settings page, in the **E-Mail and Text Messages (SMS)** section, click **Configure incoming e-mail settings**.
4. If you want to enable sites on this server to receive e-mail, on the Configure Incoming E-mail Settings page, in the **Enable Incoming E-Mail** section, click **Yes**.
5. Select the **Advanced** settings mode.

   If you select **Advanced**, you can specify a drop folder instead of using an SMTP server.

   Note:
   You can also select the **Automatic** settings mode and select whether to use Directory Management Service and whether to accept e-mail from all e-mail servers or from several specified e-mail servers. For more information, see Plan incoming e-mail (SharePoint Server 2010) (*http://technet.microsoft.com/library/ca092ed2-4aa2-4c2e-b273-661ca6a76e01(Office.14).aspx*).
6. If you want to connect to Directory Management Service, in the **Directory Management Service** section, click **Yes**.

   If you select this option, you must first configure Active Directory Domain Services (AD DS). If you use Exchange Server, you must also configure the DNS Manager and add an SMTP connector. For more information, see the "Configure AD DS to be used with Directory Management Service", "Configure DNS Manager", and "Add an SMTP connector in Exchange Server 2010" sections later in this article.

   a. In the **Active Directory container where new distribution groups and contacts will be created** box, type the name of the container in the format **OU=***ContainerName***, DC=***domain***, DC=***com*, where *ContainerName* is the name of the OU in AD DS, *domain* is the second-level domain, and *com* is the top-level domain.

Note:
The application pool identity account for Central Administration must be delegated the **Create, delete, and manage user accounts** task for the container. Access is configured in the properties for the OU in AD DS.

b. In the **SMTP mail server for incoming mail** box, type the name of the SMTP mail server. The server name must match the FQDN in the A resource record entry for the mail server in DNS Manager.

c. To accept only messages from authenticated users, click **Yes** for **Accept messages from authenticated users only**. Otherwise, click **No**.

d. To enable users to create distribution groups from SharePoint sites, click **Yes** for **Allow creation of distribution groups from SharePoint sites**. Otherwise, click **No**.

e. Under **Distribution group request approval settings**, select the actions that will require approval. Actions include the following:

- **Create new distribution group**
- **Change distribution group e-mail address**
- **Change distribution group title and description**
- **Delete distribution group**

7. If you want to use a remote Directory Management Service, select **Use remote**.

   If you select this option and you are using Exchange Server, you must configure the DNS Manager and add an SMTP connector. For more information, see the "Configure DNS Manager" and "Add an SMTP connector in Exchange Server 2010" sections later in this article. The AD DS has most likely already been configured, so you do not need to do this.

   a. In the **Directory Management Service URL** box, type the URL of the Directory Management Service that you want to use. The URL is typically in the following format: http://*server:adminport*/_vti_bin/SharePointEmailWS.asmx.

   b. In the **SMTP mail server for incoming mail** box, type the name of the SMTP mail server. The server name must match the FQDN in the A resource record entry for the mail server in DNS Manager on the domain server.

   c. To accept messages from authenticated users only, click **Yes** for **Accept messages from authenticated users only**. Otherwise, click **No**.

   d. To allow creation of distribution groups from SharePoint sites, click **Yes** for **Allow creation of distribution groups from SharePoint sites**. Otherwise, click **No**.

8. If you do not want to use Directory Management Service, click **No**.

9. In the **Incoming E-Mail Server Display Address** section, in the **E-mail server display address** box, type a display name for the e-mail server (for example, mail.fabrikam.com). You typically use this option together with the Directory Management Service.

   Tip:

You can specify the e-mail server address that is displayed when users create an incoming e-mail address for a list or group. Use this setting together with Directory Management Service to provide an e-mail server address that is easy to remember.

10. In the **E-Mail Drop Folder** section, in the **E-mail drop folder** box, type the name of the folder from which SharePoint 2010 Timer service retrieves incoming e-mail from the SMTP service. If you select this option, ensure that you configure the necessary permissions to the e-mail drop folder. For more information, see the "Configure permissions to the e-mail drop folder" section later in this article.

    It is useful to have a dedicated e-mail drop folder if the default e-mail drop folder is full or almost full.

    Ensure that the logon account for the SharePoint 2010 Timer service has Modify permissions on the e-mail drop folder. For more information, see "To configure e-mail drop folder permissions for the logon account for the SharePoint 2010 Timer service" procedure later in this article.

    Note:
    This option is available only if you selected **Advanced** settings mode.

11. In the **Safe E-Mail Servers** section, select whether you want to accept e-mail from all e-mail servers or from several specified e-mail servers.

    Note:
    This option is available only if you selected **Automatic** settings mode.

12. Click **OK**.

After you configure incoming e-mail, site administrators can configure e-mail–enabled lists and document libraries.

If you selected Directory Management Service, contact addresses that are created for document libraries appear automatically in Active Directory Users and Computers. The addresses are displayed in the OU of AD DS for SharePoint Server 2010 and must be managed by the administrator of AD DS. The AD DS administrator can add more e-mail addresses for each contact. For more information about AD DS, see Using Active Directory Service (*http://go.microsoft.com/fwlink/?LinkId=151348*).

Alternatively, the Exchange Server computer can be configured by adding a new Exchange Server Global recipient policy. The policy automatically adds external addresses that use the second-level domain name and not the subdomain or host name for SharePoint Server 2010. For more information about how to manage Exchange Server, see Recipient Configuration Node (*http://go.microsoft.com/fwlink/?LinkId=151348*).

Prepare your environment for incoming e-mail in an advanced scenario

Before you configure incoming e-mail in an advanced scenario, you need to perform additional procedures depending on how you want your incoming e-mail environment to work.

If you want to use Directory Management Service, you must first configure AD DS, and if you use Exchange Server, you must also configure the DNS Manager and add an SMTP connector.

If you want to use a specific e-mail drop folder, ensure that you configure the necessary permissions to the e-mail drop folder.

In this section:

- Configure AD DS to be used with Directory Management Service
- Configure DNS Manager
- Add an SMTP connector in Microsoft Exchange Server 2010
- Configure permissions to the e-mail drop folder

Configure AD DS to be used with Directory Management Service

If you plan to use Directory Management Service you should first create an organizational unit (OU) and make the necessary configurations in AD DS.

To use Directory Management Service on a SharePoint farm or on a remote server farm, you must configure the application pool identity account for the SharePoint Central Administration Web site to have the **Create, delete, and manage user accounts** user right to the container that you specify in AD DS. The preferred way to do this is by assigning the right to the application pool identity account for the SharePoint Central Administration Web site. An AD DS administrator must set up the OU and assign the **Create, delete, and manage user accounts** right to the container. The advantage of using Directory Management Service on a remote server farm is that you do not have to assign rights to the OU for multiple farm service accounts.

The following procedures are performed on a domain controller that runs Windows Server 2008 with DNS Manager. In some deployments, these applications might run on multiple servers in the same domain.

To create an OU in AD DS

1. Verify that you have the following administrative credentials:
    - You must be a member of the Domain Administrators group or a delegated authority for domain administration on the domain controller that is running DNS Manager.
2. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Users and Computers**.
3. In Active Directory Users and Computers, right-click the folder for the second-level domain that contains your server farm, point to **New**, and then click **Organizational Unit**.

4. Type the name of the OU, and then click **OK**.

   After you create the OU, you must delegate the **Create, delete, and manage user accounts** right to the container of the OU to manage the user accounts.

To delegate the right to the application pool identity account for Central Administration

1. Verify that you have the following administrative credentials:
   - You must be a member of the Domain Administrators group or the Enterprise Administrators group in AD DS, or a delegated authority for domain administration.
2. In Active Directory Users and Computers, find the OU that you created.
3. Right-click the OU, and then click **Delegate control**.
4. On the Welcome page of the Delegation of Control Wizard, click **Next**.
5. On the Users and Groups page, click **Add**, and then type the name of the application pool identity account that the Central Administration uses.
6. In the **Select Users, Computers, and Groups** dialog box, click **OK**.
7. On the Users or Groups page of the Delegation of Control Wizard, click **Next**.
8. On the Tasks to Delegate page of the Delegation of Control Wizard, select the **Create, delete, and manage user accounts** check box, and then click **Next**.
9. On the last page of the Delegation of Control Wizard, click **Finish** to exit the wizard.

To create and delete child objects, you must also delegate **Create all Child Objects** and **Delete all Child Objects** control of the OU to the application pool identity account for Central Administration. After this procedure is complete, the application pool identity account for Central Administration has **Create all Child Objects** and **Delete all Child Objects** control on the OU, and you can enable incoming e-mail.

To delegate Create all Child Objects and Delete all Child Objects control of the OU to the application pool identity account for Central Administration

1. Verify that you have the following administrative credentials:
   - You must be a member of the Domain Administrators group or the Enterprise Administrators group in AD DS, or a delegated authority for domain administration.
2. Right-click the OU, and then click **Delegate control**.
3. In the Delegation of Control Wizard, click **Next**.
4. Click **Add**, and then type the name of the application pool identity account for Central Administration.
5. Click **OK**.
6. Click **Next**.
7. On the Tasks to Delegate page of the Delegation of Control Wizard, select **Create a custom task to delegate**, and then click **Next**.

8. Click **This folder, existing objects in this folder, and creation of new objects in this folder**, and then click **Next**.
9. In the **Permissions** section, select **Create all Child Objects** and **Delete all Child Objects**.
10. Click **Next**.
11. On the last page of the Delegation of Control Wizard, click **Finish** to exit the wizard.

Delegating **Create all Child Objects** and **Delete all Child Objects** control of the OU to the application pool identity account for Central Administration enables administrators to enable e-mail for a list. After these controls have been delegated, administrators cannot disable e-mail for the list or document library because the Central Administration account tries to delete the contact from the whole OU instead of from the list.

To avoid this problem, you must add **Delete Subtree** permissions for the application pool identity account for Central Administration. Use the following procedure to add these permissions. After this procedure is complete, you can disable incoming e-mail for a list.

To add Delete Subtree permissions for the application pool identity account for Central Administration

1. Verify that you have the following administrative credentials:
   - You must be a member of the Domain Administrators group or the Enterprise Administrators group in AD DS, or a delegated authority for domain administration.
2. In Active Directory Users and Computers, click the **View** menu, and then click **Advanced Features**.
3. Right-click the OU, and then click **Properties**.
4. In the **Properties** dialog box, click the **Security** tab, and then click **Advanced**.
5. In the **Permission Entries** area, double-click the application pool identity account for Central Administration.

   Note:
   If the application pool identity account is listed more than once, select the first one.
6. In the **Permissions** area, select **Allow**, for **Delete Subtree**.
7. Click **OK** to close the **Permissions** dialog box.
8. Click **OK** to close the **Properties** dialog box.
9. Click **OK** to close Active Directory Users and Computers.

After you add these permissions, you must restart Internet Information Services (IIS) for the farm.

For more information, see Active Directory Users, Computers, and Groups (*http://go.microsoft.com/fwlink/?LinkId=151331*).

Configure DNS Manager

If you are using Exchange Server and are routing e-mail internally in your organization, you must create a host (A) resource record in DNS Manager to associate DNS domain names of computers (or hosts) to their IP addresses.

Your organization might have already configured DNS Manager and created an A resource record. If not, then use the following procedure.

To create an A resource record for a subdomain

1. Verify that you have the following administrative credentials:
   - You must be a member of the Administrators group on the local computer.
2. In DNS Manager, select the forward lookup zone for the domain that contains the subdomain for SharePoint Server 2010.
3. Right-click the zone, and then click **New Host (A or AAAA)**.
4. In the **New Host** dialog box, in the **Name** text box, type the host or subdomain name for SharePoint Server 2010.
5. In the **Fully qualified domain name (FQDN)** text box, type the FQDN for the server that is running SharePoint Server 2010. This is typically in the format *subdomain.domain.com*.

   Note:
   Ensure that the domains that are listed under the SMTP server in IIS match the FQDN of the server that receives e-mail. If they do not match, you must create a local domain, which is described in the following procedure.
6. In the **IP address** text box, type the IP address to which you want the FQDN to resolve.
7. Click **Add Host**.
8. In the message that confirms the creation of the host record, click **OK**.
9. In the **New Host** dialog box, click **Done**.

   The A resource record now appears in DNS Manager.

If you use the **E-mail server display address** option and if the e-mail address to which you are sending e-mails is not the same as your machine name, you must create a local domain.

To create a local domain

1. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) 6.0 Manager**.
2. In IIS Manager, expand the SMTP server.
3. Right-click **Domains**, and on the **Action** menu, point to **New**, and then click **Domain**.
4. In the **New SMTP Domain Wizard** dialog box, select **Alias**, and then click **Next**.

5. In the **Domain Name** area, in the **Name** box, type the address of the mail that is to be received by this domain.

   This address must be the same as the one that you specified in step 4 in the "To Create an A Resource Record for the Subdomain" procedure, and in step 6b in the "To Configure Incoming E-Mail in an Advanced Scenario" procedure.

6. Click **Finish**.
7. In the message that confirms the creation of the host record, click **OK**.

Note:

Restart the SMTP server so that any e-mail messages that are still in the Queue folder move to the Drop folder. The messages are then sent by the SharePoint 2010 Timer service to their destination list or library.

Note:

If you are routing e-mail from outside your organization to an SMTP server, you must use an MX record. For more information, see Add a mail exchanger (MX) resource record to a zone (*http://go.microsoft.com/fwlink/?LinkId=150827*).

Add an SMTP connector in Microsoft Exchange Server 2010

An SMTP connector gives you more control over the message flow in your organization. Other reasons to use an SMTP connector are to set delivery restrictions or to specify a specific address space. If you use Exchange Server to route incoming e-mail to SharePoint lists and libraries, you must have an SMTP connector so that all mail that is sent to the SharePoint Server 2010 domain uses the SharePoint Server 2010 servers that are running the SMTP service.

Use the following procedure to add an SMTP connector in Exchange Server. After the procedure is complete, the SMTP connector ensures that incoming e-mail messages are sent to the correct list and library in the farm.

To add an SMTP connector in Exchange Server

1. Verify that you have the following administrative credentials:
   - You must be a member of the Administrators group on the computer that is running Exchange Server.
2. In Exchange Management Console, expand the Organization Configuration group, right-click **Hub Transport**, point to **New Send Connector**.

   The **New Send Connector** wizard appears.
3. On the Introduction page, do the following and then click **Next**:
   a. In the **Name** box, type a name for the SMTP connector.
   b. In the **Select the intended use for this Send connector** box, select the **Custom** usage type for the connector.

4. On the Address Space page, click **Add**, and then click **SMTP Address Space**.
5. In the **SMTP Address Space** dialog box, do the following:
   a. In the **Address** box, type an e-mail domain for the connector.
   b. In the **Cost** box, assign an appropriate cost. By default, the cost is 1.
6. Click **OK** to return to the Address Space page, and then click **Next**.
7. On the Network settings page, select **Use domain name system (DNS) "MX" records to route mail automatically**, and then click **Next**.
8. On the Source Server page, click **Next**.

   The Source server page only appears on Hub Transport servers. By default, the Hub Transport server that you are currently working on is listed as a source server.
9. On the New Connector page, review your options and then click **New** to create the new send connector.
10. On the Completion page, ensure that the send connector was created, and then click **Finish**.

    In the Hub Transport pane, you can see that the send connector has been enabled automatically.

For more in-depth information, see Create an SMTP Send Connector (*http://go.microsoft.com/fwlink/?LinkId=195321*).

Configure permissions to the e-mail drop folder

You can specify a particular e-mail drop folder, which enables SharePoint Server 2010 to retrieve incoming e-mail from a network share on another server. You can use this option if you do not want to use an SMTP service. However, the drawback of using this option is that SharePoint Server 2010 cannot detect configuration changes on the remote e-mail server that is delivering e-mail to the drop folder. The result is that SharePoint Server 2010 cannot retrieve e-mail if the location of the e-mail messages has changed. However, this feature is useful if the default e-mail drop folder is full or almost full.

If you specified an e-mail drop folder, you must ensure that the application pool identity accounts for Central Administration and for the Web application have the required permissions to the e-mail drop folder.

Configure e-mail drop folder permissions for the application pool identity account for a Web application

If your deployment uses different application pool identity accounts for Central Administration and for one or more Web applications, each application pool identity account must have permissions to the e-mail drop folder. If the application pool identity account for the Web application does not have the required permissions, e-mail will not be delivered to document libraries on that Web application.

In most cases, when you configure incoming e-mail and select an e-mail drop folder, permissions are added for the following worker process groups:

- WSS_Admin_WPG, which includes the application pool identity account for Central Administration and the logon account for the SharePoint 2010 Timer service, and has Full Control permissions.
- WSS_WPG, which includes the application pool accounts for Web applications, and has Read & Execute, List Folder Contents, and Read permissions.

In some cases, these groups might not be configured automatically for the e-mail drop folder. For example, if Central Administration is running as the Network Service account, the groups or accounts that are needed for incoming e-mail will not be added when the e-mail drop folder is created. Check to find out whether these groups have been added automatically to the e-mail drop folder. If the groups have not been added automatically, you can add them or add the specific accounts that are required.

To configure e-mail drop folder permissions for the application pool identity account for a Web application

1. Verify that you have the following administrative credentials:
   - You must be a member of the Administrators group on the computer that contains the e-mail drop folder.
2. In Windows Explorer, right-click the drop folder, click **Properties**, and then click the **Security** tab.
3. On the **Security** tab, under the **Group or user names** box, click the **Edit** button.
4. In the **Permissions for Windows Explorer** dialog box, click the **Add** button.
5. In the **Select Users, Computers, or Groups** dialog box, in the **Enter the object names to select** box, type the name of the worker process group or application pool identity account for the Web application, and then click **OK**.

   Note:
   This account is listed on the **Identity** tab of the **Properties** dialog box for the application pool in IIS.
6. In the **Permissions for** *User or Group* box, next to **Modify**, select **Allow**.
7. Click **OK**.

Configure e-mail drop folder permissions for the logon account for the SharePoint 2010 Timer service

Ensure that the logon account for the SharePoint 2010 Timer service has Modify permissions on the e-mail drop folder. If the logon account for the service does not have Modify permissions, e-mail–enabled document libraries will receive duplicate e-mail messages.

To configure e-mail drop folder permissions for the logon account for the SharePoint 2010 Timer service

1. Verify that you have the following administrative credentials:
   - You must be a member of the Administrators group on the computer that contains the e-mail drop folder.
2. In Windows Explorer, right-click the drop folder, click **Properties**, and then click the **Security** tab.
3. On the **Security** tab, under the **Group or user names** box, click the **Edit** button.
4. In the **Permissions for Windows Explorer** dialog box, click the **Add** button.
5. In the **Select Users, Computers, or Groups** dialog box, in the **Enter the object names to select** box, type the name of the logon account for the SharePoint 2010 Timer service, and then click **OK**.

   Note:
   This account is listed on the **Log On** tab of the **Properties** dialog box for the service in the Services console.
6. In the **Permissions for** *User or Group* box, next to **Modify**, select **Allow**.
7. Click **OK**.

Are attachments missing from e-mail messages that are sent to a SharePoint document library?

If attachments are missing from e-mail messages that are sent to a SharePoint Server 2010 document library, it might be because you associated the document library with an e-mail address.

When you do this, Directory Management Service may not add the following two attributes:

- **internet Encoding = 1310720**
- **mAPIRecipient = false**

You must use Active Directory Service Interfaces (ADSI) to manually add these two missing attributes.

Note:
On servers that are running Windows Server 2008 or Windows Server 2008 R2, ADSI Edit is installed when you install the AD DS role to make a server a domain controller. You can also install Windows Server 2008 Remote Server Administration Tools (RSAT) on domain member servers or stand-alone servers. For more information, see Installing or Removing the Remote Server Administration Tools Pack (*http://go.microsoft.com/fwlink/?LinkId=143345*).

To add attributes by using ADSI Edit

1. Click **Start**, and then click **Run**.
2. In the **Run** dialog box, type **Adsiedit.msc**, and then click **OK**.
3. In the ADSI Edit window, expand **ADSI Edit**, expand **Domain [DomainName]**, expand **DC=DomainName, DC=com**, and then expand **CN=Users**.
4. Right-click the user name to which you want to add the missing attributes, and then click **Properties**.
5. In the **Properties** dialog box, double-click **internet Encoding** on the **Attribute Editor** tab.
6. In the **Integer Attribute Editor** dialog box, type **1310720** in the **Value** box, and then click **OK**.
7. In the **Properties** dialog box, double-click **mAPIRecipient** on the **Attribute Editor** tab.
8. In the **Boolean Attribute Editor** dialog box, click **False**, and then click **OK** two times.

See Also

Plan incoming e-mail (SharePoint Server 2010) (*http://technet.microsoft.com/library/ca092ed2-4aa2-4c2e-b273-661ca6a76e01(Office.14).aspx*)

Configure outgoing e-mail (SharePoint Server 2010)

This article describes how to configure outgoing e-mail for a farm or for a specific Web application for Microsoft SharePoint Server 2010. This article also describes how to install and configure the SMTP service that you must use to enable outgoing e-mail.

In this article:

- To install the SMTP service
- To install IIS 6.0 Management tools
- To configure the SMTP service
- To set the SMTP service to start automatically
- To configure outgoing e-mail for a farm by using Central Administration
- To configure outgoing e-mail for a farm by using the Stsadm command-line tool
- To configure outgoing e-mail for a specific Web application by using Central Administration
- To configure outgoing e-mail for a specific Web application by using the Stsadm command-line tool

After you have installed SharePoint Server 2010 and performed the initial configuration of your server farm, you can configure outgoing e-mail. Doing so enables users to create alerts to track such site items as lists, libraries, and documents. In addition, site administrators can receive administrative messages about site administrator issues, such as the information that site owners have exceeded their specified storage space. For more information, see Plan outgoing e-mail (SharePoint Server 2010) (*http://technet.microsoft.com/library/75401651-ef01-4348-878e-8a636f0b072d(Office.14).aspx*).

If you want to configure outgoing e-mail for a specific Web application, you must first configure the default outgoing e-mail for all Web applications in the farm. If you configure the outgoing e-mail for a specific Web application, that configuration will override the default configuration for all Web applications in the farm.

Note:
You cannot configure outgoing e-mail by using Windows PowerShell.

Before you perform these procedures, confirm that:

- Your system is running SharePoint Server 2010.
- One or more servers in the server farm are running the Simple Mail Transfer Protocol (SMTP) service and have a valid SMTP server address. Alternatively, you must know the name of a server outside the farm that is running the SMTP service.

If you have not installed and configured the SMTP service, you must perform the following procedures before you configure outgoing e-mail:

- Install and configure the SMTP service.

Install and configure the SMTP service

Before you can enable outgoing e-mail, you must determine which SMTP server to use. This SMTP server must be configured to allow anonymous SMTP e-mail submissions. The SMTP server can be a server in the farm or outside the farm.

Note:
If your organization does not allow anonymous SMTP e-mail messages to be sent by using Microsoft Exchange Server, you can use a local SMTP server in the SharePoint farm that accepts anonymous e-mail messages. The local SMTP server automatically authenticates the messages and then forwards them to the Exchange Server computer.

Install the SMTP service

To install the SMTP service, use the Add Features Wizard in Server Manager. After the wizard finishes, a default SMTP configuration has been created. You can customize this default SMTP configuration to meet the requirements of your organization.

Note:
If you already have the SMTP service installed on a server, you can skip the following procedure.

To install the SMTP service

1. Verify that you have the following administrative credentials: You must be a member of the Administrators group on the local computer
2. Click **Start**, point to **Administrative Tools**, and then click **Server Manager**.
3. In Server Manager, click **Features**.
4. In **Features Summary**, click **Add Features** to open the Add Features Wizard.
5. On the Select Features page, select **SMTP Server**.
6. In the **Add Features Wizard** dialog box, click **Add Required Features**, and then click **Next**.
7. On the Confirm Installation Selections page, click **Install**.
8. On the Installation Results page, ensure that the installation is complete, and then click **Close**.

Configure the SMTP service

After you install the SMTP service, you must configure the service to accept e-mail messages from servers in the farm.

You can decide to accept relayed e-mail messages from all servers except those that you specifically exclude. Alternatively, you can block messages from all servers except those that you specifically include. You can include servers individually or in groups by subnet or domain.

If you enable anonymous access and relayed e-mail messages, you increase the possibility that the SMTP server will be used to relay unsolicited commercial e-mail messages (spam). It is important to limit this possibility by carefully configuring mail servers to help protect against spam. One way that you can do this is by limiting relayed e-mail messages to a list of specific servers or to a domain, and by preventing relayed e-mail messages from all other servers.

Note:
To manage the SMTP service on Windows Server 2008, you must use Internet Information Services (IIS) 6.0 Manager. Ensure that you install IIS 6.0 Management tools in Server Manager.

To install IIS 6.0 Management tools

1. Verify that you have the following administrative credentials: You must be a member of the Administrators group on the local computer.
2. Click **Start**, point to **Administrative Tools**, and then click **Server Manager**.
3. In Server Manager, click **Roles**.
4. In the **Role Services** section, click **Add Role Services**.
5. On the Select Role Services page, select **Management Tools** and **IIS 6 Management compatibility**, and then click **Install**.

To configure the SMTP service

1. Verify that you have the following administrative credentials: You must be a member of the Administrators group on the local computer.
2. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) 6.0 Manager**.
3. In IIS Manager, expand the server name that contains the SMTP server that you want to configure.
4. Right-click the SMTP virtual server that you want to configure, and then click **Start**.
5. Right-click the SMTP virtual server that you want to configure, and then click **Properties**.
6. On the **Access** tab, in the **Access control** area, click **Authentication**.
7. In the **Authentication** dialog box, verify that **Anonymous access** is selected.
8. Click **OK**.
9. On the **Access** tab, in the **Relay restrictions** area, click **Relay**.
10. To enable relayed e-mail messages from any server, click **All except the list below**.
11. To accept relayed e-mail messages from one or more specific servers, follow these steps:
    a. Click **Only the list below**.
    b. Click **Add**, and then add servers one at a time by IP address, or in groups by using a subnet or domain.

c. Click **OK** to close the **Computer** dialog box.

12. Click **OK** to close the **Relay Restrictions** dialog box.
13. Click **OK** to close the **Properties** dialog box.

Note:

Ensure that the SMTP service is running and set to start automatically. To do this, use the following procedure.

To set the SMTP service to start automatically

1. Click **Start**, point to **Administrative Tools**, and then click **Services**.
2. In **Services**, right-click **Simple Mail Transfer Protocol (SMTP)**, and then select **Properties**.
3. In the **Simple Mail Transfer Protocol (SMTP) Properties** dialog box, on the **General** tab, in the **Startup type** list, select **Automatic**.
4. Click **OK**.

Configure outgoing e-mail for a farm

You can configure outgoing e-mail for a farm by using the SharePoint Central Administration Web site or by using the Stsadm command-line tool. Use the following procedures to configure outgoing e-mail. After you complete the procedures, end users can track changes and updates to individual site collections. In addition, site administrators can, for example, receive notices when users request access to a site.

To configure outgoing e-mail for a farm by using Central Administration

1. Verify that you have the following administrative credentials: You must be a member of the Farm Administrators group on the computer that is running the SharePoint Central Administration Web site.
2. In Central Administration, click **System Settings**.
3. On the System Settings page, in the **E-Mail and Text Messages (SMS)** section, click **Configure outgoing e-mail settings**.
4. On the Outgoing E-Mail Settings page, in the **Mail Settings** section, type the SMTP server name for outgoing e-mail (for example, mail.example.com) in the **Outbound SMTP server** box.
5. In the **From address** box, type the e-mail address as you want it to be displayed to e-mail recipients.
6. In the **Reply-to address** box, type the e-mail address to which you want e-mail recipients to reply.
7. In the **Character set** list, select the character set that is appropriate for your language.
8. Click **OK**.

To configure outgoing e-mail for a farm by using the Stsadm command-line tool

1. Verify that you have the following administrative credentials: You must be a member of the Administrators group on the local computer.
2. On the drive on which SharePoint Products and Technologies is installed, change to the following directory: %COMMONPROGRAMFILES%\Microsoft shared\Web server extensions\14\Bin.
3. Type the following command, and then press ENTER:

   stsadm -o email

   -outsmtpserver <*SMTP server name*>

   -fromaddress <*valid e-mail address*>

   -replytoaddress <*valid e-mail address*>

   -codepage <*valid code page*>

Example

stsadm -o email -outsmtpserver mail.example.com -fromaddress someone@example.com -replytoaddress someone@example.com -codepage 65001

For more information, see Email: Stsadm operation (Office SharePoint Server) (*http://go.microsoft.com/fwlink/?LinkId=150045*).

Configure outgoing e-mail for a specific Web application

You can configure outgoing e-mail for a specific Web application by using the Central Administration Web site or by using the Stsadm command-line tool. Use the following procedures to configure outgoing e-mail. After you complete the procedures, end users can track changes and updates to individual site collections. In addition, site administrators can, for example, receive notices when users request access to a site.

Note:
If you want to configure outgoing e-mail for a specific Web application, you must first configure the default outgoing e-mail for all Web applications in the farm. If you configure the outgoing e-mail for a specific Web application, that configuration will override the default configuration for all Web applications in the farm.

To configure outgoing e-mail for a specific Web application by using Central Administration

1. Verify that you have the following administrative credentials: You must be a member of the Farm Administrators group on the computer that is running the SharePoint Central Administration Web site.

2. In Central Administration, in the **Application Management** section, click **Manage web applications**.
3. On the Web Applications Management page, select a Web application, and then in the **General Settings** group on the Ribbon, click **Outgoing E-mail**.
4. On the Web Application Outgoing E-Mail Settings page, in the **Mail Settings** section, type the SMTP server name for outgoing e-mail (for example, mail.fabrikam.com) in the **Outbound SMTP server** box.
5. In the **From address** box, type the e-mail address (for example, the site administrator alias) as you want it to be displayed to e-mail recipients.
6. In the **Reply-to address** box, type the e-mail address (for example, a help desk alias) to which you want e-mail recipients to reply.
7. In the **Character set** list, click the character set that is appropriate for your language.
8. Click **OK**.

To configure outgoing e-mail for a specific Web application by using the Stsadm command-line tool

1. Verify that you have the following administrative credentials: You must be a member of the Administrators group on the local computer.
2. On the drive on which SharePoint Products and Technologies is installed, change to the following directory: %COMMONPROGRAMFILES%\Microsoft shared\Web server extensions\14\Bin.
3. Type the following command, and then press ENTER:

   stsadm -o email

   -outsmtpserver <*SMTP server name*>

   -fromaddress <*valid e-mail address*>

   -replytoaddress <*valid e-mail address*>

   -codepage <*valid code page*>

   [-url <*URL name*>**]**

Example

stsadm -o email -outsmtpserver mail.example.com -fromaddress someone@example.com -replytoaddress someone@example.com -codepage 65001 -url http://server_name

For more information, see Email: Stsadm operation (Office SharePoint Server) (*http://go.microsoft.com/fwlink/?LinkId=150045*).

See Also

Plan outgoing e-mail (SharePoint Server 2010) (*http://technet.microsoft.com/library/75401651-ef01-4348-878e-8a636f0b072d(Office.14).aspx*)

Configure alert settings for a Web application (SharePoint Server 2010) (*http://technet.microsoft.com/library/6f8746e6-9fff-4992-83a9-ae7a6e02eade(Office.14).aspx*)

Configure a mobile account (SharePoint Server 2010)

This article discusses how to configure and manage a mobile account for Microsoft SharePoint Server 2010 to enable users to subscribe to alerts that are sent by using Short Message Service (SMS). The alerts are sent to users' mobile phones when changes are made to a SharePoint list or item.

The mobile alert feature resembles a feature that already exists in SharePoint Server 2010 that enables outgoing e-mail alerts. However, instead of receiving alerts via e-mail when changes are made in a SharePoint list or item, users receive the alerts on their mobile phones. For more information about e-mail alerts, see Configure outgoing e-mail (SharePoint Server 2010).

A SharePoint site is usually located on an intranet. As a result, access to the SharePoint site can be difficult when users are away from the office — for example, when they are traveling or attending a business dinner. The mobile alert feature enables users to react quickly when they receive an SMS alert that an item in a SharePoint list has changed.

You can configure one mobile account for all Web applications in a server farm, or you can configure the mobile account for a specific Web application; however, you can only configure one mobile account in the farm. The scale of your implementation might determine whether you configure the mobile account for the farm or for a specific Web application. If you configure the mobile account for a server farm, everyone in the organization can subscribe to alerts. This is useful, for example, in a small organization in which management wants all users to receive certain alerts. If you have several Web applications that divide your organization into groups, you might want to configure a mobile account for only one of those groups; for example, you want to configure a mobile account to enable everyone in the sales group to subscribe to alerts.

Before you perform these procedures, confirm that:

- The Server farm account has permission to access the Internet for sending alerts.
- You have obtained the root certificate for the service provider's HTTPS Web address. You can obtain the root certificate from your service provider or by using your Web browser.

Procedures in this article:

- Import a root certificate and create a trusted root authority
- Configure a mobile account
- Retrieve mobile account information
- Delete a mobile account

Import a root certificate and create a trusted root authority

Before you configure a mobile account, you must import the root certificate of the service provider's HTTPS Web address, and then create a trusted root authority. This step can only be performed manually by using Windows PowerShell.

To import a root certificate and create a trusted root authority by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin. Additionally, you must be a member of the Farm Administrators group and a member of the local Administrators group on the computer running Windows PowerShell.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. To get the root certificate, at the Windows PowerShell command prompt, type the following command:

   ```
   $cert = Get-PfxCertificate <ObtainedCertificatePath>
   ```

   To create a trusted root authority, at the Windows PowerShell command prompt, type the following command:

   ```
   New-SPTrustedRootAuthority -Name <Name> -Certificate $cert
   ```

   Where:

   - <*Name*> is the name of the trusted root authority that you want to create.
   - <*ObtainedCertificatePath*> is the location of the root certificate file.

   Note:
   In the Certificate window, on the **Certification Path** tab, ensure that you use only the root certificate, and not the other certificates listed. The SharePoint Trusted Root Authorities list should only list root certificates.

For more information, see New-SPTrustedRootAuthority (*http://technet.microsoft.com/library/25458530-4f0d-491c-80d3-61b8f1f0dd7e(Office.14).aspx*).

Configure a mobile account

You can configure a mobile account for a server farm or for a specific Web application, either by using Central Administration or Windows PowerShell.

Note:
If you cannot configure a mobile account, you may have the wrong certificate file. In that case, contact your service provider.

To configure or edit a mobile account for a server farm by using Central Administration

1. Verify that you have the following administrative credentials:
   - To configure a mobile account for a server farm, you must be a member of the Farm Administrators group on the computer that is running the SharePoint Central Administration Web site.
2. On the Central Administration Home page, click **System Settings**.
3. On the System Settings page, in the **E-Mail and Text Messages (SMS)** section, click **Configure mobile account**.
4. On the Mobile Account Settings page, in the **Text Message (SMS) Service Settings** section, click the **Microsoft Office Online** link to access a list of service providers.
5. On the Find an Office 2010 Mobile Service Provider page, in the **Choose your wireless service provider's country/region** list, select the country or region in which your wireless service provider is located.
6. On the Find an Office 2010 Mobile Service Provider page, in the **Choose your current wireless service provider** list, select the wireless service provider that you want to use.

   After you make this selection, you are directed to the Web site of the service provider that you selected. On the Web site, you apply for the SMS service. When you receive the required information from the service provider, return to the Mobile Accounts Settings page.
7. In the **The URL of Text Message (SMS) Service** box, type the URL of the SMS service.

   Note:
   Ensure that the service URL you enter is an HTTPS URL.
8. In the **User Name box** and **Password** box, type the user name and password that you received from the SMS service provider.
9. To confirm that the URL and user credentials are correct, click **Test Service**.
10. Click **OK**.

To configure or edit a mobile account for a server farm by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin. Additionally, you must be a member of the Farm Administrators group and a member of the local Administrators group on the computer running Windows PowerShell.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

```
Set-SPMobileMessagingAccount -Identity sms -WebApplication <WebApplicationUrl> [-
ServiceUrl <ServiceUrl>] [-UserId <UserId>] [-Password <Password>]
```

Where:

- <*WebApplicationUrl*> is the Central Administration URL.
- <*ServiceUrl*> is the URL to server where the SMS service is located.
- <*UserId*> is the user name that you received from the SMS service provider.
- <*Password*> is the user password that you received from the SMS service provider.

Example:

```
Set-SPMobileMessagingAccount -Identity sms -WebApplication http://myserver:8080 -
ServiceUrl https://www.example.com/omsservice.asmx -UserId someone@example.com -
Password password1
```

Or, if you use the pipeline operator, type the following command:

```
Get-SPWebApplication -Identity http://myserver:8080 | Set-SPMobileMessagingAccount
-Identity sms -ServiceUrl https://www.example.com/omsservice.asmx -UserId
someone@example.com -Password password1
```

Note:
Ensure that the service URL you enter is an HTTPS URL.

For more information, see Set-SPMobileMessagingAccount (*http://technet.microsoft.com/library/ca94def6-f55a-4878-bb64-ee6f62373c8f(Office.14).aspx*) and Get-SPWebApplication (*http://technet.microsoft.com/library/11d6521f-f99c-433e-9ab5-7cf9e953457a(Office.14).aspx*).

To configure or edit a mobile account for a Web application by using Central Administration

1. Verify that you have the following administrative credentials:
   - To configure a mobile account for a server farm, you must be a member of the Farm Administrators group on the computer that is running the SharePoint Central Administration Web site.
2. On the Central Administration Home page, in the **Application Management** section, click **Manage web applications**.
3. On the Web Applications page, select the Web application for which you want to configure a mobile account. In **General Settings** on the ribbon, click **Mobile Account**.
4. On the Web Application Text Message (SMS) Service Settings page, in the **Text Message (SMS) Service Settings** section, click the **Microsoft Office Online** link to access a list of service providers.
5. On the Find an Office 2010 Mobile Service Provider page, in the **Choose your wireless service provider's country/region** list, select the country or region in which your wireless service provider is located.
6. On the Find an Office 2010 Mobile Service Provider page, in the **Choose your current wireless service provider** list, select the wireless service provider that you want to use.

After you make this selection, you are directed to the Web site of the service provider that you selected. On the Web site, you apply for the SMS service. When you receive the required information from the service provider, return to the Mobile Accounts Settings page and type in the information.

7. In the **The URL of Text Message (SMS) Service** box, type the URL of the SMS service.

   Note:
   Ensure that the service URL you enter is an HTTPS URL.

8. In the **User Name** box and **Password** box, type the user name and password that you received from the SMS service provider.
9. To confirm that the URL and user credentials are correct, click **Test Service**.
10. Click **OK**.

To configure or edit a mobile account for a Web application by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin. Additionally, you must be a member of the Farm Administrators group and a member of the local Administrators group on the computer running Windows PowerShell.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt (that is, PS C:\>), type the following command:

```
Set-SPMobileMessagingAccount -Identity sms -WebApplication <WebApplicationUrl> [-
ServiceUrl <ServiceUrl>] [-UserId <UserId>] [-Password <Password>]
```

   Where:

   - *<WebApplicationUrl>* is the Web application URL.
   - *<ServiceUrl>* is the URL to server where the SMS service is located.
   - *<UserId>* is the user name that you received from the SMS service provider.
   - *<Password>* is the user password that you received from the SMS service provider.

   Example:

```
Set-SPMobileMessagingAccount -Identity sms -WebApplication http://localhost -
ServiceUrl https://www.example.com/omsservice.asmx -UserId someone@example.com -
Password password1
```

   Or, if you use the pipeline operator, type the following command:

```
Get-SPWebapplication -Identity http://localhost | Set-SPMobileMessagingAccount -
Identity sms -ServiceUrl https://www.example.com/omsservice.asmx -UserId
someone@example.com -Password password1
```

Note:

Ensure that the service URL you enter is an HTTPS URL.

For more information, see Set-SPMobileMessagingAccount (*http://technet.microsoft.com/library/ca94def6-f55a-4878-bb64-ee6f62373c8f(Office.14).aspx*) and Get-SPWebApplication (*http://technet.microsoft.com/library/11d6521f-f99c-433e-9ab5-7cf9e953457a(Office.14).aspx*).

Retrieve mobile account information

You can retrieve mobile account information for a server farm or for a Web application by using Windows PowerShell. You might want to do this to view the mobile account information or, for example, verify that the **set** cmdlet works correctly.

To retrieve mobile account information for a server farm by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin. Additionally, you must be a member of the Farm Administrators group and a member of the local Administrators group on the computer running Windows PowerShell.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

   ```
   Get-SPMobileMessagingAccount -WebApplication <WebApplicationUrl>
   ```

   Where <*WebApplicationUrl*> is the Central Administration URL.

   Example

   ```
   Get-SPMobileMessagingAccount -WebApplication http://myserver
   ```

   Or, if you use the pipeline operator, type the following command:

   ```
   Get-SPWebApplication -Identity http://myserver | Get-SPMobileMessagingAccount -
   AccountType sms
   ```

For more information, see Get-SPMobileMessagingAccount (*http://technet.microsoft.com/library/03b69f50-07ec-4feb-bc9c-567237d200ea(Office.14).aspx*) and Get-SPWebApplication (*http://technet.microsoft.com/library/11d6521f-f99c-433e-9ab5-7cf9e953457a(Office.14).aspx*).

To retrieve mobile account information for a Web application by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin. Additionally, you must be a member of the Farm Administrators group and a member of the local Administrators group on the computer running Windows PowerShell.

2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

   ```
   Get-SPMobileMessagingAccount -WebApplication <WebApplicationUrl>
   ```

   Where <*WebApplicationUrl*> is the Web application URL.

   Example

   ```
   Get-SPMobileMessagingAccount -WebApplication http://localhost
   ```

   Or, if you use the pipeline operator, type the following command:

   ```
   Get-SPWebApplication -Identity http://localhost | Get-SPMobileMessagingAccount -
   AccountType sms
   ```

For more information, see Get-SPMobileMessagingAccount (*http://technet.microsoft.com/library/03b69f50-07ec-4feb-bc9c-567237d200ea(Office.14).aspx*) and Get-SPWebApplication (*http://technet.microsoft.com/library/11d6521f-f99c-433e-9ab5-7cf9e953457a(Office.14).aspx*).

Delete a mobile account

You can delete a mobile account for a server farm or for a Web application. This makes the account unavailable so users can no longer subscribe to SMS alerts, but it does not delete the account that you set up with the service provider. You might want to delete a mobile account if, for example, the organization decides that there is no business value in sending out SMS alerts to users.

Note:
There is no equivalent Windows PowerShell functionality.

To delete a mobile account for a server farm

1. Verify that you have the following administrative credentials:
   - To delete a mobile account for a server farm, you must be a member of the Farm Administrators group on the computer that is running the SharePoint Central Administration Web site.
2. On the Central Administration Home page, click **System Settings**.
3. On the System Settings page, in the **E-mail and Text Messages (SMS)** section, click **Configure mobile account**.
4. On the Mobile Account Settings page, clear entries from all the boxes, and then click **OK**.

To delete a mobile account for a Web application

1. Verify that you have the following administrative credentials:

- To delete a mobile account for a server farm, you must be a member of the Farm Administrators group on the computer that is running the SharePoint Central Administration Web site.

2. On the Central Administration Home page, in the **Application Management** section, click **Manage web applications**.
3. In **General Settings** on the ribbon, click **Mobile Account**.
4. On the Web application Text Message (SMS) Service Settings page, delete entries from all the boxes, and then click **OK**.

See Also

Configure outgoing e-mail (SharePoint Server 2010)

Install and configure Remote BLOB Storage (RBS) with the FILESTREAM provider (SharePoint Server 2010)

This article describes how to install and configure Remote BLOB Storage (RBS) with the FILESTREAM provider on a Microsoft SQL Server 2008 database server that supports a Microsoft SharePoint Server 2010 system. RBS is typically recommended in the case where the content databases are 4 gigabytes (GB) or larger.

RBS is a library API set that is incorporated as an add-on feature pack for Microsoft SQL Server 2008 and Microsoft SQL Server 2008 Express. RBS is designed to move the storage of binary large objects (BLOBs) from database servers to commodity storage solutions. RBS ships with the RBS FILESTREAM provider, which uses the RBS APIs to store BLOBs. Before installing and implementing RBS, we highly recommend that you read the articles Plan for Remote BLOB Storage (RBS) (SharePoint Server 2010) (*http://technet.microsoft.com/library/c1f83b4f-a507-42f7-bd82-fed5404ed1ad(Office.14).aspx*) and Overview of Remote BLOB Storage (SharePoint Server 2010) (*http://technet.microsoft.com/library/d359cdaa-0ebd-4c59-8fc5-002cba241b18(Office.14).aspx*).

If you want to implement RBS with a provider other than FILESTREAM, read the article Install and configure Remote BLOB Storage (RBS) without the FILESTREAM provider (SharePoint Server 2010) (*http://technet.microsoft.com/library/7c877590-c6b4-48b2-aee3-330c5d42d44c(Office.14).aspx*).

In this article:

- Enable FILESTREAM and provision the RBS data store
- Install RBS
- Enable and test RBS

The user account that you use to perform the procedures that are described in this article must have the following memberships:

- Administrators group on the Web servers and application servers.
- Farm Administrators group for the SharePoint Server 2010 farm.
- SQL Server **dbcreator** and **securityadmin** fixed server roles on the computer that is running SQL Server 2008 Express or SQL Server 2008.

Enable FILESTREAM and provision the RBS data store

You must enable and configure FILESTREAM on the computer that is running SQL Server 2008 that hosts the SharePoint Server 2010 databases.

To enable and configure FILESTREAM, follow the instructions in How to: Enable FILESTREAM (*http://go.microsoft.com/fwlink/?LinkID=166110&clcid=0x409*). After you have enabled and configured FILESTREAM, provision a BLOB store as described in the following procedure.

To provision a BLOB Store with the FILESTREAM provider

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Management Studio**.
2. Expand **Databases**.
3. Select the content database for which you want to create a BLOB store, and then click **New Query**.
4. In the Query pane, copy and execute each of the following SQL queries in the sequence provided here.

   Tip:
   For best performance, simplified troubleshooting, and as a general best practice, we recommend that you create the BLOB store on a volume that does not contain the operating system, paging files, database data, log files, or the tempdb file.

```
use [ContentDbName]

if not exists (select * from sys.symmetric_keys where name =
N'##MS_DatabaseMasterKey##')create master key encryption by password = N'Admin Key
Password !2#4'

use [ContentDbName]

if not exists (select groupname from sysfilegroups where
groupname=N'RBSFilestreamProvider')alter database [ContentDbName]

 add filegroup RBSFilestreamProvider contains filestream

use [ContentDbName]

alter database [ContentDbName] add file (name = RBSFilestreamFile, filename =
'c:\Blobstore') to filegroup RBSFilestreamProvider
```

   Where *[ContentDbName]* is the content database name, and *c:\BLOBStore* is the volume\directory in which you want the BLOB store created. Be aware that you can provision a BLOB store only one time. If you attempt to provision the same BLOB store more than one time, you will receive an error.

Install RBS

You must install RBS on the database server and on all Web servers and application servers in the SharePoint farm. You must configure RBS separately for each associated content database.

Warning:
We do not recommend that you install RBS by running the RBS_X64.msi file and launching the Install SQL Remote BLOB Storage wizard. The wizard configures the RBS Maintainer to run a scheduled task every 30 days. This setting might not be optimal for your environment. For more information about the RBS Maintainer, see the SQL Server Help documentation that is included with the SQL Server Remote BLOB Store installation package from the Feature Pack for Microsoft SQL Server 2008 R2.

To install RBS on a Web server

1. On any Web server, go to http://go.microsoft.com/fwlink/?LinkID=177388 (*http://go.microsoft.com/fwlink/?LinkID=177388*) to download the RBS_X64.msi file.

   Important:
   You must install the version of RBS that is included in the SQL Server Remote BLOB Store installation package from the Feature Pack for Microsoft SQL Server 2008 R2. The version of RBS must be **10.50.xxx**. No earlier version of RBS is supported for SharePoint Server 2010.

2. Click **Start** and then type **cmd** in the text box. In the list of results, right-click **cmd**, and then click **Run as administrator**. Click **OK**.
3. Copy and paste the following command at the command prompt:

   ```
   msiexec /qn /lvx* rbs_install_log.txt /i RBS_X64.msi TRUSTSERVERCERTIFICATE=true
   FILEGROUP=PRIMARY DBNAME="<ContentDbName>" DBINSTANCE="<DBInstanceName>"
   FILESTREAMFILEGROUP=RBSFilestreamProvider FILESTREAMSTORENAME=FilestreamProvider_1
   ```

   Where:

   - <*ContentDbName*> is the database name.
   - <*DBInstanceName*> is the SQL Server instance name.

   The operation should complete in approximately one minute.

To install RBS on all additional Web and application servers

1. On a Web server, go to http://go.microsoft.com/fwlink/?LinkID=177388 (*http://go.microsoft.com/fwlink/?LinkID=177388*) and download the RBS_X64.msi file.

   Important:
   You must install the version of RBS that is included in the SQL Server Remote BLOB Store installation package from the SQL Server Remote BLOB Store installation package from the Feature Pack for SQL Server 2008 R2. The version of RBS must be 10.50.xxx. No earlier version of RBS is supported for SharePoint Server 2010.

2. Click **Start** and then type **cmd** in the text box. In the list of results, right-click **cmd**, and then click **Run as administrator**. Click **OK**.
3. Copy and paste the following command at the command prompt:

```
msiexec /qn /lvx* rbs_install_log.txt /i RBS_X64.msi DBNAME="ContentDbName"
DBINSTANCE="DBInstanceName"
ADDLOCAL="Client,Docs,Maintainer,ServerScript,FilestreamClient,FilestreamServer"
```

   Where:

   - *ContentDbName* is the database name
   - *DBInstanceName* is the name of the SQL Server instance.

   The operation should finish within approximately one minute.
4. Repeat this procedure on all Web servers and application servers. If you do not install RBS on every Web and application server, users will encounter errors when they try to write to the content databases.

To confirm the RBS installation

1. The rbs_install_log.txt log file is created in the same location as the RBS_X64.msi file. Open the rbs_install_log.txt log file with a text editor and scroll toward the bottom of the file. Within the last 20 lines of the end of the file, an entry should read as follows: “Product: SQL Remote Blob Storage – Installation completed successfully”.
2. On the computer that is running SQL Server 2008, verify that the RBS tables were created in the content database. Several tables should reside under the content database with names that are preceded by the letters "mssqlrbs".

Enable and test RBS

You must enable RBS on one Web server in the SharePoint farm. It does not matter which Web server you choose for this activity, as long as RBS was installed on it by using the previous procedure.

To enable RBS

1. On the **Start** menu, click **Programs**, click **Microsoft SharePoint 2010 Products**, and then click **SharePoint 2010 Management Shell**.
2. At the Windows PowerShell command prompt, type each of the following commands.

```
$cdb = Get-SPContentDatabase –WebApplication <http://SiteName>
```

   Where <*http://SiteName*> is the URL of the Web application that is connected to the content database.

```
$rbss = $cdb.RemoteBlobStorageSettings
$rbss.Installed()
```

```
$rbss.Enable()

$rbss.SetActiveProviderName($rbss.GetProviderNames()[0])

$rbss
```

To test the RBS data store

1. Connect to a document library on any Web server.
2. Upload a file that is at least 100 kilobytes (KB) to the document library.
3. On the computer that contains the RBS data store, click **Start**, and then click **Computer**.
4. Browse to the RBS data store directory.
5. Browse to the file list and open the folder that has the most recent modified date (other than $FSLOG). In that folder, open the file that has the most recent modified date. Verify that this file has the same size and contents as the file that you uploaded. If it does not, ensure that RBS is installed and enabled correctly.

To enable additional databases to use RBS, see Set a content database to use Remote BLOB Storage (RBS) (SharePoint Server 2010) (*http://technet.microsoft.com/library/83ee2a38-1b91-4c6a-83de-1e968db74611(Office.14).aspx*).

See Also

Migrate content into or out of Remote BLOB Storage (RBS) (SharePoint Server 2010) (*http://technet.microsoft.com/library/0317c126-78ed-47e0-9109-fe143ddb0259(Office.14).aspx*)

Disable Remote BLOB Storage (RBS) on a content database (SharePoint Server 2010) (*http://technet.microsoft.com/library/75096f60-b94e-44c2-bc84-8aa3e2c4fff3(Office.14).aspx*)

Configure and use the Exchange connector (SharePoint Server 2010)

This article describes how to create a crawl rule and add a content source for crawling Microsoft Exchange Server public folders. Before you begin, ensure that the crawler has at least Read permission to the public folder.

Create a crawl rule

This section describes the steps that are required to create a crawl rule. You must create a crawl rule if the default content access account does not have Read permission to the Exchange public folders that you want to crawl.

To create a crawl rule

1. Verify that the user account that is performing this procedure is an administrator for the Search service application.
2. On the Home page of the SharePoint Central Administration Web site, in the **Application Management** section, click **Manage service applications**.
3. On the Manage Service Applications page, click **Search Service Application**.
4. On the Search Administration page, in the **Crawling** section, click **Crawl Rules**.
5. On the Manage Crawl Rules page, click **New Crawl Rule**.
6. In the **Path** section, in the **Path** box, type a path to be crawled.

   When creating a crawl rule, the URL that you type in the **Path** box should be in the following form: <*protocol*>://*hostname*/*

   where <*protocol*> is the protocol that you want to use (typically http or https), and *hostname* is the NetBIOS or fully qualified domain name of the server that is running Exchange Server.
7. In the **Crawl Configuration** section, select one of the following:
   - Exclude all items in this path (Default)
   - Include all items in this path
8. In the **Specify Authentication** section, select the type of crawl authentication to use. This section is used only when **Include all items in this path** is selected.
9. Click **OK** to add the crawl rule.

Add a content source for Microsoft Exchange public folders

Use the following procedures to create a content source for Exchange Server public folders.

Note that there are two procedures listed that describe separate processes to add a content source based on the Exchange Server version. The first procedure describes how to add a content source for Exchange Server 2007 and earlier, the second describes how to add a content source from Exchange Server 2007 with Service Pack 2 (SP2) and later versions. This includes Microsoft Exchange Server 2010.

To add a content source for Exchange Server 2007 and earlier public folders

1. Verify that the user account that is performing this procedure is an administrator for the Search service application.
2. On the SharePoint Central Administration Home page, in the **Application Management** section, click **Manage service applications**.
3. On the Manage Service Applications page, click **Search Service Application**.
4. On the Search Administration page, in the **Crawling** section, click **Content Sources**.
5. On the Manage Content Sources page, click **New Content Source**.
6. On the Add Content Source page, in the **Name** box, type the name that you want to use for the content source.
7. In the **Content Source Type** section, select **Exchange Public Folders**.
8. In the **Start Addresses** section, type the URLs for the Exchange Server public folders that you want to crawl. These URLs are typically in one of the following forms:
   - *<protocol>*://*host name*/public

     where *<protocol>* can be http or https, and *host name* is the NetBIOS or fully qualified domain name (FQDN) of the server that is running Exchange Server.
   - *<protocol>*://*host name*/public/*subfolder*

     where *<protocol>* can be http or https, *host name* is the NetBIOS or FQDN of the server that is running Exchange Server, and *subfolder* is the name of the specific subfolder that you want to crawl.

   For example, if you want to crawl all subfolders in the public folder on a server that is named exch-01 and that is in the Contoso domain, and that server does not use SSL, you could type either **http://exch-01/public** or **http://exch-01.contoso.com**. To crawl only a specific subfolder named Bob in the same public folder, type **http://exch-01/public/bob** or **http://exch-01.contoso.com/bob**.

Note:

For performance reasons, you cannot add the same start addresses to multiple content sources.

9. In the **Crawl Settings** section, select the behavior for the kind of content that you selected.
10. In the **Crawl Schedules** section, you can optionally specify when to start full and incremental crawls:
    - You can create a full crawl schedule by clicking the **Create Schedule** link below the **Full Crawl** list.
    - You can create an incremental crawl schedule by clicking the **Create schedule** link below the **Incremental Crawl** list.
11. Click **OK**.
12. Repeat steps 4 through 10 for any additional content sources that you want to create.

To add content sources for Microsoft Exchange 2007 SP2 and later versions public folders

1. Verify that the user account that is performing this procedure is an administrator for the Search service application.
2. Open a Web browser and navigate to the Outlook Web Access Web page for the Exchange Server that has the public folders that you want to crawl.
3. Log on to Outlook Web Access using any user account that has Read access to the public folders that you want to crawl.
4. Navigate to the public folder that you want to crawl, right-click that folder, and then select **Open in New Window**.
5. When the new window opens, navigate to the new window's address bar and copy the complete URL. This is the Outlook Web Access public folder address.
6. On the SharePoint Central Administration Web site, in the Application Management section, click **Manage service applications**.
7. On the Manage Service Applications page, click **Search Service Application**.
8. On the Search Administration page, in **Crawling** section, click **Content Sources**.
9. Click **New Content Source**.
10. On the Add Content Source page, in the **Name** box, type the name that you want to use for the content source.
11. In the **Content Source Type** section, select **Exchange Public Folders**.
12. In the **Start Addresses** section, paste the Outlook Web Access public folder address that you copied in step 5.
13. In the **Crawl Settings** section, select the behavior for the kind of content that you selected.
14. In the **Crawl Schedules** section, you can optionally specify when to start full and incremental crawls:

- You can create a full crawl schedule by clicking the **Create Schedule** link below the **Full Crawl** list.
- You can create an incremental crawl schedule by clicking the **Create schedule** link below the **Incremental Crawl** list.

15. To create the content source, click **OK**.

Deploy Microsoft Business Intelligence Indexing Connector (SharePoint Server 2010)

This section describes deployment steps for Microsoft Business Intelligence Indexing Connector.

- Install and configure Business Intelligence Indexing Connector (back end)

  This article describes how to install and configure the Microsoft Business Intelligence Indexing Connector for the back end after you have successfully installed Microsoft FAST Search Server 2010 for SharePoint as a back-end.

- Install or uninstall SharePoint Solution Pack for Business Intelligence Indexing Connector (front end)

  This article describes how to install and configure Microsoft Business Intelligence Indexing Connector.

- Install or uninstall SharePoint Solution Pack language pack for Business Intelligence Indexing Connector (front end)

  These instructions describe how to apply a language pack for the installation of SharePoint Solution Pack.

Install and configure Business Intelligence Indexing Connector (back end)

This article describes how to install and configure the Microsoft Business Intelligence Indexing Connector for the back end. To download the Business Intelligence Indexing Connector (back end), see Microsoft Business Intelligence Indexing Connector – Back end (*http://go.microsoft.com/fwlink/?LinkId=195026*).

Microsoft FAST Search Server 2010 for SharePoint must be installed first to provide indexing and query matching capabilities, and the administration of these features. For more information, see Install Microsoft SharePoint Server 2010 and Microsoft SQL Server 2008 (FAST Search Server 2010 for SharePoint) (*http://technet.microsoft.com/library/7d391df2-3f55-4084-974e-43eaf66fc12a(Office.14).aspx*).

Important:
Microsoft SharePoint Server 2010 and FAST Search Server 2010 for SharePoint must **not** be installed on the same server.

Before you start

Make sure that you have performed the steps for Install FAST Search Server 2010 for SharePoint (*http://technet.microsoft.com/library/224430a1-4c94-451e-9a17-c8f53ac4865b(Office.14).aspx*).

Review Determine software requirements for Business Intelligence Indexing Connector (*http://technet.microsoft.com/library/b2d2b716-10d3-409f-8887-f047ae92e6c8(Office.14).aspx*) and Overview of Business Intelligence Indexing Connector architecture (*http://technet.microsoft.com/library/6b768a7a-a305-4947-b0fc-69fed70ef2ee(Office.14).aspx*).

We recommend that you grant the Service Identity account, the account under which Business Intelligence Indexing Connector service is run, the following fixed database roles:

- **db_datareader**
- **db_denydatawriter**

The Service Identity account should be added to the **FASTSearchAdministrators** group so that business intelligence properties are recognized on a document.

Considerations for crawling external data sources

This section explains how you can successfully crawl external data sources, specifically Microsoft SQL Server Reporting Services (SSRS) report server, Microsoft SQL Server 2008 R2 PowerPivot for Microsoft SharePoint 2010, and other files.

To crawl files that are created by Microsoft SQL Server PowerPivot for Microsoft Excel, you must deploy the same version of PowerPivot for Excel onto the instance of Microsoft FAST Search Server

2010 for SharePoint that is installed on the client. The client files that are crawled must be Microsoft Excel 2010 files that have the same version of PowerPivot for Excel that is installed on FAST Search Server 2010 for SharePoint. To crawl SSRS .rdl files on a computer that is running Microsoft SharePoint Server with the Reporting Services Add-in, you must append `NoGetRedirect="TRUE"` so that the tag in %ProgramFiles%\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\XML\ServerFiles_ReportServer.xml resembles the following.

```
<Mapping FileExtension="rdl"
RedirectUrlTemplate="/_layouts/ReportServer/RSViewerPage.aspx?rv:RelativeReportUrl=|0"
NoGetRedirect="TRUE"/>
```

If you have multiple SharePoint Web applications in a farm configuration, you must change the ServerFiles_ReportServer.xml file on each computer that has a front-end Web and is configured to manage report server content.

Note:
If the Business Intelligence Indexing Connector front end (SharePoint Server 2010) and back end (FAST Search Server 2010 for SharePoint) are installed on the same computer, you do not have to render the SSRS report in the browser. Clicking an .rdl document (SSRS report) in a PowerPivot Gallery gives you the option to download the .rdl document rather than rendering the report in the browser. This does not affect other SharePoint document libraries, which will render SSRS reports in the browser.

To query dynamic data in SSRS reports, you must specify the HTTP redirect as the URL after you select **Enable External Data Source Crawling**. The redirect would resemble the following.

```
http://<computername>/_vti_bin/reportserver
```

To crawl SSRS report files, you must have the Business Intelligence Indexing Connector installed on the admin server. By default, all servers in a multiple server deployment, except the admin server, are non-admin servers. To learn how to determine whether a configured admin server is running, see Install FAST Search Server 2010 for SharePoint (*http://technet.microsoft.com/library/224430a1-4c94-451e-9a17-c8f53ac4865b(Office.14).aspx*).

Install the Business Intelligence Indexing Connector

When you click **Download** on the download page, a Microsoft Installer Package (MSI) runs on your computer and a wizard opens. Follow the procedures in sequence.

To install the Business Intelligence Indexing Connector

1. A welcome page appears. Click **Next** to continue with installation.
2. Browse to change the default destination folder location. The default install location is %Program Files%\Microsoft\Business Intelligence Indexing Connector. Click **Next** and then click **Install**.
3. Click **Finish**.

To configure the Business Intelligence Indexing Connector

1. On the **Start Menu**, point to **All Programs**, and then locate FAST Search Server 2010 for SharePoint. Select **Business Intelligence Indexing Connector** to start the wizard. A welcome page opens. Click **Next**
2. On the BI Indexing Connector Service Settings page, select service identity account settings and optionally change the port. Click **Next**.

   Note:
   To crawl external data sources, such as SSRS report server, you must enter an account that has database access.
3. On the BI Indexing Connector Filters Settings page, you have to option to enable external data source crawling. You can also select an SSRS report server as an external data source.

   If you select to **Enable External Data Source Crawling** for Excel or SSRS, you must apply security restrictions to indexed information about Active Directory service accounts. Security restrictions give you the option to select one or more user accounts and one or more groups.

   Click **Next**.

   Warning:
   If you select **Enable External Data Source Crawling**, you may enable search users to breach the network security because of design limitations. If your file has specific data-level permissions lower than the file security, consider excluding that specific report from the crawl completely. You can do this by isolating it in a secure content source or by not giving access to the report—or underlying data—from the Service Identity account.

   Warning:
   Users with write access to external data sources, including the Master database, can potentially index the complete database schema and execute SQL scripts. As a best practice, the database administrator should only grant read access to the Service Identity account.
4. Click **Configure** to complete the configuration of your choices.

Changes on the configuration made through the configuration wizard may stop and restart the FAST Search Server 2010 for SharePoint service after configuration has completed successfully. If the service stops after configuration or cannot restart, refer to Troubleshooting Business Intelligence Indexing Connector.

Note:
If you reinstall Office Filter Pack after the Business Intelligence Indexing Connector is configured, you must re-run the configuration wizard.

Verification of configuration

When configuration is completed successfully, the BI Indexing Connector service is set to start automatically.

See Also

Determine software requirements for Business Intelligence Indexing Connector (*http://technet.microsoft.com/library/b2d2b716-10d3-409f-8887-f047ae92e6c8(Office.14).aspx*)

Overview of Business Intelligence Indexing Connector architecture (*http://technet.microsoft.com/library/6b768a7a-a305-4947-b0fc-69fed70ef2ee(Office.14).aspx*)

Uninstall Business Intelligence Indexing Connector

This article describes how to uninstall the Microsoft Business Intelligence Indexing Connector.

Note:
When you uninstall the Business Intelligence Indexing Connector, the related managed and crawled properties are not removed from the Microsoft FAST Search Server 2010 for SharePoint back-end installation. You can remove these properties by using the SharePoint Central Administration Web site. Removing the managed and crawled properties manually requires you to re-index your content.

To uninstall the Business Intelligence Indexing Connector

1. Open the **Uninstall or change a program** control panel item.
2. Select Business Intelligence Indexing Connector.
3. Click **Uninstall**.

Install or uninstall SharePoint Solution Pack for Business Intelligence Indexing Connector (front end)

This article describes how to install and configure Microsoft Business Intelligence Indexing Connector (front end). To download Business Intelligence Indexing Connector (front end), see Microsoft Business Intelligence Indexing Connector – Front end (*http://go.microsoft.com/fwlink/?LinkId=195027*). This article does not show how to install Microsoft SQL Server or Microsoft SharePoint Server 2010.

Before you start

Before you start deployment, do the following.

- Review the topology for back-end and front-end deployments of Business Intelligence Indexing Connector. See Overview of Business Intelligence Indexing Connector architecture (*http://technet.microsoft.com/library/6b768a7a-a305-4947-b0fc-69fed70ef2ee(Office.14).aspx*).
- Make sure the following are ready as part of your SharePoint Server 2010 setup.
  - FAST Search Server 2010 for SharePoint is configured as a back-end for Microsoft SharePoint Server 2010. For more information, see Deploy FAST Search Server 2010 for SharePoint (*http://go.microsoft.com/fwlink/?LinkId=171127*).
  - A FAST Search site collection using the Enterprise FAST Search template exists. You will use the URL later in the process in this section. For more information about how to create a site collection, see Create a site collection (SharePoint Server 2010).
- Verify that you have necessary administrative credentials. To enable Business Intelligence Indexing Connector (front end), you must be a member of the farm administrators group.

SharePoint Solution Pack for SharePoint Server

By installing the SharePoint Solution Pack, you add a **Reports** tab to the front-end FAST Search site that enables users to use the Business Intelligence Indexing Connector.

To install SharePoint Solution Pack

1. At a command prompt, type the following command to extract the contents of BIConnectorFrontend.exe. `BIConnectorFrontend.exe /extract:<folder location>`

   Example: `BIConnectorFrontend.exe /extract:C:\BIConnectorFrontendFolder`
2. Open a Command Prompt window and navigate to where Stsadm.exe is located: %ProgramFile%\Common Files\Microsoft Shared\Web Server Extensions\14\Bin.
3. Type the following stsadm command and make sure that you indicate the folder location of the

BIConnectorFrontend.wsp file. This file was extracted in step 1.

```
stsadm -o addsolution -filename <folder location>\FullPack\BIConnectorFrontend.wsp
```

Press ENTER.

4. After the operation completes successfully, type the following Stsadm command.

   ```
   stsadm -o deploysolution -name BIConnectorFrontend.wsp -local -allowgacdeployment
   ```

   Press ENTER.

5. After the operation completes successfully, type the following Stsadm command and make sure that you indicate the URL of the FAST Search site collection that you created.

   ```
   stsadm -o activatefeature -name BIIndexingConnector -url [Website URL]
   ```

   Press ENTER.

6.

   Note:
   Before testing the **Reports** tab in your FAST Search site, wait 60 seconds for Reports search scope to be deployed. Otherwise, you will receive the error: "The scope selected is not valid".

   Go to the FAST Search site to verify that a **Reports** tab was added.

   Note:
   Only a single SharePoint Solution Pack from any language can be installed.

Uninstall SharePoint Solution Pack

This section shows how to uninstall the SharePoint Solution Pack for SharePoint Server.

Note:
All language packs must be uninstalled before you continue with the uninstallation of the SharePoint Solution Pack.

To uninstall SharePoint Solution Pack

1. Open a Command Prompt window and navigate to where Stsadm.exe is located: %ProgramFile%\Common Files\Microsoft Shared\Web Server Extensions\14\Bin.
2. Type the following Stsadm command and make sure that you indicate the URL of the FAST Search site collection that you used in the SharePoint Solution Pack installation.

   ```
   stsadm -o deactivatefeature -name BIIndexingConnector -url <Website URL>
   ```

   Press ENTER.

3. After the operation completes successfully, type the following Stsadm command.

   ```
   stsadm -o retractsolution -name BIConnectorFrontend.wsp -local
   ```

   Press ENTER.

4. After the operation completes successfully, type the following Stsadm command and press ENTER.

```
stsadm -o deletesolution -name BIConnectorFrontend.wsp
```

See Also

Determine software requirements for Business Intelligence Indexing Connector (*http://technet.microsoft.com/library/b2d2b716-10d3-409f-8887-f047ae92e6c8(Office.14).aspx*)

Overview of Business Intelligence Indexing Connector architecture (*http://technet.microsoft.com/library/6b768a7a-a305-4947-b0fc-69fed70ef2ee(Office.14).aspx*)

Install or uninstall SharePoint Solution Pack language pack for Business Intelligence Indexing Connector (front end)

The following instructions show how to apply a language pack for the installation of SharePoint Solution Pack for Microsoft Business Intelligence Indexing Connector (front end). To download Business Intelligence Indexing Connector (front end), see Microsoft Business Intelligence Indexing Connector – Front end (*http://go.microsoft.com/fwlink/?LinkID=195027*).

SharePoint Solution Pack language pack

By installing the SharePoint Solution Pack with a language pack, you add a **Reports** tab to the front-end FAST Search site that enables international audiences to use Business Intelligence Indexing Connector.

To install SharePoint Solution Pack language pack

1. Create a localized FAST Search site collection using the Enterprise FAST Search template. You will use the URL later. For more information about how to create a site collection, see Create a site collection (SharePoint Server 2010).
2. Open a Command Prompt window and navigate to where you installed BIConnectorFrontend.exe.
3. Extract the contents of BIConnectorFrontend.exe by typing the following command.

   ```
   BIConnectorFrontend.exe /extract:<folder location>
   ```
4. Open a Command Prompt window and navigate to where Stsadm.exe is located at %ProgramFiles%\Common Files\Microsoft Shared\Web Server Extensions\14\Bin.
5. Type the following Stsadm command and make sure that you indicate the folder location of the BIConnectorFrontend.wsp file. This file is extracted in step 1. Also indicate the LCID code of the language that you want to install.

   ```
   stsadm -o addsolution -filename <folder location>\LanguagePack\BIConnectorFrontend.wsp -lcid <code>
   ```

   Press ENTER.
6. After the operation is completed successfully, type the following Stsadm command and make sure that you indicate the LCID code of the language that you want to install.

   ```
   stsadm -o deploysolution -name BIConnectorFrontend.wsp -local -allowgacdeployment -lcid <code>
   ```

   Press ENTER.

7. After the operation is completed successfully, type the following Stsadm command and make sure that you indicate the URL of the FAST Search site collection that you created.

   ```
   stsadm -o activatefeature -name BIIndexingConnector -url <Website URL>
   ```

   Press ENTER.

   Note:
   Any number of language packs can be deployed with each SharePoint Solution Pack.

Uninstall SharePoint Solution Pack Language Pack

This section shows how to uninstall the SharePoint Solution Pack language pack.

Note:
All language packs must be uninstalled before you continue with the uninstallation of the SharePoint Solution Pack.

To uninstall SharePoint Solution Pack for SharePoint Server Language Pack

1. Open a Command Prompt window and navigate to where Stsadm.exe is located, %ProgramFile%\Common Files\Microsoft Shared\Web Server Extensions\14\Bin.
2. Type the following Stsadm command and make sure that you indicate the URL of the FAST Search site collection that you used in your Language Pack installation.

   ```
   stsadm -o deactivatefeature -name BIIndexingConnector -url <Website URL>
   ```

   Press ENTER.
3. After the operation is completed successfully, type the following Stsadm command and make sure that you indicate the LCID code of the language that you want to un-install.

   ```
   stsadm -o retractsolution -name BIConnectorFrontend.wsp -local -lcid <code>
   ```

   Press ENTER.
4. After the operation is completed successfully, type the following Stsadm command and make sure that you indicate the LCID code of the language that you want to uninstall.

   ```
   stsadm -o deletesolution -name BIConnectorFrontend.wsp -lcid <code>
   ```

See Also

Determine software requirements for Business Intelligence Indexing Connector (*http://technet.microsoft.com/library/b2d2b716-10d3-409f-8887-f047ae92e6c8(Office.14).aspx*)

Overview of Business Intelligence Indexing Connector architecture (*http://technet.microsoft.com/library/6b768a7a-a305-4947-b0fc-69fed70ef2ee(Office.14).aspx*)

Troubleshooting Business Intelligence Indexing Connector

This article describes considerations when you install, configure, and crawl data sources for business intelligence assets by using Microsoft Business Intelligence Indexing Connector.

Data sources

Business Intelligence Indexing Connector crawls files and can optionally crawl external data sources to data-driven files such as an Microsoft Excel file connected to a Microsoft SQL Server source or a Microsoft SQL Server Reporting Services (SSRS) .rdl file connected to a SQL Server database. This is so that you can see the database name of the data source and also elements of the data, including dimensions and measures of a Microsoft SQL Server Analysis Services (SSAS) cube. The following are considerations and issues that you may encounter when Business Intelligence Indexing Connector crawls Excel and SSRS .rdl files.

Excel and PowerPivot files

Crawling Excel files with custom SQL statements is not supported. If a data source for an Excel file has a custom SQL statement for the data source query, you will receive an error message in an enabled event log that the SQL query cannot be read.

To crawl files for Microsoft SQL Server PowerPivot for Microsoft Excel, you must deploy PowerPivot for Excel 2010 on the instance of FAST Search Server 2010 for SharePoint that is installed on the client. The client files that are crawled must be Excel 2010 files with the same version of PowerPivot for Excel that is installed on FAST Search Server 2010 for SharePoint.

Unified Logging Service log file or trace logs

Any changes on the configuration made through the configuration wizard will stop and restart the service after configuration has completed successfully. If the service stops or cannot restart after configuration, review the Unified Logging Service (ULS) log. The ULS writes Microsoft SharePoint Server 2010 events to SharePoint Server 2010 Trace Logs which are located at:

```
%ProgramFile%\Common Files\Microsoft Shared\web server extensions\14\LOGS.
```

Collectively this location and its files are commonly referred to as the “ULS Logs” though MSDN calls them the Trace Logs. The ulscommoncore.msi file is installed when FAST Search Server 2010 for SharePoint is installed. If it has not been installed, run ulscommoncore.msi from %FastSetupFolder%\installer\files to enable the Business Intelligence Indexing Connector log file.

Configure services (SharePoint Server 2010)

In Microsoft SharePoint Server 2010, individual services can be configured independently, and you can implement only the services that your organization needs. For information about how to start, stop, and configure services, see Manage services on the server (SharePoint Server 2010) (*http://technet.microsoft.com/library/6641730b-7099-47c0-938f-783ea8ef8e62(Office.14).aspx*)

Services that are deployed are named *service applications*. A service application provides a resource that can be shared across sites within a farm or sometimes across multiple farms, and can be accessed by users through a hosting Web application. Service applications are associated to Web applications by *service application connections*. For more information about managing and publishing service applications, see Service application and service management (SharePoint Server 2010).

For more information about service applications and services, see Technical diagrams (SharePoint Server 2010) (*http://technet.microsoft.com/library/bcbae7bd-656b-4003-969c-8411b81fcd77(Office.14).aspx*)

Note:
If you plan to use Office Web Apps, you must install and configure them to work with SharePoint 2010 Products. For more information, see Office Web Apps (Installed on SharePoint 2010 Products) (*http://technet.microsoft.com/library/8a58e6c2-9a0e-4355-ae41-4df25e5e6eee(Office.14).aspx*)

This section contains the following articles:

- Manage service applications (SharePoint Server 2010) (*http://technet.microsoft.com/library/0a785758-694c-4c5c-abe6-70a17b490c6e(Office.14).aspx*)

  This article discusses the structures of service applications and services in Microsoft SharePoint Server 2010, and explains how service applications and services can be managed.

- Configure the security token service (SharePoint Server 2010)

  This article provides guidance to help you to configure the Microsoft SharePoint Server 2010 security token service (STS). An STS is a specialized Web service that is designed to respond to requests for security tokens and provide identity management.

- Configure the Secure Store Service (SharePoint Server 2010)

  This article describes the Microsoft SharePoint Server 2010 Secure Store Service operations that solution designers can use to create target applications that map user and group credentials to the credentials of external data sources.

- Post-installation steps for search (SharePoint Server 2010)

  This article describes post-installation steps for configuring the search system in a single-server deployment.

Service application and service management (SharePoint Server 2010)

Articles in this section discuss the structures of service applications and services in Microsoft SharePoint Server 2010, and explain how service applications and services can be managed. These articles are for farm administrators and service application administrators who will operate SharePoint Server 2010.

In This Section

- About service applications and services (SharePoint Server 2010) (*http://technet.microsoft.com/library/ba00a200-ea96-45f6-a423-3cfcffba77f8(Office.14).aspx*)

  Provides an introduction to the logical infrastructure of service applications and services.
- Manage service applications (SharePoint Server 2010) (*http://technet.microsoft.com/library/0a785758-694c-4c5c-abe6-70a17b490c6e(Office.14*).aspx

  Describes how to manage, create, configure, and share service applications.
- Manage services on the server (SharePoint Server 2010) (*http://technet.microsoft.com/library/6641730b-7099-47c0-938f-783ea8ef8e62(Office.14).aspx*)

  Describes how services can be started, stopped, and configured.

See Also

Web applications management (SharePoint Server 2010) (*http://technet.microsoft.com/library/1ad49d4e-598d-446c-aabd-de4a8f6ad391(Office.14).aspx*)

Technical diagrams (SharePoint Server 2010) (*http://technet.microsoft.com/library/bcbae7bd-656b-4003-969c-8411b81fcd77(Office.14).aspx*)

Configure the security token service (SharePoint Server 2010)

This article provides guidance to enable you to configure the Microsoft SharePoint Server 2010 security token service (STS). An STS is a specialized Web service that is designed to respond to requests for security tokens and provide identity management. The core functionality of every STS is the same, but the nature of the tasks that each STS performs depends on the role the STS plays in relation to the other STS Web services in your design.

In this article:

- How Web applications that use an STS work
- Configure a SharePoint claims-based Web application by using Windows PowerShell
- Edit bindings
- Configure a Web application that uses an STS

How Web applications that use an STS work

Web applications that use a security token service handle requests to issue, manage, and validate security tokens. Security tokens consist of a collection of identity claims (such as a user's name, role, or an anonymous identifier). Tokens can be issued in different formats, such as Security Assertion Markup Language (SAML) tokens. Security tokens can be protected with an X.509 certificate to protect the token's contents in transit and to enable validation of trusted issuers. For additional information about the Security Token Service, see Plan authentication methods (SharePoint Server 2010) (*http://technet.microsoft.com/library/40117fda-70a0-4e3d-8cd3-0def768da16c(Office.14).aspx*)

An Identity Provider-STS (IP-STS) is a Web service that handles requests for trusted identity claims. An IP-STS uses a database called an identity store to store and manage identities and their associated attributes. The identity store for an identity provider may be a simple, such as a SQL database table. An IP-STS may also use a complex identity store, such as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Service (AD LDS).

An IP-STS is available to clients who want to create and manage identities, and to relying party applications that must validate identities presented to them by clients. Each IP-STS has a federated trust relationship with, and issues tokens to, federation partner Relying Party STS Web applications, each of which are referred to as an RP-STS. Clients can create or provision managed Information Cards (using a card selector such as CardSpace) that represent identities registered with the IP-STS. Clients interact with the IP-STS when they request security tokens that represent an identity that is contained in the identity store of the IP-STS. After authentication, the IP-STS issues a trusted security token that the client can present to a relying party application. Relying party applications can establish trust relationships with an IP-STS. This enables them to validate the security tokens issued by an IP-

STS. After the trust relationship is established, relying party applications can examine security tokens presented by clients and determine the validity of the identity claims they contain.

A relying party STS (RP-STS) is an STS that receives security tokens from a trusted federation partner IP-STS. In turn, the RP-STS issues new security tokens to be consumed by a local relying party application. The use of RP-STS Web applications in federation with IP-STS Web applications enables organizations to offer Web single-sign-on (SSO) to users from partner organizations. Each organization continues to manage its own identity stores.

Configure a SharePoint claims-based Web application by using Windows PowerShell

Perform the following procedures to use Windows PowerShell to configure a SharePoint claims-based Web application.

To configure a SharePoint claims-based Web application by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt (that is, PS C:\>), create an x509Certificate2 object, as shown in the following example:

```
$cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2("path to cert
file")
```

6. Create a claim type mapping to use in your authentication provider, as shown in the following example:

```
New-SPClaimTypeMapping
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
-IncomingClaimTypeDisplayName "EmailAddress" -SameAsIncoming
```

7. Create a trusted login provider by first creating a value for the realm parameter, as shown in the following example:

```
$realm = "urn:" + $env:ComputerName + ":domain-int"
```

8. Create a value for the `signinurl` parameter that points to the Web application, as shown in the following example:

```
$signinurl = "https://test-2/FederationPassive/"
```

9. Create the trusted login provider, using the same `IdentifierClaim` value as in a claim mapping (`$map1.InputClaimType`), as shown in the following example:

```
$ap = New-SPTrustedIdentityTokenIssuer -Name

"WIF" -Description "Windows® Identity Foundation" -Realm

$realm -ImportTrustCertificate $cert

-ClaimsMappings $map1[,$map2..] -SignInUrl

$signinurl -IdentifierClaim $map1.InputClaimType
```

10. Create a Web application by first creating a value for the application pool account (for the current user), as shown in the following example:

```
$account = "DOMAIN\" + $env:UserName
```

Note:
The application pool account must be a managed account. To create a managed account, use `New-SPManagedAccount`.

11. Create a value for the Web application URL (`$webappurl = "https://" + $env:ComputerName`), as shown in the following example:

```
$wa = New-SPWebApplication -name "Claims WIF"

-SecureSocketsLayer -ApplicationPool "SharePoint SSL"

-ApplicationPoolAccount $account -Url $webappurl -Port 443

-AuthenticationProvider $ap
```

12. Create a site by first creating a claim object, as shown in the following example:

```
$claim = New-SPClaimsPrincipal

-TrustedIdentityTokenIssuerr $ap -Identity

$env:UserName
```

13. Create a site, as shown in the following example:

```
$site = New-SPSite $webappurl -OwnerAlias

$claim.ToEncodedString() -template "STS#0"
```

Edit bindings

After you have configured a SharePoint claims-based Web application, edit the bindings.

To edit bindings

1. Start IIS Manager by typing **INETMGR** at a command prompt.
2. Go to the **Claims Web Application** site in IIS.
3. In the left pane, right-click **Claims Web Application**, and select **Edit Bindings**.
4. Select **https** and click **Edit**.
5. Under **SSL Certificate**, select any listed certificate.

Configure a Web application that uses an STS

After you have configured a SharePoint Server 2010 claims-based Web application, edited the bindings and configured the Web.Config file, you can use the procedure in this section to configure a Security Token Service Web application.

To configure a Web application that uses an STS

1. Open the Active Directory Federation Services (AD FS) 2.0Management console.
2. In the left pane, expand **Policy**, and select **Relying Parties**.
3. In the right pane, click **Add Relying Party**. This opens the Active Directory Federation Services (AD FS) 2.0 configuration wizard.
4. On the first page of the wizard, click **Start**.
5. Click **Enter relying party configuration manually**, and click **Next**.
6. Type a relying party name and click **Next**.
7. Make sure **Active Directory Federation Services (AD FS) 2.0 Server Profile** is selected, and click **Next**.
8. If you are not planning to use an encryption certificate, click **Next**.
9. Select **Enable support for Web-browser-based identity federation**.
10. Type the name of the Web application URL, and append **/_trust/** (for example: **https://servername/_trust/**). Click **Next**.
11. Type an identifier, and click **Add**. Click **Next**.
12. On the Summary page, click **Next** and then click **Close**. This opens the Rules Editor Management console. Use this console to configure the mapping of claims from an LDAP Web application to SharePoint.
13. In the left pane, expand **New Rule**, and select **Predefined Rule**.
14. Select **Create Claims from LDAP Attribute Store**.
15. In the right pane, from the **Attribute Store** drop-down list, select **Enterprise Active Directory User Account Store**.
16. Under **LDAP Attribute**, select **sAMAccountName**.
17. Under **Outgoing Claim Type**, select **E-Mail Address**.
18. In the left pane, click **Save**.

Configure the Secure Store Service (SharePoint Server 2010)

This article describes the Microsoft SharePoint Server 2010 Secure Store Service operations that enable solution designers to create target applications that map user and group credentials to the credentials of external data sources. By using these target applications, external content types in the Business Data Connectivity service will be able to interact with their external data sources to read, write, create, and edit data stored in external data sources. For an overview of the Secure Store Service, see Plan the Secure Store Service (SharePoint Server 2010) (*http://technet.microsoft.com/library/e1196de0-1eb9-4c50-aeca-93e0eba7be0d(Office.14).aspx*)

Before using the Secure Store Service to create target applications, you must provide it with a pass phrase. The pass phrase is used to generate a key that is used to encrypt and decrypt the credentials that are stored in the Secure Store Service database. If you have to supply the initial pass phrase, you will see the following message when you open a Secure Store Service application instance: "Please generate a new key for this Secure Store Service application."

In this article:

- Initialize an instance of a Secure Store Service application
- Refresh the encryption key
- Generate a new encryption key
- Create a target application
- Set credentials for a target application
- Enable the audit log

Note:
You use Central Administration to complete the following procedures. If you want to use Windows PowerShell, see Configure Secure Store Service using PowerShell (*http://go.microsoft.com/fwlink/?LinkId=207030*) on the Script Center and the "Creating Secure Store Service and Proxy" lines from Todd Carter's following blog post, The Wizard Likes His GUIDs (*http://go.microsoft.com/fwlink/?LinkId=207031*). Additionally, if you want to use an audit log, you need to use the **AuditingEnabled** and **AuditlogMaxSize** parameters of either the New-SPSecureStoreServiceApplication (*http://technet.microsoft.com/library/d8e359da-87e6-4e26-92ec-522b18fa965b(Office.14).aspx*) let or the Set-SPSecureStoreServiceApplication (*http://technet.microsoft.com/library/6a41f2b2-9cf9-41be-afd9-65b97f100f3c(Office.14).aspx*) cmdlet.

Initialize an instance of a Secure Store Service application

You can use commands in the **Edit** group of the Ribbon to initialize an instance of a Secure Store Service application.

To initialize an instance of a Secure Store Service application

1. Verify that you have the following administrative credentials:
   - You must be a Service Application Administrator for the instance of the Secure Store Service.
2. In an instance of a Secure Store Service application, click the **Manage** tab.
3. In the Key Management group, click **Generate New Key**.
4. On the **Generate New Key** page, type a pass phrase string in the **Pass Phrase** box, and type the same string in the **Confirm Pass Phrase** box.

   Important

   A pass phrase string must be at least eight characters and must have at least three of the following four elements:

   Tip:
   The pass phrase that you enter will not be stored. Make sure you write this down and store it in a safe place. You will need it to refresh the key, such as when you add a new application server to the server farm.
5. Click **OK**.

You may be required to refresh the encryption key if:

- You add a new application server to the server farm.
- You restore a previously backed up Secure Store Service database and have since changed the encryption key.
- You receive an “Unable to get master key” error message.

Refresh the encryption key

You can use commands in the **Key Management** group of the Ribbon to refresh the encryption key.

To refresh the encryption key

1. Verify that you have the following administrative credentials:
   - You must be a Service Application Administrator for the instance of the Secure Store Service.
2. In an instance of a Secure Store Service application, click the **Manage** tab.

3. In the **Key Management** group, click **Refresh Key**.
4. In the **Pass Phrase** box, type the pass phrase that you initially used to generate the encryption key.

   This will either be the pass phrase that you used when you initialized the Secure Store Service application or one that you used when you created a new key by using the **Generate a New Key** command.
5. Click **OK**.

Generate a new encryption key

For security precautions or as part of regular maintenance you may decide to generate a new encryption key and optionally force the Secure Store Service to be re-encrypted based on the new key.

Caution:
You should back up the database of the Secure Store Service application before generating a new key.

To generate a new encryption key

1. Verify that you have the following administrative credentials:
   - You must be a Service Application Administrator for the instance of the Secure Store Service.
2. In an instance of a Secure Store Service application, click the **Manage** tab.
3. In the **Key Management** group, click **Generate New Key**.
4. On the **Generate New Key** page, type a pass phrase string in the **Pass Phrase** box, and type the same string in the **Confirm Pass Phrase** box.

   Important
   A pass phrase string must be at least eight characters and must have at least three of the following four elements:

   Tip:
   The pass phrase that you enter will not be stored. Make sure you write this down and store it in a safe place. You will need it to refresh the key, such as when you add a new application server to the server farm.
5. To force re-encryption of the Secure Store Service database, click **Re-encrypt the database using the new key**.
6. Click **OK**.

Create a target application

You use the Secure Store Service to create target applications. A target application maps the credentials of a user, group, or claim to a set of credentials on an external data source such as a SQL Server database or a Web service. After a target application is created, you can associate it with an external content type or application model to provide access to an external data source.

To create a target application

1. Verify that you have the following administrative credentials:
   - You must be a Service Application Administrator for the instance of the Secure Store Service.
2. In an instance of a Secure Store Service application, click the **Manage** tab.
3. In the **Manage Target Applications** group, click **New**.
4. In the **Target Application ID** box, type a text string.

   This is the unique string that will be used internally by the Secure Store Service application to identify this target application.
5. In the **Display Name** box, type a text string that will be used to display the identifier of the target application in the user interface.
6. In the **Contact Email** box, type the e-mail address of the primary contact for this target application.

   This can be any legitimate e-mail address and does not have to be the identity of an administrator of the Secure Store Service application.
7. When you create a target application of type Individual (see below), you can implement a custom Web page that lets users add individual credentials for the destination data source. This requires custom code to pass the credentials to the target application. If you did this, type the full URL of this page in the **Target Application Page URL** field. There are three options:
   - **Use provided page**: Any Web sites that use the target application to access external data will have an individual sign-up page added automatically. The URL of this page will be http:/<samplesite>/_layouts/SecureStoreSetCredentials.aspx?TargetAppId=<TargetApplicationID>, where <TargetApplicationID> is the string typed in the **Target Application ID** box. By publicizing the location of this page, you can enable users to add their credentials for the external data source.
   - **Use custom page**: You provide a custom Web page that lets users provide individual credentials. Type the URL of the custom page in this field.
   - **None**: There is no sign-up page. Individual credentials are added only by a Secure Store Service administrator using the Secure Store Service application.
8. In the **Target Application Type** box, type the target application type: **Group**, for group credentials, or, **Individual** if each individual is to be mapped to a unique set of credentials on the external data source.

Note

There are two primary types for creating a target application:

9. If the credentials on the external data source are Windows credentials, check the **Windows** check box.

   Clear the check box if the credentials on the external data source are not Windows credentials.

10. Click **Next** to configure the fields to use to submit credentials to the external data source.
11. Use the **Specify the credential fields for your Secure Store Target Application** page to configure the various fields which may be required to provide credentials to the external data source. Two fields are listed by default: User Name and Password.

    To add an additional field for supplying credentials to the external data source, on the **Specify the credential fields for your Secure Store Target Application** page, click **Add Field**.

    By default, the type of the new field is Generic. The following field types are available:

Field	Description
Generic	Values that do not fit in any of the other categories.
User Name	A user account that identifies the user.
Password	A secret word or phrase.
PIN	A personal identification number.
Key	A parameter that determines the functional output of a cryptographic algorithm or cipher.
Windows User Name	A Windows user account that identifies the user.
Windows Password	A secret word or phrase for a Windows account.

- To change the type of a new or existing field, click the arrow that appears next to the type of the field, and then select the new type of field.

  Note:
  Every field that you add will be required to have data when submitted to set credentials.

- You can change the name that a user sees when interacting with a field. In the Field Name column of the **Specify the credential fields for your Secure Store Target Application** page, change a field name by selecting the current text and typing new text.

- When a field is masked, each character that a user types is not displayed but is replaced with a mask character such as the asterisk “*”. To mask a field, click the check box for that field in the **Is Masked** column of the page.
- To delete a field, click the delete icon for that field in the **Delete** column of the page.

When you have finished editing the credential fields, click **Next**.

12. In the **Specify the membership settings** page, in the **Target Application Administrators Field**, list all users who have access to manage the target application settings.
13. If the target application type is group, in the **Members** field, list the user groups to map to a set of credentials for this target application.
14. Click **OK** to complete configuring the target application.

Set credentials for a target application

After creating a target application, an administrator of that target application can set credentials for it. These credentials will be used by the Microsoft Business Connectivity Services and other services to provide access to an external data source. If the target application is of type Individual, you can optionally enable individuals to supply their own credentials.

To set credentials for a target application

1. Verify that you have the following administrative credentials:
   - You must be a Service Application Administrator for the instance of the Secure Store Service.
2. In an instance of a Secure Store Service application, point to the target application identifier, click the arrow that appears, and then, in the menu, click **Set credentials**.

   If the target application is of type Group, type the credentials for the external data source. Depending on the information that is required by the external data source, the fields for setting credentials will vary.

   If the target application is of type Individual, type the user name of the individual who will be mapped to a set of credentials on the external data source, and type the credentials for the external data source. Depending on the information that is required by the external data source, the fields for setting credentials will vary.

Enable the audit log

Audit entries for the Secure Store service are stored in the Secure Store Service database. By default, the audit log file is disabled.

An audit log entry stores information about a Secure Store Service action, including when it was performed, whether it succeeded, why it failed if it didn't succeed, the Secure Store Service user who

performed it, and optionally the Secure Store Service user on whose behalf it was performed. Therefore, a valid reason to enable an audit log file is to troubleshoot an authentication issue.

Note:
If the Secure Store Service database is set to read-only, the audit log file must be disabled, otherwise the following error message will be displayed during authentication: "Cannot complete this action as the Secure Store Shared Service is not responding. Please contact your administrator."

To enable the audit log by using Central Administration

1. Verify that the user account that is performing this procedure is a member of the Farm Administrators SharePoint group.
2. In Central Administration, on the Home page, click **Application Management**.
3. On the Application Management page, in the **Service Applications** section, click **Manage service applications**.
4. On the **Service Application** tab, click **Secure Store** (the type should be associated with the Secure Store Service Application).
5. On the ribbon, click **Properties**.
6. From the **Enable Audit** section, click to select the **Audit log enabled** check box.
7. To change the number of days that entries will be purged from the audit log file, specify a number in days in the **Days Until Purge** field. The default value is 30 days.
8. Click **OK**.

Post-installation steps for search (SharePoint Server 2010)

This article describes post-installation steps for configuring the search system in a single-server deployment. Some steps differ depending on whether the product was installed with the Standalone or the Server Farm installation option. This article provides the steps for both cases.

In this article:

- Configure basic search functionality to enable end-users to get search results.
- Enable search reports by ensuring that the Web Analytics service application is running.
- Enable people search so that people can find others in the organization.
- Optional steps.

Configure basic search functionality

The following table describes the configuration of key search components of SharePoint Server 2010 after a Standalone or Server Farm installation.

Requirement	Exists after Standalone installation?	Exists after Server Farm installation?
Search service application	Yes[1]	Depends[1]
Search Center site	Depends[2]	Depends[2]
Contact e-mail address	Yes. Configured with fictitious address: someone@example.com	Yes. Configured with fictitious address: someone@example.com
Default content access account	Yes. Configured as NT Authority\Local Service	Yes Configured by using the account specified in the Farm Configuration Wizard during installation.

[1]Assumes that the administrator did not clear the Search check-box when running the Farm Configuration Wizard during installation.

[2]A Search Center site exists only if the administrator who installed SharePoint Server 2010 selected a site template that contains a Search Center site when creating the site collection.

To go to the Search Administration page

1. Verify that the user account that is performing this procedure is an administrator for the Search service application.
2. On the Home page of the Central Administration Web site, in the **Application Management** section, click **Manage service applications**.
3. On the Manage Service Applications page, , click **Search Service Application**. The Search Administration page appears.

Specify the default content access account

A user account is assigned as the default content access account during installation. The crawler uses this account to crawl content that does not have a crawl rule that specifies a different account. For the default content access account, we recommend that you specify a domain user account that has read access to as much of the content that you want to crawl as possible.

You can change the default content access account at any time. For information about creating a crawl rule, see Create a crawl rule later in this article.

To specify the default content access account

1. On the Search Administration page, in the **System Status** section, look at the account that is shown in the **Default content access account** row. If the account that is shown is the one that you want crawlers to use when crawling content, skip to the next procedure. Otherwise proceed to step 2.
2. Click the link in the **Default content access account** row.
3. On the Default Content Access Account dialog box, in the **Account** box, type a different user name in the form of domain\user name.
4. Type the password for this account in the **Password** and **Confirm Password** boxes.
5. Click **OK**.

Specify the contact e-mail address

The contact e-mail address is written to the logs of crawled servers. By default, the contact e-mail address is someone@example.com. We recommend that you change this to the account that an external administrator can contact when a crawl might be contributing to a problem such as performance degradation on a server that the system is crawling.

To specify the contact e-mail address

1. On the Search Administration page, in the **System Status** section, click the link for the **Contact e-mail address**.
2. In the **Search Email Setting** dialog box, in the **E-mail Address** box, type the e-mail address that you want to appear in the logs of servers that are crawled by the search system.
3. Click **OK**.

Configure farm-wide search settings

Farm-wide search settings include the following:

- Proxy server

  The proxy server that the index server uses when crawling content that is external to the server farm. By default no proxy server is selected.

- Search time-out setting

  The amount of time that you want the system to wait while connecting to other services, and how long the system waits for a request for content to be acknowledged. By default, this is set to 60 seconds to wait for a connection and 60 seconds before requesting an acknowledgement.

- SSL warnings

  Specify whether to ignore SSL certificate name warnings. By default, this is set to **No**.

To configure farm-wide search settings

1. On the Search Administration page, in the Quick Launch, click **Farm-Wide Search Administration**.
2. To specify a proxy server, click the link in the **Proxy server** row. Otherwise, skip to step 3. For more information, see Configure farm-level proxy server settings (SharePoint Server 2010).
3. To specify a different time-out duration, click the link in the **Time-out (seconds)** row. Otherwise, skip to step 4. For more information, see Configure search time-out settings (SharePoint Server 2010) (*http://technet.microsoft.com/library/ae686fd8-4e65-40f6-b86d-14ef50b0d354(Office.14).aspx*).
4. To specify whether to ignore SSL certificate warnings, click the link in the **Ignore SSL Warnings** row. Otherwise, skip to the next section. For more information, see Configure SSL certificate warning settings (SharePoint Server 2010) (*http://technet.microsoft.com/library/432973c1-21c1-4fdc-87e6-a2252596937a(Office.14).aspx*).

Create content sources for crawling content

Crawling content requires at least one content source. A content source called Local SharePoint sites is created by default during installation and is configured to crawl all of the sites in the server farm.

If you perform a Standalone installation of SharePoint Server 2010, a full crawl of local SharePoint sites is automatically performed following installation and an incremental crawl is scheduled to occur every 20 minutes after that. For an Server Farm installation of SharePoint Server 2010, this content source is created, but no crawls are automatically performed or scheduled.

Creating and configuring a content source in SharePoint Server 2010 is the same as it was in the previous version, with the following exceptions:

- The Business Data Catalog is now a subset of the Line of Business Data content source type.
- There is a new content source type for crawling custom repositories. This option requires that one or more custom connectors are registered.
- A new section named **Content Source Priority** is available when adding or editing a content source. You can use this section to specify that the content source runs with either normal or high priority. The default setting is Normal.

Search Center site

After installation, depending on the kind of installation that you performed and depending on the template you selected, you might have a Search Center site.

In your Web browser, go to the top-level site for the site collection that installation created. If that site does not have a tab named Search or if it is not a Search Center site, you must create a Search Center site. If you do not have a Search Center site, use the following procedure to create one.

To create a site collection by using the Search Center template

1. Verify that the user account that is performing this procedure is a member of the Farm Administrators SharePoint group.
2. On the Home page of the Central Administration Web site, in the **Application Management** section, click **Create site collections**.
3. On the Create Site Collection page, do the following:
    a. In **Title and Description** section, in the **Title** box, type the name you want for this Search Center site. Optionally, type a description in the **Description** box.
    b. In the **Web Site Address** section, type the last part of the URL for this site.
    c. In the **Template Selection** section, click the **Enterprise** tab and then select either the **Enterprise Search Center** or **Basic Search Center** template. If you intend to use people search, you must select the Enterprise Search Center template.
    d. In the **Primary Site Collection Administrator** section, in the **User name** box, type the user name of the primary site collection administrator for this site collection in the form domain\user name.
    e. (Optional) In the **Secondary Site Collection Administrator** section, type the user name of a secondary site collection administrator.
    f. Click **OK**.

4. On the Top-Level Site Successfully Created page, click the link to the Search Center site that you just created.

Grant all authenticated users access to the Search Center

1. Verify that the user account that is performing this procedure is a site collection administrator for the Search Center site.
2. In a Web browser, go to the Search Center site.
3. On the **Site Actions** menu, click **Site Settings**.
4. In the **Users and Permissions** section, click **People and groups**.
5. On the Quick Launch, click either **Enterprise Search Center Visitors** or **Basic Search Center Visitors**, depending on the template that you used to create this Search Center.
6. In the **New** menu, click **Add Users**.
7. In the **Users/Groups** box, type **NT Authority\authenticated users**.
8. Click **OK**.

Enable search reporting

Search-related reports depend on the Web Analytics Web Service. This service is created when you run the Farm Configuration Wizard during installation.

Verify that the Web Analytics service application is started

1. Verify that the user account that is performing this procedure is an administrator for the Search service application.
2. On the Home page of the Central Administration Web site, in the **Application Management** section, click **Manage service applications**.
3. On the Manage Service Applications page, verify that the Web Analytics service application is in the list of service applications, and then do one of the following:
   - If the service application exists, ensure that **Started** is displayed in the **Status** column for this service application. If the **Status** column displays **Started** for this service application, skip to the “Run search queries” section later in this article. Otherwise, go to the procedure titled “Start the Web Analytics service application. If the Web Analytics service application was created without using the Farm Configuration Wizard, the dependent service for the application is not started.
   - If the Web Analytics service application does not exist, use the following procedure to create it.

Create the Web Analytics service application

To create the Web Analytics service application

1. Verify that the user account that is performing this procedure is a member of the Farm Administrators SharePoint group.
2. On the Home page of the Central Administration Web site, in the Quick Launch, click **Configuration Wizards**.
3. Click **Launch the Farm Configuration Wizard**.
4. On the Help Make SharePoint Better page, select one for the following:
   - **Yes, I am willing to participate (Recommended)**.
   - **No, I don't wish to participate**.
5. Click **OK**.
6. On the Configure your SharePoint farm page, accept the default setting, and then click **Next**.
7. On the Configure your SharePoint farm page, in the **Service Account** section, select **Use existing managed account**.
8. Ensure that the Web Analytics Web Service Application check box is selected, and then click **Next**.
9. On the Create Site Collection page, click **Skip**.
10. On the Configure your SharePoint farm page, click **Finish**.
11. Proceed to Run search queries.

Start the Web Analytics Web service

To start the Web Analytics Web service

1. Verify that the user account that is performing this procedure is a member of the Farm Administrators SharePoint group.
2. On the Home page of the Central Administration Web site, in the **System Settings** section, click **Manage servers in this farm**.
3. On the Servers in Farm page, in the **Server** column, click the link for the server.
4. On the Services on Server page, in the **Web Analytics Web Service** row, in the **Action** column, click **Start**.

   Note that if the link name is **Stop**, this service is already started and no further action is necessary.

Run search queries

Use the Search Center Web site and optionally the Search box in your other site collections, such as a Team site, to run search queries. Queries reports are compiled separately for each site collection, including the site collection that hosts the Search Center Web site.

Tip:

Query reports are processed daily. Typically you must wait about 24 hours after the queries run before they appear in query reports.

View query reports

Search administrators can view query reports at two levels: Web application level and site collection level. Other search reports, primarily reports concerning crawling content, are available.

Note:

You must be a farm administrator to perform the following steps.

View query reports at the Web application level

1. In the Central Administration Web site, on the Quick Launch, click **Home**.
2. In the Monitoring section, click **View web analytics reports**.
3. On the Web Analytics Reports – Summary page, in the **Web Application Name** column, click the name of the Web application for which you want to view reports.

   By default, this name is SharePoint – 80.

   A summary of all reports is displayed.
4. On the Quick Launch in the Search section, click **Number of Queries** to view a detailed query report.

Note:

You must be a farm administrator or site collection administrator of the site collection on which you view reports to perform the following steps.

View query reports at the site-collection level

1. In your Web browser, go to the top-level site of the site collection for which you want to see query reports.

   For example, your Team site or the Search Center site.

2. On the site Actions menu, click **Site Settings**.
3. In the Site Collection Administration section, click **Site collection web analytics reports**.
4. On the Site collection Web Analytics Reports – Summary page, on the Quick Launch, click the links in the Search section to see the various search related reports.

Note:

You must be a farm administrator to perform the following steps.

View other search-related reports

1. In the Central Administration Web site, on the Quick launch, click **Monitoring**.
2. In the Reporting section, click **View administrative reports**.
3. In the Type column, click the **Search** folder.
4. In the Name column, click the name of the report you want to view.

Enable people search

People search requires that the following service applications were created and that the services on which they depend were started.

- Search
- User Profile
- Managed Metadata

The User Profile service application is an important part of people search because it stores much of the information that appears in results for people search. For example, information in My Site sites and a lightweight directory access protocol (LDAP) store, such as Active Directory Domain Services (AD DS) are stored by the User Profile service application.

In addition to the service applications, people search requires a Search Center site that is created by using the Enterprise Search Center template. Unlike the Basic Search Center, the Enterprise Search Center has a tab that is used to search for people.

Ensure service applications are running

This section explains how to verify the existence of the service applications that people search requires, helps you to create the service applications if they do not exist, and explains how to ensure that dependent services are started.

To verify that service applications exist

1. Verify that the user account that is performing this procedure is a member of the Farm Administrators SharePoint group.
2. On the Home page of the Central Administration Web site, in the **Application Management** section, click **Manage service applications**.
3. On the Manage Service Applications page, verify that the following are listed in the **Name** column:
   - Managed Metadata Service
   - Search Service Application
   - User Profile Service
4. If any of these service applications do not exist, follow the steps in the **Create service applications** section. Otherwise, skip to **Ensure Managed Metadata Service application is started**.

To create service applications

1. On the Home page of the Central Administration Web site, in the Quick Launch, click **Configuration Wizards**.
2. On the Configuration Wizards page, click **Launch the Farm Configuration Wizard**.
3. On the Help Make SharePoint Better page, click one of the following options, and then click **OK**:
   - **Yes, I am willing to participate (Recommended)**
   - **No, I don't want to participate**
4. On the Configure your SharePoint farm page, select **Walk me through the settings using this wizard**, and then click **Next**.
5. On the Configure your SharePoint farm page, in the Service Account section, select **Use an existing managed account**.
6. Select the check-boxes for the service applications that you need to create, and then click **Next**.
7. On the Create Site Collection page, click **Skip**.
8. On the Configure your SharePoint farm page, click **Finish**.

To ensure Managed Metadata Service application is started

1. In the **Application Management** section, click **Manage service applications**.
2. Ensure that the **Status** column for the Managed Metadata Service application shows **Started**. If the farm configuration wizard was not used to create this service application, the Managed Metadata Web Service might not be running. If this is the case, continue to step 2. Otherwise, skip to **Configure Web application**.

3. On the Central Administration Home page, in the **System Settings** section, click **Manage services on server**.
4. On the Services on Server page, in the Managed Metadata Web Service row, click the **Start** link in the **Action** column.

   Note:
   If the link in the **Action** column is named Stop, the Managed Metadata Web Service is already running.

Configure Web application and site collections

In a production environment, you typically create a separate Web application to host the site collection for My Site sites. For evaluation purposes, you can use the same Web application to host separate site collections for your SharePoint sites, Search Center site, and My Site sites.

This article assumes that you will use the same Web application to host the site collections for SharePoint sites, the Search Center site, and My Site sites. The server farm should already have at least two Web applications: one for the Central Administration Web site and one for SharePoint sites. These conclusions are based on the assumption that you used the procedures in Deploy a single server with SQL Server (SharePoint Server 2010) to install SharePoint Server.

Ensure a Web application has been created for SharePoint sites, the Search Center site, and My Site sites

1. Verify that the user account that is performing this procedure is a member of the Farm Administrators SharePoint group.
2. On the Home page of the Central Administration Web site, in the **Application Management** section, click **Manage web applications**.
3. On the Web Applications Management page, all Web applications that have been created for this server farm are listed.

   The steps in Deploy a single server with SQL Server (SharePoint Server 2010) in the deployment guide create a Web application that uses port 80. You can use that Web application to host the site collections for your SharePoint sites, Search Center site, and My Site sites.

   Tip:
   If you performed a Standalone installation, a Web application named SharePoint – 80 is automatically created on port 80 and a site collection is created in that Web application by using the Team Site template. This Web application could have also been created after an Server Farm installation if a site collection was created by using the Farm Configuration Wizard.
4. If a Web application does not exist for your SharePoint sites and My Site sites, go to step 5. Otherwise, skip to “Ensure service applications are associated with Web application.”

5. On the Web applications Management page, on the ribbon, in the **Contribute** group, click **New**..
6. In the **Create New Web Application** dialog box, select **Create a new IIS web site** and optionally type a name for the site in the **Name** box.
7. In the **Port** box, type the port number that you want this Web application to use.
8. Specify any other settings, and then click **OK**.
9. On the Application Created page, click **OK**.

Ensure service applications are associated with Web application

1. On the Web Applications Management page, in the Name column, click the row that contains the Web application that you want to ensure is associated with the required service applicationsTypically, this is named SharePoint -80.

   Note

   You must ensure that the Web applications for your SharePoint sites, Search Center site, and My Site sites are associated with the following service applications:
2. On the ribbon, in the **Manage** group, click **Service Connections**.
3. In the **Configure Service Application Associations** dialog box, ensure that the following check boxes are selected: Managed Metadata Service Connection, User Profile Service Proxy, Search Service Application Proxy.
4. Click **OK**.

Add user profiles to the User Profile service application

Before you can get meaningful people search results, you must add user profiles to the User Profile service application. There are two ways to do this:

- Add user profiles manually
- Synchronize with a directory service or external data source, such as AD DS, third party LDAP, or the Business Data Connectivity service

Important:

For a test environment, we recommend that you do not synchronize the profile store to a directory service or other external data source that is in a production environment. Instead, make a copy of the directory service with which to synchronize the User Profile service application.

View a list of user profiles

1. On the Manage Service Applications page, click the row that contains the User Profile service application, and then on the ribbon, click **Manage**.

2. On the Manage Profile Service page, in the **People** section, click **Manage User Profiles**.
3. On the **Manage User Profiles** page, in the **Find profiles** box, type the name of the domain to which the users are a member, and then click **Find**.

   For example, if your users are members of the Contoso.com domain, type Contoso in the **Find profiles** box. Do not type the fully qualified domain name.
4. Click **Find**.

Configure My Site settings

My Site settings are configured, by default, when the Farm Configuration Wizard is used to create the User Profile service application.

Perform the following steps if you want to change the default settings.

1. On the Manage Service Applications page, click the row that contains the User Profile service application.
2. In the **Operations** group of the ribbon, click **Manage**.
3. On the Manage Profile Service page, in the **My Site Settings** section, click **Setup My Sites**.
4. In the **Preferred Search Center** section, in the **Preferred Search Center** box, ensure that the path of the Search Center site is correct.
5. In the **Search scope for finding people** list, ensure that **People** is selected.
6. In the **Search scope for finding documents** list, select the search scope that you want to use for finding documents.

   We recommend that you select the **All Sites** scope.
7. In the **My Site Host** section, in the **My Site Host location** box, ensure that the path to your My Site host is correct.

   By default, the path is my/personal.
8. In the **Site Naming Format** section, select the format you want to use to name new personal sites.

   Note that **User name (do not resolve conflicts)** is selected by default.
9. In the **Language Options** section, optionally choose whether to allow users to choose the language of their personal site and whether the user profile will be edited using the language of their personal site.
10. In the **Multiple Deployments** section, you can optionally enable My Sites to support global deployments.

    Doing so will allow a user whose My Site is hosted by a different User Profile service application to perform actions such as adding colleagues and links to their default My Site and viewing people search results grouped by social distance.

11. In the **Read Permission Level** section, optionally enter one or more accounts that will be granted the Read permission level in the personal site when it is created.

    By default, all authenticated users are granted this permission.
12. In the **MySite Provisioning** section, optionally configure the way MySites are provisioned.
13. In the **My site E-mail Notifications** section, optionally specify a different name that will appear in all e-mail notifications sent from all My Site sites.

    By default, the sender's name is My Site.
14. Click **OK**.

Add information to My Site sites

A My Site stores its information in the User Profile service application. Therefore, the more information you add to a My Site site, the more interesting people search results will be.

For example, we recommend that you provide information about user's expertise and interest in My Site sites to see how they appear on search results pages.

To add information to My Site sites, log on as a user for whom a user profile has been created in the User Profile service application and go to the URL of your My Site. To do this, in a Web browser, go to http:// *hostname*/my/. If your My Site has not been created, one will be automatically created for you.

Crawl the profile store

In earlier versions of SharePoint Products and Technologies, the URL that is used to crawl the profile store was automatically provided in the default content source. This is not the case with SharePoint Server 2010. We recommend that you create a new content source so that you can crawl the profile store on a separate schedule from your SharePoint sites.

Before you crawl the profile store, we recommend that you ensure that the user account assigned to the default content access account is an administrator on the User Profile service application.

To add an administrator to the User profile service application

1. On the Manage Service Applications page, click the row that contains the User Profile service application, and then in the ribbon, click **Administrators**.
2. In the **Administrators for User Profile Service** dialog box, in the **To add an account…** box, type the user account you want to add in the form of domain\user name.
3. Click **Add**.
4. In the **Permissions** list box, select the **Full Control** check box.
5. Click **OK**.

Create a content source to crawl the profile store

1. Verify that the user account that is performing this procedure is an administrator for the Search service application.
2. On the Home page of the Central Administration site, in the **Application Management** section, click **Manage service applications**.
3. On the Manage Service Applications page, click **Search Service Application**.
4. On the Search Administration page, in the Quick Launch, in the **Crawling** section, click **Content sources**.
5. On the Manage Content Sources page, click **New Content Source**.
6. On the Add Content Source page, in the **Name** section, type a name for this content source.
7. In the **Content Source Type** section, ensure that **SharePoint Sites** is selected.
8. In the **Start Addresses** section, type **sps3://hostname**, where host name is the host name of the server farm.
9. In the **Crawl Schedules** section, optionally create schedules for full and incremental crawls.
10. In the **Start Full Crawl** section select the **Start full crawl of this content source** check box, and then click **OK**.

Verify the crawl

1. On Search Administration page, in the Quick Launch, click **Search Administration**.

   Note:
   We recommend that you view the crawl log to determine if any crawl errors occurred.
2. In the **Crawl History** section, click the **Refresh** button to refresh the crawl status.

   After the crawl of the content source completes, verify that a zero is listed in the **All Errors** column. If errors occurred, the number in the **All Errors** column will be greater than zero. In that case, click the number in the **All Errors** column to view the crawl log for that content source. Otherwise if no errors occurred, proceed to “Optional Steps”.

   Note:
   If the crawl log indicates that an authentication error occurred, you can resolve the problem by either following the steps described in the “Add an administrator to the People service application” section earlier in this article or by creating a crawl rule to use the account of an existing administrator of the People service application to authenticate the crawler. For information about creating a crawl rule, see “Create a crawl rule (optional)” later in this article.

Optional steps

The following procedures are optional.

Create a crawl rule (optional)

If the account you assigned as the default content access account in the “Specify the default content access account” section earlier in this article is not an administrator on the User Profile service application, we recommend that you create a crawl rule that the crawler can use to authenticate to the User Profile service application.

Note:
This procedure is not necessary if you performed the steps shown in the “Add an administrator to the User Profile service application section”, earlier in this article.

1. On the Home page of the Central Administration Web site, in the Application Management section, click **Manage service applications**..
2. On the Mange Service Applications page, click **Search Service Application**.
3. On the Search Administration page, in the Quick Launch, in the **Crawling** section, click **Crawl rules**.
4. On the Manage Crawl Rules page, click **New Crawl Rule**.
5. In the Path section, in the **Path** box, type the start address for the User Profile service application in the form of **sps3://hostname**, where hostname is the host name of the server farm.
6. In the **Crawl Configuration** section, select **Include all items in this path**.
7. In the **Specify Authentication** section, select **Specify a different content access account**.
8. In the **Account** box that appears, type the name of user account that is an administrator on the User Profile service application in the form of domain\user name.
9. Type the password for the account you specified in the previous step in the **Password** and **Confirm Password** boxes.
10. Click **OK**.

Configure a managed metadata service application (SharePoint Server 2010)

In Microsoft SharePoint Server 2010, you can configure a managed metadata service application manually or by using Windows PowerShell.

The following articles provide detailed information about how to configure a managed metadata service application and a managed metadata service connection:

- Create, update, publish, or delete a managed metadata service application (SharePoint Server 2010) (*http://technet.microsoft.com/library/be30ab95-f6bf-41e1-93ac-2f5c78ac703b(Office.14).aspx*)
- Create, update, or delete a managed metadata service connection (SharePoint Server 2010) (*http://technet.microsoft.com/library/94f8453b-bebb-4fbd-ab9b-ddb1f6136a50(Office.14).aspx*)

See Also

Managed metadata service application overview (SharePoint Server 2010) (*http://technet.microsoft.com/library/ecd9e1d1-3e56-4ab9-80e1-29972586c5a7(Office.14).aspx*)

Prepare to host sites (SharePoint Server 2010)

After you have installed Microsoft SharePoint Server 2010 and performed the initial configuration, you can begin to create SharePoint sites. Before you create a site collection, you must first create a Web application in which to create the site collection. If you want to use Kerberos authentication or claims-based authentication, you must also configure Web application authentication.

The articles in this section help you determine the hierarchy of your SharePoint sites and how to configure claims authentication and Kerberos authentication in the server farm.

- Create a Web application (SharePoint Server 2010)

  SharePoint sites are hosted by Web applications, so you must create one or more Web applications before you can create any sites. This article covers how to create a Web application from the Central Administration Web site or by using Windows PowerShell 2.0.

- Configure claims authentication (SharePoint Server 2010)

  This article describes how to configure a SharePoint Server 2010 claims-based Web application that will provide a claims-based sign-in and services infrastructure for your farm. The article also describes how to configure anonymous access for your claims-based Web application and how to configure your Web application zones for forms-based authentication or Web SSO authentication, as appropriate.

- Configure Kerberos authentication (SharePoint Server 2010)

  To deploy a server farm running SharePoint Server 2010 using Kerberos authentication, you must install and configure a variety of applications on your computers. This article describes an example server farm running SharePoint Server 2010 and provides guidance for deploying and configuring the farm to use Kerberos authentication to work in a SharePoint server farm.

- Create a site collection (SharePoint Server 2010)

  After you have created a Web application, you can create a site collection. This article helps you create a site collection from the Central Administration Web site or by using Windows PowerShell 2.0. If you want to enable users to create their own sites, you need to configure self-service site management for the Web application.

Create a Web application (SharePoint Server 2010)

A Web application is composed of an Internet Information Services (IIS) Web site that acts as a logical unit for the site collections that you create. Before you can create a site collection, you must first create a Web application.

Each Web application is represented by a different IIS Web site with a unique or shared application pool. You can assign each Web application a unique domain name, which helps to prevent cross-site scripting attacks.

You use Web applications to isolate content. When you create a new Web application, you also create a new content database and define the authentication method used to connect to the database. In addition, you define an authentication method to be used by the IIS Web site in SharePoint Server 2010.

SharePoint Server 2010 offers two ways of authenticating users, as follows:

- Classic mode authentication, through which users log on to a Web application by using Windows authentication. For more information, see Create a Web application that uses Windows-classic authentication (SharePoint Server 2010).
- Claims-based authentication, through which users log on to a Web application by using Windows authentication, forms-based authentication (FBA), or Trusted Identity provider (SAML). If you use FBA or SAML, you must perform additional configuration steps. For more information about claims-based authentication, see Create a Web application that uses Windows-claims authentication (SharePoint Server 2010).

For more information about both types of authentication, see Plan authentication methods (SharePoint Server 2010) (*http://technet.microsoft.com/library/40117fda-70a0-4e3d-8cd3-0def768da16c(Office.14).aspx*).

SharePoint Server 2010 provides a set of services applications that are available for each Web application. You can select which service applications you want to use for each Web application that you create. For more information, see Technical diagrams (SharePoint Server 2010) (*http://technet.microsoft.com/library/bcbae7bd-656b-4003-969c-8411b81fcd77(Office.14).aspx*), Define managed paths (SharePoint Server 2010) (*http://technet.microsoft.com/library/0f95a1e6-7044-487e-8681-b1d717caabb3(Office.14).aspx*), and Service application and service management (SharePoint Server 2010).

In this section:

- Create a Web application that uses Windows-classic authentication (SharePoint Server 2010)
- Create a Web application that uses Windows-claims authentication (SharePoint Server 2010)

See Also

Extend a Web application (SharePoint Server 2010) (*http://technet.microsoft.com/library/02dc86bd-5918-4a01-89e9-d04508c3cc72(Office.14).aspx*)

Create a site collection (SharePoint Server 2010)

Logical architecture planning (SharePoint Server 2010) (*http://technet.microsoft.com/library/bc5991d1-c187-495c-8057-3bf2c5653aff(Office.14).aspx*)

Configure Web Server Security (IIS 7) (*http://go.microsoft.com/fwlink/?LinkId=188002*)

Create a Web application that uses Windows-classic authentication (SharePoint Server 2010)

This article describes how to create a Web application that uses Windows-classic authentication.

Tip:
If you want to use Windows-claims authentication instead, see Create a Web application that uses Windows-claims authentication (SharePoint Server 2010).

Before you perform this procedure, confirm that:

- Your system is running Microsoft SharePoint Server 2010.
- You have your logical architecture design in place. For more information, see Logical architecture components (SharePoint Server 2010) (*http://technet.microsoft.com/library/aaed3a01-f4dc-4353-abda-0beced2080b6(Office.14).aspx*).
- You have planned authentication for your Web application. For more information, see Plan authentication methods (SharePoint Server 2010) (*http://technet.microsoft.com/library/40117fda-70a0-4e3d-8cd3-0def768da16c(Office.14).aspx*), Configure Kerberos authentication (SharePoint Server 2010) and Choose security groups (SharePoint Server 2010) (*http://technet.microsoft.com/library/1a45c752-a485-439a-a19b-340a5bd7a6f3(Office.14).aspx*).
- You have selected the service applications that you want to use for your Web application. For more information, see Service application and service management (SharePoint Server 2010).
- If you use Secure Sockets Layer (SSL), you must associate the SSL certificate with the Web application's IIS Web site after the IIS Web site has been created. For more information about setting up SSL, see How to Setup SSL on IIS 7.0 (*http://go.microsoft.com/fwlink/?LinkId=187887*).
- You have read about alternate access mappings.
- If you have User Account Control (UAC) turned on in Windows, and you use Windows PowerShell 2.0 to create a Web application, you must right-click the SharePoint 2010 Management Shell and select **Run as administrator**.

You can create a Web application by using the SharePoint Central Administration Web site or Windows PowerShell. You typically use Central Administration to create a Web application. If you want to automate the task of creating a Web application, which is common in enterprises, use Windows PowerShell. After the procedure is complete, you can create one or several site collections on the Web application that you have created.

To create a Web application that uses Windows-classic authentication by using Central Administration

1. Verify that you have the following administrative credentials:
   - To create a Web application, you must be a member of the Farm Administrators SharePoint group and member of the local Administrator group on the computer running Central Administration.
2. On the Central Administration Home page, in the **Application Management** section, click **Manage web applications**.
3. On the ribbon, click **New**.
4. On the Create New Web Application page, in the **Authentication** section, click **Classic Mode Authentication**.
5. In the **IIS Web Site** section, you can configure the settings for your new Web application by selecting one of the following two options:
   - Click **Use an existing web site**, and then select the Web site on which to install your new Web application.
   - Click **Create a new IIS web site**, and then type the name of the Web site in the **Name** box.
6. In the **IIS Web Site** section, in the **Port** box, type the port number you want to use to access the Web application. If you are creating a new Web site, this field is populated with a random port number. If you are using an existing Web site, this field is populated with the current port number.

   Note:
   The default port number for HTTP access is 80, and the default port number for HTTPS access is 443. If you want users to access the Web application without typing in a port number, they should use the appropriate default port number.
7. Optional: In the **IIS Web Site** section, in the **Host Header** box, type the host name (for example, www.contoso.com) you want to use to access the Web application.

   Note:
   In general, this field is not set unless you want to configure two or more IIS Web sites that share the same port number on the same server, and DNS has been configured to route requests to the same server.
8. In the **IIS Web Site** section, in the **Path** box, type the path to the IIS Web site home directory on the server. If you are creating a new Web site, this field is populated with a suggested path. If you are using an existing Web site, this field is populated with the current path of that Web site.
9. In the **Security Configuration** section, configure authentication and encryption for your Web application.
   a. In the **Authentication Provider** section, click **Negotiate (Kerberos)** or **NTLM**.

Note:

To enable Kerberos authentication, you must perform additional configuration. For more information, see Configure Kerberos authentication (SharePoint Server 2010).

b. In the **Allow Anonymous** section, click **Yes** or **No**. If you choose to allow anonymous access, this enables anonymous access to the Web site by using the computer-specific anonymous access account (that is, IIS_IUSRS).

Note:

If you want users to be able to access any site content anonymously, you must enable anonymous access for the entire Web application zone before you enable anonymous access at the SharePoint site level; later, site owners can configure how anonymous access is used within their sites. If you do not enable anonymous access at the Web application level, you cannot enable anonymous access later, at the site level. For more information, see Choose security groups (SharePoint Server 2010) (*http://technet.microsoft.com/library/1a45c752-a485-439a-a19b-340a5bd7a6f3(Office.14).aspx*).

c. In the **Use Secure Sockets Layer (SSL)** section, click **Yes** or **No**. If you choose to enable SSL for the Web site, you must configure SSL by requesting and installing an SSL certificate. For more information about setting up SSL, see How to Setup SSL on IIS 7.0 (*http://go.microsoft.com/fwlink/?LinkId=187887*).

10. In the **Public URL** section, type the URL for the domain name for all sites that users will access in this Web application. This URL will be used as the base URL in links shown on pages within the Web application. The default URL is the current server name and port, and is automatically updated to reflect the current SSL, host header, and port number settings on the page. If you are deploying SharePoint Server 2010 behind a load balancer or proxy server, then this URL may need to be different than the SSL, host header, and port settings on this page.

    The **Zone** value is automatically set to **Default** for a new Web application.

    Note:

    You can change the zone when you extend a Web application. For more information, see Extend a Web application (SharePoint Server 2010) (*http://technet.microsoft.com/library/02dc86bd-5918-4a01-89e9-d04508c3cc72(Office.14).aspx*).

11. In the **Application Pool** section, do one of the following:

    - Click **Use existing application pool**, and then select the application pool you want to use from the drop-down menu.
    - Click **Create a new application pool**, and then type the name of the new application pool or keep the default name.

For more information, see Logical architecture components (SharePoint Server 2010) (*http://technet.microsoft.com/library/aaed3a01-f4dc-4353-abda-0beced2080b6(Office.14).aspx*).

12. Under **Select a security account for this application pool**, do one of the following:
    - Click **Predefined** to use a predefined security account, and then select the security account from the drop-down menu.
    - Click **Configurable** to specify a new security account to be used for an existing application pool.

    Note:
    You can create a new account by clicking the **Register new managed account** link.
13. In the **Database Name and Authentication** section, choose the database server, database name, and authentication method for your new Web application, as described in the following table.

Item	Action
Database Server	Type the name of the database server and Microsoft SQL Server instance you want to use in the format <*SERVERNAME\instance*>. You can also use the default entry.
Database Name	Type the name of the database, or use the default entry.
Database Authentication	Select the database authentication to use by doing one of the following: • If you want to use Windows authentication, leave this option selected. We recommend this option because Windows authentication automatically encrypts the password when it connects to SQL Server. • If you want to use SQL authentication, click **SQL authentication**. In the **Account** box, type the name of the account you want the Web application to use to authenticate to the SQL Server database, and then type the password in the **Password** box. **Note:** SQL authentication sends the SQL authentication password to the SQL Server unencrypted. We recommend that you only use SQL authentication if you force protocol encryption to the SQL Server of encrypt your network traffic by using IPsec.

14. If you use database mirroring, in the **Failover Server** section, in the **Failover Database Server** box, type the name of a specific failover database server that you want to associate with a

content database.

15. In the **Service Application Connections** section, select the service application connections that will be available to the Web application. In the drop-down menu, click **default** or **custom**. You use the **custom** option to choose the services application connections that you want to use for the Web application.
16. In the **Customer Experience Improvement Program** section, click **Yes** or **No**.
17. Click **OK** to create the new Web application.

To create a Web application that uses Windows-classic authentication by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin. You also need to be a member of the local Administrators group on the computer running Windows PowerShell. In addition, some procedures require membership in the SQL Server fixed server roles **dbcreator** and **securityadmin**.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

```
New-SPWebApplication -Name <Name> -ApplicationPool <ApplicationPool> -
ApplicationPoolAccount <ApplicationPoolAccount> -Port <Port> -URL <URL>
```

Where:

- <*Name*> is the name of the new Web application.
- <*ApplicationPool*> is the name of the application pool.
- <*ApplicationPoolAccount*> is the user account that this application pool will run as.
- <*Port*> is the port on which the Web application will be created in IIS.
- <*URL*> is the public URL for the Web application.

Example

```
New-SPWebApplication -Name "Contoso Internet Site" -ApplicationPool "ContosoAppPool"
-ApplicationPoolAccount (Get-SPManagedAccount "DOMAIN\jdoe") -Port 80 -URL
"http://www.contoso.com"
```

For more information, see New-SPWebApplication (*http://technet.microsoft.com/library/eaeb5bed-81e7-4275-b005-aa7fc465e6d5(Office.14).aspx*).

Note:
We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

See Also

Extend a Web application (SharePoint Server 2010) (*http://technet.microsoft.com/library/02dc86bd-5918-4a01-89e9-d04508c3cc72(Office.14).aspx*)

Create a site collection (SharePoint Server 2010)

Create a Web application that uses Windows-claims authentication (SharePoint Server 2010)

Configure Web Server Security (IIS 7) (*http://go.microsoft.com/fwlink/?LinkId=188002*)

Create a Web application that uses Windows-claims authentication (SharePoint Server 2010)

This article describes how to create a Web application that uses Windows-claims authentication.

Tip:
If you want to use Windows-classic authentication instead, see Create a Web application that uses Windows-classic authentication (SharePoint Server 2010).

Before you perform this procedure, confirm that:

- Your system is running Microsoft SharePoint Server 2010.
- You have your logical architecture design in place. For more information, see Logical architecture components (SharePoint Server 2010) (*http://technet.microsoft.com/library/aaed3a01-f4dc-4353-abda-0beced2080b6(Office.14).aspx*).
- You have planned authentication for your Web application. For more information, see Plan authentication methods (SharePoint Server 2010) (*http://technet.microsoft.com/library/40117fda-70a0-4e3d-8cd3-0def768da16c(Office.14).aspx*), Configure Kerberos authentication (SharePoint Server 2010) and Choose security groups (SharePoint Server 2010) (*http://technet.microsoft.com/library/1a45c752-a485-439a-a19b-340a5bd7a6f3(Office.14).aspx*).
- You have selected the service applications that you want to use for your Web application. For more information, see Service application and service management (SharePoint Server 2010).
- If you use Secure Sockets Layer (SSL), you must associate the SSL certificate with the Web application's IIS Web site after the IIS Web site has been created. For more information about setting up SSL, see How to Setup SSL on IIS 7.0 (*http://go.microsoft.com/fwlink/?LinkId=187887*).
- You have read about alternate access mappings.
- If you have User Account Control (UAC) turned on in Windows, and you use Windows PowerShell 2.0 to create a Web application, you must right-click the SharePoint 2010 Management Shell and select **Run as administrator**.

You can create a Web application by using the SharePoint Central Administration Web site or Windows PowerShell. You typically use Central Administration to create a Web application. If you want to automate the task of creating a Web application, which is common in enterprises, use Windows PowerShell. After the procedure is complete, you can create one or several site collections on the Web application that you have created.

To create a Web application with Windows-claims authentication by using Central Administration

1. Verify that you have the following administrative credentials:
   - To create a Web application, you must be a member of the Farm Administrators SharePoint group and a member of the local Administrators group on the computer running Central Administration.
2. On the Central Administration Home page, in the **Application Management** section, click **Manage web applications**.
3. On the ribbon, click **New**.
4. On the Create New Web Application page, in the **Authentication** section, click **Claims Based Authentication**.
5. In the **IIS Web Site** section, you can configure the settings for your new Web application by selecting one of the following two options:
   - Click **Use an existing web site**, and then select the Web site on which to install your new Web application.
   - Click **Create a new IIS web site**, and then type the name of the Web site in the **Name** box.
6. In the **IIS Web Site** section, in the **Port** box, type the port number you want to use to access the Web application. If you are creating a new Web site, this field is populated with a random port number. If you are using an existing Web site, this field is populated with the current port number.

   Note:
   The default port number for HTTP access is 80, and the default port number for HTTPS access is 443. If you want users to access the Web application without typing in a port number, they should use the appropriate default port number.
7. Optional: In the **IIS Web Site** section, in the **Host Header** box, type the host name (for example, www.contoso.com) you want to use to access the Web application.

   Note:
   In general, this field is not set unless you want to configure two or more IIS Web sites that share the same port number on the same server, and DNS has been configured to route requests to the same server.
8. In the **IIS Web Site** section, in the **Path** box, type the path to the IIS Web site home directory on the server. If you are creating a new Web site, this field is populated with a suggested path. If you are using an existing Web site, this field is populated with the current path of that Web site.
9. In the **Security Configuration** section, choose whether or not to use allow anonymous access and whether or not to use Secure Sockets Layer (SSL).
   a. Under **Allow Anonymous**, click **Yes** or **No**. If you choose to allow anonymous access, this

enables anonymous access to the Web site by using the computer-specific anonymous access account (that is, IIS_IUSRS).

Note:
If you want users to be able to access any site content anonymously, you must enable anonymous access for the entire Web application zone before you enable anonymous access at the SharePoint site level; later, site owners can configure how anonymous access is used within their sites. If you do not enable anonymous access at the Web application level, you cannot enable anonymous access later, at the site level. For more information, see Choose security groups (SharePoint Server 2010) (*http://technet.microsoft.com/library/1a45c752-a485-439a-a19b-340a5bd7a6f3(Office.14).aspx*).

b. Under **Use Secure Sockets Layer (SSL)**, click **Yes** or **No**. If you choose to enable SSL for the Web site, you must configure SSL by requesting and installing an SSL certificate. For more information about setting up SSL, see How to Setup SSL on IIS 7.0 (*http://go.microsoft.com/fwlink/?LinkId=187887*).

10. In the **Claims Authentication Types** section, select the authentication that you want to use for the Web application.

a. If you want to enable Windows authentication, select **Enable Windows Authentication** and, in the drop-down menu, select **Negotiate (Kerberos)** or **NTLM**. For more information, see Configure Kerberos authentication (SharePoint Server 2010).

If you do not want to use Integrated Windows authentication, clear **Integrated Windows authentication**.

If you want users' credentials to be sent over a network in a nonencrypted form, select **Basic authentication (password is sent in clear text)**.

Note:
You can select basic authentication or integrated Windows authentication, or both. If you select both, SharePoint Server 2010 will offer both authentication types to the client Web browser. The client Web browser then determines which type of authentication to use. If you only select basic authentication, ensure that SSL is enabled; otherwise, the credentials can be intercepted by a malicious user.

b. If you want to enable forms-based authentication, select **Enable Forms Based Authentication (FBA)**, and then enter the membership provider name and the role manager name in the boxes.

For more information, see Configure forms-based authentication for a claims-based Web application (SharePoint Server 2010).

Note:
If you select this option, ensure that SSL is enabled; otherwise, the credentials can be intercepted by a malicious user.

c. If you have set up Trusted Identity Provider authentication in Windows PowerShell, the **Trusted Identity provider** check box is selected.

   For more information, see Configure authentication using a SAML security token (SharePoint Server 2010).

You can use one or more claims authentication types. For more information, see Plan authentication methods (SharePoint Server 2010) (*http://technet.microsoft.com/library/40117fda-70a0-4e3d-8cd3-0def768da16c(Office.14).aspx*).

11. In the **Sign In Page URL** section, choose one of the following options to sign into SharePoint Server 2010:
    - Select **Default Sign In Page URL** if you want users to be redirected to a default sign-in Web site for claims-based authentication.
    - Select **Custom Sign In page URL** and then type the sign-in URL if you want users to be redirected to a customized sign-in Web site for claims-based authentication.
12. In the **Public URL** section, type the URL for the domain name for all sites that users will access in this Web application. This URL will be used as the base URL in links shown on pages within the Web application. The default URL is the current server name and port, and is automatically updated to reflect the current SSL, host header, and port number settings on the page. If you are deploying SharePoint Server 2010 behind a load balancer or proxy server, then this URL may need to be different than the SSL, host header, and port settings on this page.

    The **Zone** value is automatically set to **Default** for a new Web application.

    Note:
    You can change the zone when you extend a Web application. For more information, see Extend a Web application (SharePoint Server 2010) (*http://technet.microsoft.com/library/02dc86bd-5918-4a01-89e9-d04508c3cc72(Office.14).aspx*).
13. In the **Application Pool** section, do one of the following:
    - Click **Use existing application pool**, and then select the application pool you want to use from the drop-down menu.
    - Click **Create a new application pool**, and then type the name of the new application pool or keep the default name.

    For more information, see Logical architecture components (SharePoint Server 2010) (*http://technet.microsoft.com/library/aaed3a01-f4dc-4353-abda-0beced2080b6(Office.14).aspx*).
14. Under **Select a security account for this application pool**, do one of the following:
    - Click **Predefined** to use a predefined security account, and then select the security account from the drop-down menu.
    - Click **Configurable** to specify a new security account to be used for an existing application pool.

Note:

You can create a new account by clicking the **Register new managed account** link.

15. In the **Database Name and Authentication** section, choose the database server, database name, and authentication method for your new Web application as described in the following table.

Item	Action
Database Server	Type the name of the database server and Microsoft SQL Server instance you want to use in the format <*SERVERNAME\instance*>. You can also use the default entry.
Database Name	Type the name of the database, or use the default entry.
Database Authentication	Select the database authentication to use by doing one of the following: • If you want to use Windows authentication, leave this option selected. We recommend this option because Windows authentication automatically encrypts the password when it connects to SQL Server. • If you want to use SQL authentication, click **SQL authentication**. In the **Account** box, type the name of the account you want the Web application to use to authenticate to the SQL Server database, and then type the password in the **Password** box. **Note:** SQL authentication sends the SQL authentication password to the SQL Server unencrypted. We recommend that you only use SQL authentication if you force protocol encryption to the SQL Server of encrypt your network traffic by using IPsec.

16. If you use database mirroring, in the **Failover Server** section, in the **Failover Database Server** box, type the name of a specific failover database server that you want to associate with a content database.
17. In the **Service Application Connections** section, select the service application connections that will be available to the Web application. In the drop-down menu, click **default** or **custom**. You use the **custom** option to choose the services application connections that you want to use for the Web application.
18. In the **Customer Experience Improvement Program** section, click **Yes** or **No**.
19. Click **OK** to create the new Web application.

To create a Web application that uses Windows-claims authentication by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin. You also need to be a member of the local Administrators group on the computer running Windows PowerShell. In addition, some procedures require membership in the SQL Server fixed server roles **dbcreator** and **securityadmin**.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. To create a Windows-claims authentication provider, at the Windows PowerShell command prompt, type the following command:

   ```
   $ap = New-SPAuthenticationProvider
   ```

   To create a Web application that uses Windows-claims authentication, at the Windows PowerShell command prompt, type the following command:

   ```
   $wa = New-SPWebApplication -Name <ClaimsWindowsWebApplication> -ApplicationPool
   <ClaimsApplicationPool> -ApplicationPoolAccount <ClaimsApplicationPoolAccount> -
   URL <URL> -Port <Port> -AuthenticationProvider $ap
   ```

   Note:
   We recommend that the application pool account is a managed account on the server farm.

   Where:

   - <*Name*> is the name of the new Web application that uses Windows claims authentication.
   - <*ApplicationPool*> is the name of the application pool.
   - <*ApplicationPoolAccount*> is the user account that this application pool will run as.
   - <*URL*> is the public URL for the Web application.
   - <*Port*> is the port on which the Web application will be created in IIS.

     Example

     ```
     $ap = New-SPAuthenticationProvider
     $wa = New-SPWebApplication -Name "Contoso Internet Site" -ApplicationPool
     "ContosoAppPool" -ApplicationPoolAccount (Get-SPManagedAccount "DOMAIN\jdoe") -URL
     "http://www.contoso.com" -Port 80 -AuthenticationProvider $ap
     ```

For more information, see New-SPWebApplication (*http://technet.microsoft.com/library/eaeb5bed-81e7-4275-b005-aa7fc465e6d5(Office.14).aspx*) and New-SPAuthenticationProvider (*http://technet.microsoft.com/library/c1056674-30b6-4c9c-bfc7-a2d336064b62(Office.14).aspx*).

Note:

We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

See Also

Extend a Web application (SharePoint Server 2010) (*http://technet.microsoft.com/library/02dc86bd-5918-4a01-89e9-d04508c3cc72(Office.14).aspx*)

Create a site collection (SharePoint Server 2010)

Configure forms-based authentication for a claims-based Web application (SharePoint Server 2010)

Configure authentication using a SAML security token (SharePoint Server 2010)

Create a Web application that uses Windows-classic authentication (SharePoint Server 2010)

Configure Web Server Security (IIS 7) (*http://go.microsoft.com/fwlink/?LinkId=188002*)

Configure claims authentication (SharePoint Server 2010)

In this section:

- Create a Web application that uses Windows-claims authentication (SharePoint Server 2010)
- Configure anonymous access for a claims-based Web application (SharePoint Server 2010)
- Configure forms-based authentication for a claims-based Web application (SharePoint Server 2010)
- Configure Kerberos authentication for the claims to Windows token service (SharePoint Server 2010)
- Configure authentication using a SAML security token (SharePoint Server 2010)
- Configure claims-based authentication using Windows Live ID (SharePoint Server 2010)

Configure anonymous access for a claims-based Web application (SharePoint Server 2010)

After you have configured a Microsoft SharePoint Server 2010 claims-based Web application, you can use the procedure in this article to configure anonymous access for your claims-based Web application. For more information, see Create a Web application that uses Windows-claims authentication (SharePoint Server 2010).

Configure anonymous access for a claims-based Web application

To configure anonymous access for a claims-based Web application

1. Verify that the user account that is performing this procedure is a site collection administrator.
2. In Central Administration, go to the **Security** section.
3. Under **Anonymous Access**, select **Enable Anonymous**.
4. Click **Save**.
5. Go to the site for the appropriate Web application.
6. Select **Site Actions**.
7. Select **Site Permissions**.
8. On the ribbon, select **Anonymous Access**.
9. Select either **Entire Web Site** or **Lists and Libraries**, depending on how you want to scope anonymous access for this site.

Configure forms-based authentication for a claims-based Web application (SharePoint Server 2010)

The procedures in this article provide guidance to enable you to configure forms-based authentication for a Microsoft SharePoint Server 2010 claims-based Web application.

Perform the steps in the following procedures to configure a forms-based Web application to use an LDAP provider.

- Configure a forms-based Web application to use an LDAP provider by using Central Administration
- Configure the LDAP Web.Config files
- Configure a forms-based Web application to use an LDAP provider by using Windows PowerShell

Configure a forms-based Web application to use an LDAP provider by using Central Administration

Perform the steps in the following procedure to use Central Administration to configure forms-based authentication for a claims-based Web application.

To configure forms-based authentication for a claims-based Web application by using Central Administration

1. Verify that the user account that is performing this procedure is a site collection administrator.
2. In Central Administration, in the **Application Management** section, click **Manage web applications**.
3. In the **Contribute** group of the ribbon, click **New**.
4. In the **Authentication** section of the **Create New Web Application** dialog box, click **Claims Based Authentication**.
5. In the **Claims Authentication Types** section, select **Enable Forms Based Authentication (FBA)**.
6. Type a membership provider name and a role manager name. In the example Web.Config file depicted in this article, the name of the membership provider is **membership**, and the name of the role manager is **rolemanager**.
7. Click **OK** to create the Web application.

Configure the LDAP Web.Config files

After you have successfully created the Web application (described in the preceding procedure), modify the following Web.Config files:

- The Central Administration Web application Web.Config file
- The Security Token Service Web.Config file
- The forms-based authentication claims-based Web application Web.Config file

To configure the Central Administration Web.Config file

1. Start IIS Manager by typing **INETMGR** at a command prompt.
2. Go to the **SharePoint Central Administration** site in IIS.
3. Right-click **SharePoint Central Administration** and then click **Explore**.
4. Open the Web.Config file.
5. Find the `<Configuration> <system.web>` section and add the following entry:

```
<membership defaultProvider="AspNetSqlMembershipProvider">
      <providers>
        <add name="membership"
             type="Microsoft.Office.Server.Security.LdapMembershipProvider,
Microsoft.Office.Server, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
             server="yourserver.com"
             port="389"
             useSSL="false"
             userDNAttribute="distinguishedName"
             userNameAttribute="sAMAccountName"
             userContainer="OU=UserAccounts,DC=internal,DC=yourcompany,DC= distinguishedName
(of your userContainer)"
             userObjectClass="person"
             userFilter="(ObjectClass=person)"
             scope="Subtree"
             otherRequiredUserAttributes="sn,givenname,cn" />
      </providers>
    </membership>
    <roleManager enabled="true" defaultProvider="AspNetWindowsTokenRoleProvider" >
      <providers>
        <add name="roleManager"
```

```
            type="Microsoft.Office.Server.Security.LdapRoleProvider, Microsoft.Office.Server,
Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
            server="yourserver.com"
            port="389"
            useSSL="false"
            groupContainer="DC=internal,DC=yourcompany,DC= distinguishedName (of your
groupContainer)"
            groupNameAttribute="cn"
            groupNameAlternateSearchAttribute="samAccountName"
            groupMemberAttribute="member"
            userNameAttribute="sAMAccountName"
            dnAttribute="distinguishedName"
            groupFilter="((ObjectClass=group)"
            userFilter="((ObjectClass=person)"
            scope="Subtree" />
      </providers>
 </roleManager>
```

Important:
After you have added the preceding entry, save and close the Web.Config file.

To configure the Security Token Service Web.Config file

1. Start IIS Manager by typing **INETMGR** at a command prompt.
2. Go to the **SharePoint Web Services** site.
3. Go to the **SecurityTokenServiceAppliction** sub-site.
4. Right-click **SecurityTokenServiceAppliction** and then click **Explore**.
5. Open the Web.Config file.
6. Find the `<Configuration> <system.web>` section and add the following entry:

```
<membership>
      <providers>
        <add name="membership"
            type="Microsoft.Office.Server.Security.LdapMembershipProvider,
Microsoft.Office.Server, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
            server="yourserver.com"
            port="389"
```

```
            useSSL="false"
            userDNAttribute="distinguishedName"
            userNameAttribute="sAMAccountName"
            userContainer="OU=UserAccounts,DC=internal,DC=yourcompany,DC=com"
            userObjectClass="person"
            userFilter="(&amp;(ObjectClass=person))"
            scope="Subtree"
            otherRequiredUserAttributes="sn,givenname,cn" />
      </providers>
    </membership>
    <roleManager enabled="true" >
      <providers>
        <add name="rolemanager"
            type="Microsoft.Office.Server.Security.LdapRoleProvider, Microsoft.Office.Server,
Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
            server="yourserver.com"
            port="389"
            useSSL="false"
            groupContainer="DC=internal,DC=yourcompany,DC=com"
            groupNameAttribute="cn"
            groupNameAlternateSearchAttribute="samAccountName"
            groupMemberAttribute="member"
            userNameAttribute="sAMAccountName"
            dnAttribute="distinguishedName"
            groupFilter="(&amp;(ObjectClass=group))"
            userFilter="(&amp;(ObjectClass=person))"
            scope="Subtree" />
      </providers>
    </roleManager>
```

Important:
After you have added the preceding entry, save and close the Web.Config file.

To configure the forms-based authentication claims-based Web application Web.Config file

1. Start IIS Manager by typing **INETMGR** at a command prompt.
2. Go to the **Claims Forms** site.
3. Right-click **Claims Forms** and then click **Explore**.
4. Open the Web.Config file.
5. Find the `<Configuration> <system.web>` section.
6. Find the `<membership defaultProvider="i">` section and add the following entry:

```
<add name="membership"
             type="Microsoft.Office.Server.Security.LdapMembershipProvider,
Microsoft.Office.Server, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
             server="yourserver.com"
             port="389"
             useSSL="false"
             userDNAttribute="distinguishedName"
             userNameAttribute="sAMAccountName"
             userContainer="OU=UserAccounts,DC=internal, DC=yourcompany,DC=com"
             userObjectClass="person"
             userFilter="(&amp;(ObjectClass=person))"
             scope="Subtree"
             otherRequiredUserAttributes="sn,givenname,cn" />
```

Find the `<roleManager defaultProvider="c" enabled="true" cacheRolesInCookie="false">` section and add the following entry:

```
<add name="roleManager"
             type="Microsoft.Office.Server.Security.LdapRoleProvider, Microsoft.Office.Server,
Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
             server="yourserver.com"
             port="389"
             useSSL="false"
             groupContainer="DC=internal,DC=yourcompany,DC=com"
             groupNameAttribute="cn"
             groupNameAlternateSearchAttribute="samAccountName"
             groupMemberAttribute="member"
             userNameAttribute="sAMAccountName"
```

```
dnAttribute="distinguishedName"
groupFilter="(&amp;(ObjectClass=group))"
userFilter="(&amp;(ObjectClass=person))"
scope="Subtree" />
```

Important:
After you have added the preceding entry, save and close the Web.Config file.

Warning:
Do not overwrite any existing entries in this Web.Config file.

Configure a forms-based Web application to use an LDAP provider by using Windows PowerShell

Perform the steps in the following procedure to use Windows PowerShell to configure forms-based authentication for a claims-based Web application.

To configure a forms-based Web application to use an LDAP provider by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt, type the following:

```
$ap = New-SPAuthenticationProvider -Name "ClaimsForms" -ASPNETMembershipProvider
"membership" -ASPNETRoleProviderName "rolemanager"
$wa = New-SPWebApplication -Name "Claims Windows Web App" -ApplicationPool "Claims
App Pool" -ApplicationPoolAccount "internal\appool"
  -Url http://servername -Port 80 -AuthenticationProvider $ap
```

   Note:
   The value of the **ApplicationPoolAccount** parameter must be a managed account on the farm.

6. After you have successfully created an authentication provider and a Web application, modify the following Web.Config files by using the sample entries provided in the Configure the LDAP Web.Config files section of this article:
   - To configure the Central Administration Web.Config file
   - To configure the Security Token Service Web.Config file

- To configure the forms-based authentication claims-based Web application Web.Config file

7. After you have modified the Web.Config files, create a SPClaimsPrincipal and a site collection, as shown in the following example:

```
$cp = New-SPClaimsPrincipal -Identity "membership:SiteOwner" -IdentityType
FormsUser

$sp = New-SPSite http://servername:port -OwnerAlias $cp.Encode() -Template "STS#0"
```

   For more information, see New-SPClaimsPrincipal (*http://technet.microsoft.com/library/0831e64b-3ec0-4016-8128-639991530172(Office.14).aspx*).

Note:
We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

See Also

Migrate from forms-based authentication to claims-based authentication (SharePoint Server 2010) (*http://technet.microsoft.com/library/95a7fcfc-0b5e-46eb-acd4-2b1a910131c4(Office.14).aspx*)

Configure Kerberos authentication for the claims to Windows token service (SharePoint Server 2010)

This article is obsolete.

For information about how to configure Kerberos authentication, see Configuring Kerberos Authentication for Microsoft SharePoint 2010 Products and Technologies (white paper). (*http://go.microsoft.com/fwlink/?LinkID=197178*) white paper.

See Also

Plan authentication methods (SharePoint Server 2010) (*http://technet.microsoft.com/library/40117fda-70a0-4e3d-8cd3-0def768da16c(Office.14).aspx*)

Configure authentication using a SAML security token (SharePoint Server 2010)

The procedures in this article provide explain how to configure authentication using a Security Assertion Markup Language (SAML) security token for a Microsoft SharePoint Server 2010 claims-based Web application.

SAML sign-in is typically used in enterprise federation scenarios, for example, to provide access to a business partner. SAML sign-in is also deployed to provide access to internal users whose accounts reside in a domain that is not part of the forest that contains SharePoint Server 2010.

Before you configure authentication using a SAML security token for a SharePoint Server 2010 claims-based Web application, you must configure a server running Active Directory Federation Services (AD FS) 2.0. For information about configuring a server to run AD FS 2.0, see the AD FS 2.0 Deployment Guide (*http://go.microsoft.com/fwlink/?LinkId=191723*).

In this article:

- Configure an Identity Provider STS (IP-STS) Web application by using Windows PowerShell
- Configure a Relying Party STS (RP-STS) Web application
- Establish a trust relationship with an Identity Provider STS (IP-STS) by using Windows PowerShell
- Export the trusted IP-STS certificate by using Windows PowerShell
- Define a unique identifier for claims mapping by using Windows PowerShell
- Create a new SharePoint Web application and configure it to use SAML sign-in

Configure an Identity Provider STS (IP-STS) Web application by using Windows PowerShell

Perform the following procedures to use Windows PowerShell to configure a SharePoint claims-based Web application.

To configure an Identity Provider STS (IP-STS) Web application by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.

5. From the Windows PowerShell command prompt, create an x509Certificate2 object, as shown in the following example:

```
$cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2("path to cert
file")
```

6. Create a claim type mapping to use in your trusted authentication provider, as shown in the following example:

```
New-SPClaimTypeMapping
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
-IncomingClaimTypeDisplayName "EmailAddress" -SameAsIncoming
```

7. Create a trusted login provider by first creating a value for the `realm` parameter, as shown in the following example:

```
$realm = "urn:" + $env:ComputerName + ":domain-int"
```

8. Create a value for the `signinurl` parameter that points to the Security Token Service Web application, as shown in the following example:

```
$signinurl = "https://test-2/FederationPassive/"
```

9. Create the trusted login provider, using the same `IdentifierClaim` value as in a claim mapping (`$map1.InputClaimType`), as shown in the following example:

```
$ap = New-SPTrustedIdentityTokenIssuer -Name
"WIF" -Description "Windows® Identity Foundation" -Realm
$realm -ImportTrustCertificate $cert
-ClaimsMappings $map1[,$map2..] -SignInUrl
$signinurl -IdentifierClaim $map1.InputClaimType
```

10. Create a Web application by first creating a value for the application pool account (for the current user), as shown in the following example:

```
$account = "DOMAIN\" + $env:UserName
```

Note:
The application pool account must be a managed account. To create a managed account, use `New-SPManagedAccount`.

11. Create a value for the Web application URL (`$webappurl = "https://" + $env:ComputerName`), as shown in the following example:

```
$wa = New-SPWebApplication -name "Claims WIF"
-SecureSocketsLayer -ApplicationPool "SharePoint SSL"
-ApplicationPoolAccount $account -Url $webappurl -Port 443
-AuthenticationProvider $ap
```

12. Create a site by first creating a claim object, as shown in the following example:

```
$claim = New-SPClaimsPrincipal

-TrustedIdentityTokenIssuerr $ap -Identity

$env:UserName
```

13. Create a site, as shown in the following example:

```
$site = New-SPSite $webappurl -OwnerAlias

$claim.ToEncodedString() -template "STS#0"
```

Configure a Relying Party STS (RP-STS) Web application

Use the procedure in this section to configure a relying-party STS Web application.

To configure a Relying Party STS (RP-STS) Web application

1. Open the Active Directory Federation Services (AD FS) 2.0 Management console.
2. In the left pane, expand **Policy**, and select **Relying Parties**.
3. In the right pane, click **Add Relying Party**. This opens the Active Directory Federation Services (AD FS) 2.0 configuration wizard.
4. On the first page of the wizard, click **Start**.
5. Select **Enter relying party configuration manually**, and click **Next**.
6. Type a relying party name and click **Next**.
7. Make sure **Active Directory Federation Services (AD FS) 2.0 Server Profile** is selected, and click **Next**.
8. Do not use an encryption certificate. Click **Next**.
9. Select **Enable support for Web-browser-based identity federation**.
10. Type the name of the Web application URL, and append **/_trust/** (for example: **https://servername/_trust/**). Click **Next**.
11. Type the name of an identifier (for example: **urn:COMPUTERNAME:Geneva**), and click **Add**. Click **Next**.
12. On the Summary page, click **Next** and then click **Close**. This opens the Rules Editor Management console. Use this console to configure the mapping of claims from an LDAP Web application to SharePoint.
13. In the left pane, expand **New Rule**, and select **Predefined Rule**.
14. Select **Create Claims from LDAP Attribute Store**.
15. In the right pane, from the **Attribute Store** drop-down list, select **Enterprise Active Directory User Account Store**.

16. Under **LDAP Attribute**, select **sAMAccountName**.
17. Under **Outgoing Claim Type**, select **E-Mail Address**.
18. In the left pane, click **Save**.

Establish a trust relationship with an Identity Provider STS (IP-STS) by using Windows PowerShell

Use the procedure in this section to establish a trust relationship with an IP-STS.

To establish a trust relationship with an IP-STS by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt, establish a trust relationship, as shown in the following example:

```
$waurl = "https://" + $env:ComputerName

$title = "SAML-Claims"
```

Export the trusted IP-STS certificate by using Windows PowerShell

Use the procedure in this section to export the certificate of the IP-STS with which you want to establish a trust relationship, and then copy the certificate to a location that Microsoft SharePoint Server 2010 can access.

To export the trusted IP-STS certificate by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt, export the trusted IP-STS certificate, as shown in the following example:

```
$cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2("c:\geneva.cer")
```

Define a unique identifier for claims mapping by using Windows PowerShell

Use the procedure in this section to define an e-mail address that will serve as a unique identifier for claims mapping. Typically, the administrator of the trusted STS will have to provide this information because only the owner of the STS knows which value in the token will be always unique for each user. Note that the administrator of the trusted STS can create a URI to represent the e-mail address.

To define a unique identifier for claims mapping by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt, create a mapping, as shown in the following example:

```
$map = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress" -
IncomingClaimTypeDisplayName "EmailAddress" -SameAsIncoming
```

Create a new authentication provider

Use the procedure in this section to create a new authentication provider that the Web application will use.

To create a new authentication provider by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt, create a new authentication provider, as shown in the following example. Note that the realm is the parameter used by the trusted STS to identify a specific SharePoint farm.

```
$realm = "urn:" + $env:ComputerName + ":Geneva"

$ap = New-SPTrustedIdentityTokenIssuer -Name "Geneva" -Description "Geneva" -Realm
$realm -ImportTrustCertificate $cert -ClaimsMappings $map -SignInUrl "https://
test-2/FederationPassive/" -IdentifierClaim
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
```

Create a new SharePoint Web application and configure it to use SAML sign-in

In this step, you create and configure the Web application.

To create a new SharePoint Web application and configure it to use SAML sign-in by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt, create a new SharePoint Web application and configure it to use SAML sign-in. Note that you must replace *"WebAppUrl"* and *"domain\admin"* with the valid values.

```
$wa = New-SPWebApplication -Name "SAML Sign-In" -SecureSocketsLayer -
ApplicationPool "SAML Sign-In" -ApplicationPoolAccount "domain\admin" -
Url "WebAppUrl" -Port 443 -AuthenticationProvider $ap
```

Note:
You are enabling SSL, because with SAML sign-in, cookies are used as the single sign-on ticket for the user. This allows administrators to grant access to the SharePoint resources for the duration of the token without the need of re-authenticate the user. Without SSL, these cookies can be easily hijacked by a malicious user and be used to impersonate the original user.

When you have completed these procedures, create a SharePoint site and assign an owner. For information about creating a SharePoint site, see Create a site collection (SharePoint Server 2010).

Configure claims-based authentication using Windows Live ID (SharePoint Server 2010)

Claims-based authentication in Microsoft SharePoint Server 2010 can delegate authentication to the Windows Live ID Security Token Service (STS). This is important if you want to implement a scenario in which you use Windows Live ID for password management. The Windows Live ID service is configured as the identity provider for SharePoint Server 2010. A one-way, certificate-based trust relationship is established between SharePoint Server 2010 and the Windows Live ID service. When a user provides Windows Live ID credentials, the Windows Live ID service returns a Passport Unique Identity (PUID) and e-mail information encapsulated in a Security Assertion Markup Language (SAML) version 1.1 claims token. The Windows Live ID public key, which is part of Windows Live ID Metadata XML, encrypts this claims token.

For more information about Windows Live ID, refer to the following resources:

- Introduction to Windows Live ID (*http://go.microsoft.com/fwlink/?LinkId=201477*)
- Microsoft Federation Gateway (*http://go.microsoft.com/fwlink/?LinkId=150843*)
- Windows Live Developer Center (*http://go.microsoft.com/fwlink/?LinkId=191075*)

The Windows Live ID cookie is cached on the client computer and sent to SharePoint Server 2010 by way of a POST response to a successful authentication request. SharePoint Server 2010 converts the Windows Live ID SAML token to a SharePoint Server 2010 SAML token. The PUID for the user is generated based on the user principal name (UPN) claim returned in the SAML token. This value is used throughout SharePoint Server 2010 to uniquely identify the user and perform access control. SharePoint Server 2010 can augment user tokens with additional claims by using a custom claims provider, which is configured in the SharePoint Server 2010 Web application. The SharePoint Server 2010 cookie is also returned to the client computer and cached for subsequent requests. When the Windows Live ID or SharePoint Server 2010 cookie expires, the user is redirected to a Windows Live ID server.

In this article:

Configure the Window Live ID Security Token Service

Configure SharePoint for Window Live ID authentication

Convert a Window Live ID internal environment to a production environment

Create different types of SharePoint claims-based Web applications

Grant permissions to all Window Live ID authenticated users

Configure the Window Live ID Security Token Service

The WS-Federation protocol is implemented by the Windows Live ID service, and provides the infrastructure of the Live ID STS that is designated as a trusted identity provider. You can extract a Windows Live ID public certificate from a metadata XML `X509Certificate` node and save it to an Internet security certificate with a .cer file extension. If the metadata XML contains multiple `X509Certificate` nodes, you can use any of them. Provide read access to the SharePoint Server 2010 farm application pool account in Internet security certificate (.cer file).

Configure the Microsoft Services Manager (MSM) using the following values:

Value	Description
Domain Name	The domain name for which authentication requests to the Live ID STS will be generated. Use a Fully Qualified Domain Name (FQDN).
Default Return URL	The URL that the Windows Live ID STS will redirect a user to after successful authentication, for example: **https://username.global.corp.contoso.com/_trust/default.aspx**.
DNS Name	The unique identifier provided in an authentication request to the Windows Live ID STS. This unique identifier enables look-up functionality for the Default Return URL. The DNS Name must correspond to the realm value specified in Windows Live ID authentication request.
WRealm Parameter	The WRealm parameter must match the DNS field in the MSM Site configuration. The WRealm parameter must be created by using one of the following formats: **sub.domain.top** or **Urn:domain:name**.
Override Authentication Policy	Configure the Override Authentication Policy by using the following value: MBI_FED_SSL.

Configure SharePoint for Window Live ID authentication

Use the procedures in this section to configure SharePoint Server 2010 for Windows Live ID authentication.

To configure SharePoint for Windows Live ID authentication by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.

4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt (that is, PS C:\>), define the realm value to match the DNS Name value specified in the Microsoft Services Manager. The realm value in Windows Live ID integration should correspond to the correct DNS name, as shown in the following example:

```
$realm = "urn:" + $env:ComputerName + ":ServerName"
```

6. Get the PUID value of the account that you will use as the Farm Administrator account by first signing in to following Web site: Windows Live ID(*https://accountservices.passport-int.net/?ru=https://accountservices.passport-int.net/Credentials.srf%3Fvv%3D750%26mkt%3DEN-US%26lc%3D1033&vv=750&mkt=EN-US&lc=1033&id=10*), and then locating the `Unique ID` field on the Credentials page.
7. Specify the PUID value using the following format: **PUID**@live.com.
8. Locate one of the `<X509Certificate>` nodes in the following source: Metadata XML URL (*https://nexus.passport-int.com/federationmetadata2/2007-06/federationmetadata.xml*).
9. Copy the contents of either of the two `X509Certificate` nodes, as shown in the following example:

```
MIICWzCCAcSgAwIBAgIJAJEzHoaEodSoMA0GCSqGSIb3DQEBBQUAMCkxJzAlBgNV
BAMTHkxpdmUgSUQgU1RTIFNpZ25pbmcgUHVibGljIEtleTAeFw0wODEwMzAyMjA5
MjNaFw0xMzEwMjkyMjA5MjNaMCkxJzAlBgNVBAMTHkxpdmUgSUQgU1RTIFNpZ25p
bmcgUHVibGljIEtleTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEArz97XPae
GNAC4UnKl5zReyhgk3Bzf08U+CgD0R9+GZOahmpakJXFpI213gQWiHrUGaMN9nsK
4kzSfDPiquAMsV6vBYyWuPLZ0XrMzTAOV/WHSK3bCsYWWQZeH9Xn8G1Hkz+gQSC/
92lBbq9oBCZfLv3OlkobOmT8d+ldRKGU4pUCAwEAAaOBijCBhzAdBgNVHQ4EFgQU
VbJyIcGL0AjB4/Wm4DqUZux6uUkwWQYDVR0jBFIwUIAUVbJyIcGL0AjB4/Wm4DqU
Zux6uUmhLaQrMCkxJzAlBgNVBAMTHkxpdmUgSUQgU1RTIFNpZ25pbmcgUHVibGlj
IEtleYIJAJEzHoaEodSoMAsGA1UdDwQEAwIBxjANBgkqhkiG9w0BAQUFAAOBgQAO
/5vGfu+Vg1TKBuxsAIMqjqKXX7aRrANNZM/5ACdwAUtMDG/n8INoXgOKr851fbF6
4yBesmFjg2TbR8y0/ITAD+d+iyEpR7IO3/is9rWAj4ggbw8yqaDWn26eh3bAdoa+
p38qtqJHkUGF5vApeHiu6zO573bKs+nXcKVM8mNbjA==
```

10. Paste the contents of either `X509Certificate` node into a new Notepad file and save the Notepad file with the following file name: **LiveID-INT.cer**.
11. Configure the Windows Live ID certificate (extracted from metadata XML), as shown in the following example:

```
$certloc = "C:\LiveIDWithSAML\LiveID-INT.cer"
```

12. Define a new trusted root authority in SharePoint Server 2010, as shown in the following example:

```
$rootcert = Get-PfxCertificate $certloc

New-SPTrustedRootAuthority "NewRootAuthority" -Certificate $rootcert | Out-Null
```

13. Create an object with a Windows Live ID certificate, as shown in the following example:

```
$cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2($certloc)
```

14. Define the claim you will use as the unique identifier of the user. Map the UPN claim to the reserved claim name Identifier. The e-mail Address claim can also be mapped, as shown in the following example:

```
$map1 = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.xmlsoap.org/claims/EmailAddress" -IncomingClaimTypeDisplayName
"http://schemas.xmlsoap.org/claims/EmailAddress" -SameAsIncoming

$map2 = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier" -
IncomingClaimTypeDisplayName "UPN" -LocalClaimType
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"
```

15. Create a new SharePoint Server 2010 authentication provider for a new Web application, as shown in the following example:

```
$apSAML = New-SPTrustedIdentityTokenIssuer -Name "LiveID" -Description "LiveID" -
Realm $realm -ImportTrustCertificate $cert -ClaimsMappings $map1,$map2 -SignInUrl
"https://login.live-int.com/login.srf" -IdentifierClaim
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier"
```

16. Create a new SharePoint Server 2010 Web application to use with the authentication provider created in the previous step, as shown in the following example:

```
$waurl = https://" + $env:ComputerName - You might use FQDN url of your site here.

$title = "Site Title"

$waexe = New-SPWebApplication -Name $title -ApplicationPool $title -
ApplicationPoolAccount $owner -Url $waurl -AuthenticationProvider

$scexe = New-SPSite $siteurl -Name $title -Description $title -Template 'STS#1' -
OwnerAlias
```

17. Start IIS Manager by typing **INETMGR** at a command prompt.
18. Go to the **Claims Web Application** site in IIS.
19. In the left pane, right-click **Claims Web Application**, and select **Edit Bindings**.
20. Select **https** and click **Edit**.
21. Under **SSL Certificate**, select any listed certificate. Consider using a self-signed certificate.

22. Import the Windows Live ID public certificate to the **Local computer**, SharePoint Server 2010, and **Trusted People** folders.

Convert a Window Live ID internal environment to a production environment

Use the procedures in this section to convert a Windows Live ID internal environment to a production environment.

To convert a Windows Live ID internal environment to a production environment

1. Make sure the site is migrated to a production environment in MSM, and that compliance is complete. A compliance review is not required if the Windows Live ID environment in MSM is internal.
2. Make sure that the authentication policy of the Windows Live ID production environment is configured with the following value: MBI_FED_SSL.
3. Make sure that the Windows Live ID production environment uses HTTPS-based URLs because the production environment authentication policy is configured for SSL transport. The production environment sites send POST requests over SSL to **https://login.live.com**. In the **SPTrustedIdentityTokenIssuer** there is a Provider URI that should be the live login URI. Make sure the live logon URI is HTTPS-based.
4. If the Windows Live ID claims provider is configured to use an e-mail address instead of a PUID, the production environment site should be in the Microsoft policy group. Be aware that this policy group is auto-approved for internal partners, and explicit approval is required for external partners.

Create different types of SharePoint claims-based Web applications

Use the procedures in this section to run a Windows PowerShell script to create different types of SharePoint Server 2010 claims-based Web applications.

To create different types of SharePoint claims-based Web applications by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt, run the **DeployLiveIdWithSAML** script, as

shown in the following example:

```
#.SYNOPSIS
#    Script for creating different types of claims web applications from the Windows PowerShell command line.
#.DESCRIPTION
#    Script will create ANON, WIN, FBA, MULTI, MIXED, SAML and combinations of these web applications.
#.NOTES
#    Script: ClaimsWA.ps1
#    Remark: The script will load/unload additional snap-ins depending on where it's being executed from.
#    Update: 1/15/2010 (v2.0)
#.PARAMETER type
#   Indicates the type of claims web app to create (see examples for full list of valid supported types)
#If not specified, this will default to ALL and each of the supported types of claims web apps will be created
#.PARAMETER port
#   Indicates the port number to create the web app on (See reserved ports at http://support.microsoft.com/kb/832017)
#If not specified, this will default to port 201 and will be incremented in sequence for multiple web apps
#.PARAMETER owner
#   Indicates the domain account that will be used for App Pool (should be registered as a SharePoint Server managed account)
#If not specified, this will default to logged on user and will use USERDOMAIN & USERNAME environment values
#.EXAMPLE
#   claimswa.ps1 WIN (create WIN-claims web app at port# 201 and use logged on user for app pool account)
#
#   Here are some more examples of HOWTO use the script:
#      claimswa.ps1 ANON (create ANON web app at port# 201)
#      claimswa.ps1 ANON/FBA 701 (create ANON/FBA web app at port# 701)
```

```
#      claimswa.ps1 FBA (create FBA web app at port# 201 using LDAP provider;
default is REDMOND instance)
#      claimswa.ps1 FBA/IBM (create FBA web app at port# 201 using LDAP provider
pointing to the IBM instance)
#      claimswa.ps1 FBA/SQL 851 (create forms-based authentication web app at
port# 851 using SQL provider)
#      claimswa.ps1 WIN/FBA/MIXED 501 (create Windows/forms-based authentication
mixed-mode web apps at port# 501)
#      claimswa.ps1 WIN/SAML/MULTI 901 (create Windows/SAML multi-auth web apps at
port# 901)
#
#   Here is the full list of all the support TYPEs (combine options delimited with
slash for your config):
#
#   Basic auth types:
#      WIN   : create Windows claims web application on the port# specified on
command line
#      FBA   : create forms-based authentication claims web apps with the
specified membership provider (SQL Server/LDAP listed below)
#      SAML  : create SAML-claims web application on the default HTTPS port# 443
#      ANON  : indicator switch for creating the web application to allow ANON
mode
#   Complex auth types:
#      MULTI : create claims web application with multiple auth types using a
single URL to access
#      MIXED : create claims web application with multiple auth types using
multiple URLs to access
#   FBA membership/rolemanager providers
#      RED   : use the REDMOND domain LDAP provider; this is the default setting
if a provider is not specified
#      SQL   : use the SQL Server provider for connecting to forms-based
authentication web apps (connects to the ASPNETDB instance on ZADANG)
#      PPL   : use the PEOPLEDC domain LDAP provider that is a private domain used
for testing PEOPLE features
#      SUN   : use the SUNOne LDAP provider in the PEOPLEDC domain which is used
```

```
for profile import/sync testing
#      IBM   : use the IBM LDAP provider in the PEOPLEDC domain which is used for
profile import/sync testing
#      NVL   : use the Novell LDAP provider in the PEOPLEDC domain which is used
for profile import/sync testing

# TODO (no specific ETA for these updates):
#    1. Set the default IIS cert bindings for SAML web
#    2. Use IIS CMDlets instead of updating XML object
#    3. We should be able to define MixedMode base auth
#    4. Use the domain for logged on user for LDAP string
#    5. Do not attempt to write to CA/STS if running on WFE


# Define the args list that we will accept & work with
param ([string]$type, [int]$port, [string]$owner)

function main() {
    # Valid options list
    $auths  = @("WIN", "FBA", "SAML", "ANON")
    $extnd  = @("MULTI", "MIXED")
    $provs  = @("SQL", "RED", "PPL", "SUN", "IBM", "NVL")
    $optns  = @("APP", "FIX")
    $typeOK = $true

    # Do we have the minimum args data before we can proceed
    # I'm not doing extensive validation but at least minimum
    foreach ($arg in $type.split("/")) {
        if (($auths+$extnd+$optns+$provs) -notcontains $arg) {
            write-host -Fore Red "`nInvalid TYPE argument was specified; execution
aborted!`nTo see a list of valid TYPEs, execute with -examples option`n"
            $typeOK=$false; break
        }
```

```
    }

    if ($typeOK) {
        $type = @($type.toupper().split("/") | Sort | Get-Unique)
        switch ($type.count) {
            1 {
                foreach ($arg in $type) {
                    if (($auths+$extnd+$optns) -notcontains $arg) {
                        write-host -Fore Red "`nInvalid AUTH argument was
specified; execution aborted!`nTo see a list of valid AUTHs, execute with -
examples option`n"
                        $typeOK=$false; break
                    }
                }
                if (($type -eq "MULTI") -or ($type -eq "MIXED")) {
                    $type += @("WIN", "FBA"); write-host -Fore Yellow "MULTI/MIXED
auth combo not specified; defaulting to $type"
                }
                if ($type -eq "ANON") {
                    $type += @("WIN"); write-host -Fore Yellow "ANON auth combo
not specified; defaulting to $type"
                }
            }

            2 {
                if ($type -contains "ANON") {
                    foreach ($arg in $type) {
                        if ($auths -notcontains $arg) {
                            write-host -Fore Red "`nInvalid ANON combo was
specified; execution aborted!`nTo see a list of valid PROVIDERs, execute with -
examples option`n"
                            $typeOK=$false; break
                        }
                    }
```

```
                }
                else {
                    $multiOK=$true
                    foreach ($arg in $type) {
                        if ($auth -notcontains $arg) {
                            $multiOK=$false; break
                        }
                    }
                    if ($multiOK) {$type += @("MULTI"); write-host -Fore Yellow
"Multiple auth types specified; defaulting to $type"}
                }
            }
        }

        if (($type -contains "MULTI") -or ($type -contains "MIXED") -and
($type.count -lt 3)) {
            write-host -Fore Red "`nMULTI/MIXED option requires 2 base auth types
be specified!`nTo see a list of valid TYPEs, execute with -examples option`n"
            $typeOK=$false
        }
    }

    if ($typeOK) {
        # We seem to have the TYPE argument, let's check the others

        if (-not $port) {
            if ($type -contains "SAML") {$port=443} else {$port=201}
            write-host -Fore Yellow "PORT not specified; defaulting to $port"
        }

        if (-not $owner) {
            $owner = $env:UserDomain + "\" + $env:UserName.tolower()
            write-host -Fore Yellow "OWNER not specified; defaulting to $owner"
```

```
        }

        #In case somebody attempts to execute this script in the regular PS/ISE
console,
        #let's load the IIS/SP snap-in to ensure we have everything we need to
work with
        Manage-SnapIns (1)

        # check what flavor of SERVER we're running
        $product = Get-SPProduct | Where-Object
{$_.ProductName.contains("SharePoint Server 2010")};
        if ($product.ProductName.contains("Debug")) {$flavor="DEBUG"} else
{$flavor="SHIP"}
        write-host -Fore Green "Detected $flavor flavor of MOSS installed on this
farm!"

        if ($type -contains "APP") {
            Write-WEBConfigs 0 "APP"
        }
        elseif ($type -contains "FIX") {
            Fix-Environment
        }
        else {
            Create-WebApp $type $port
        }

        # We're done with the snap-ins, so let's unload them
        Manage-SnapIns (0)
    }
}

function Fix-Environment {
    # This is just a series of steps to clean up
```

```
    # Not recommended to use unless you know why!
    Remove-SPTrustedRootAuthority NewRootAuthority
    Remove-SPTrustedIdentityTokenIssuer ServerName

    # I need to add the other clean up stuff here...
}

# This is the core script block that creates the different web apps
function Create-WebApp ([string]$type, [int]$port) {
    $waurl = http://" + $env:ComputerName

    if ($type.contains("SAML")) { $waurl = $waurl.replace("http", "https") }
    $siteurl = $waurl + ":" + $port
    $title = "ClaimsWA-$port-" + $type.replace(" ","-")

    # Let's construct the WA/SC CMDlet call that we'll invoke later
    $waexe = "New-SPWebApplication -Name $title -ApplicationPool $title -
ApplicationPoolAccount $owner -Url $waurl -AuthenticationProvider"
    $scexe = "New-SPSite $siteurl -Name $title -Description $title -Template
'STS#1' -OwnerAlias"

    write-host -Fore Cyan "`nSetting up $title on port $port now:"

    if ($type.contains("WIN")) {
        $apWIN = New-SPAuthenticationProvider -DisableKerberos:$true
        $cpWIN = New-SPClaimsPrincipal -Identity $owner -IdentityType 1
    }

    if ($type.contains("FBA")) {
        if ($type.contains("SQL")) {
            $membership="SQLms"; $rolemanager="SQLrm"; $identity = "sqlms:user1"
        }
        elseif ($type.contains("PPL")) {
```

```
            $membership="PPLms"; $rolemanager="PPLrm"; $identity =
"pplms:fbauser1"
        }
        elseif ($type.contains("SUN")) {
            $membership="SUNms"; $rolemanager="SUNrm"; $identity =
"sunms:fbauser1"
        }
        elseif ($type.contains("IBM")) {
            $membership="IBMms"; $rolemanager="IBMrm"; $identity =
"ibmms:fbauser1"
        }
        elseif ($type.contains("NVL")) {
            $membership="NVLms"; $rolemanager="NVLrm"; $identity =
"nvlms:fbauser1"
        }
        else {
            $membership="REDms"; $rolemanager="REDrm"; $identity =
("redms:$env:UserName").tolower()
        }

        $apFBA = New-SPAuthenticationProvider -ASPNETMembershipProvider
$membership -ASPNETRoleProviderName $rolemanager;
        $cpFBA = New-SPClaimsPrincipal -Identity $identity -IdentityType 4
    }

    if ($type.contains("SAML")) {
        $realm = "urn:" + $env:ComputerName + ":ServerName"
        $user  = "000300008448E34D@live.com"
        $certloc = "C:\LiveIDWithSAML\LiveID-INT.cer"

        $rootcert = Get-PfxCertificate $certloc
        New-SPTrustedRootAuthority "NewRootAuthority" -Certificate $rootcert |
Out-Null
```

```
       $cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2($certloc)
       $map1 = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.xmlsoap.org/claims/EmailAddress" -IncomingClaimTypeDisplayName
"http://schemas.xmlsoap.org/claims/EmailAddress" -SameAsIncoming
       $map2 = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier" -
IncomingClaimTypeDisplayName "UPN" -LocalClaimType
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"

       $apSAML = New-SPTrustedIdentityTokenIssuer -Name "LiveID" -Description
"LiveID" -Realm $realm -ImportTrustCertificate $cert -ClaimsMappings $map1,$map2 -
SignInUrl "https://login.live-int.com/login.srf" -IdentifierClaim
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier"
       $cpSAML = New-SPClaimsPrincipal -TrustedIdentityTokenIssuer $apSAML -
Identity $user.tolower()
    }

    if ($type.contains("WIN")) {
        $waexe += " `$apWIN"; $scexe += " `$cpWIN.ToEncodedString()"
    }
    elseif ($type.contains("FBA")) {
        $waexe += " `$apFBA"; $scexe += " `$cpFBA.ToEncodedString()"
    }
    else {
        $waexe += " `$apSAML -SecureSocketsLayer"; $scexe += "
`$cpSAML.ToEncodedString()"
    }

    if ($type.contains("MULTI")) {
        if ($type.contains("WIN")) {
            if ($type.contains("FBA")) {
                $waexe += ",`$apFBA"; $scexe += " -SecondaryOwnerAlias
`$cpFBA.ToEncodedString()"
            }
```

```
            if ($type.contains("SAML")) {
                $waexe += ",`$apSAML -SecureSocketsLayer"; if
(!$scexe.contains("Secondary")) { $scexe += " -SecondaryOwnerAlias
`$cpSAML.ToEncodedString()" }
            }
        }
        else {
            $waexe += ",`$apSAML -SecureSocketsLayer"; $scexe += " -
SecondaryOwnerAlias `$cpSAML.ToEncodedString()"
        }
    }

    # Check if we're creating the ANON web apps
    if ($type.contains("ANON")) { $waexe += " -AllowAnonymousAccess" }

    $waexe += " -Port $port | Out-Null"; $scexe += " | Out-Null"

    write-host -Fore Cyan "Deploying app..." -noNewLine
    Invoke-Expression $waexe

    # We could do this with a simple if/else but there may be other auth types too
    if ($type.contains("WIN"))  { Create-UserPolicy $siteurl
$cpWIN.ToEncodedString()  }
    if ($type.contains("FBA"))  { Create-UserPolicy $siteurl
$cpFBA.ToEncodedString()  }
    if ($type.contains("SAML")) { Create-UserPolicy $siteurl
$cpSAML.ToEncodedString() }

    write-host -Fore Cyan "Creating site..." -noNewLine
    Invoke-Expression $scexe

    # If this is the ANON web app, then set the root site access to entire web
    if ($type.contains("ANON")) { $web = Get-SPWeb $siteurl;
$web.AnonymousState="On"; $web.Update() }
```

```
    # At this time, let's also check if it's going to be a MixedMode web app
    if ($type.contains("MIXED")) {
        # If it's a Mixed-Mode web app we need to extend the base app to another
auth type too
        $port++; write-host -Fore Cyan "Extending port $port..." -noNewLine
        $waurl = $waurl.replace("https", "http")
        $waexe = "Get-SPWebApplication $siteurl | New-SPWebApplicationExtension -
Name $title-Ext -Zone `"Intranet`" -URL $waurl -Port $port -
AuthenticationProvider"
        if ($type.contains("WIN")) {
            if ($type.contains("FBA")) { $waexe += " `$apFBA" } else { $waexe += "
`$apSAML" }
        }
        else {
            $waexe += " `$apSAML"
        }
        Invoke-Expression $waexe
    }

    # If we've created a FBA web app, then it's time to update the CA/STS/FBA
web.config files
    if ($type.contains("FBA")) { Write-WEBConfigs 0 $port.tostring() }; write-host
-Fore Cyan "done!"
}

function Create-UserPolicy ([string]$weburl, [string]$encodeduser) {
    $webapp = Get-SPWebApplication $weburl
    $policy = $webapp.Policies.Add($encodeduser, "ClaimsWA.ps1 User")
    $role =
$webapp.PolicyRoles.GetSpecialRole([Microsoft.SharePoint.Administration.SPPolicyRo
leType]::FullControl)
    $policy.PolicyRoleBindings.Add($role)
    $webapp.Update()
```

```
}

function Write-WEBConfigs ([int]$begin, [string]$vroot) {
    # For now I'm using the XML object to load/save the config files
    # Eventually we should use the IIS:CMDlets from WebAdministration

    write-host -Fore Cyan "Writing WEBConfig..." -noNewLine
    #$filei = "\\back\scratch\suntoshs\backup\webconfigs.xml"
    $filei = "\\back\scratch\suntoshs\scripts\oobinstall\webconfigs.xml"

    $xmli = [xml](get-content $filei)
    $root = $xmli.get_DocumentElement()

    for ($j=$begin; $j -le 2; $j++) {
        if ($j -eq 0) {

[void][reflection.assembly]::LoadWithPartialName("Microsoft.SharePoint")
            $fileo =
[Microsoft.SharePoint.Administration.SPAdministrationWebApplication]::Local.IisSet
tings.get_Item(0).Path.FullName + "\web.config"
        }
        elseif ($j -eq 1) {
            $fileo = $env:CommonProgramFiles + "\Microsoft Shared\Web Server
Extensions\14\WebServices\SecurityToken\web.config"
            if ($flavor -eq "DEBUG") { $fileo = $fileo.replace("Shared", "Shared
Debug") }
        }
        else {
            if ($vroot -ne "APP") { $fileo = $env:HomeDrive +
"\inetpub\wwwroot\wss\VirtualDirectories\$vroot\web.config" }
        }

        $xmlo = [xml](get-content $fileo)
        $perf = $xmlo.CreateElement("clear")
```

```
        if ($flavor -eq "DEBUG") {
            $ship = $root.config[1].tokens.token[0].value
            $debug = $root.config[1].tokens.token[1].value
            $token =
$root.config[0]["system.web"].membership.providers.add[0].type

$root.config[0]["system.web"].membership.providers.add[0].SetAttribute("type",
$token.replace($ship,$debug)) | Out-Null
            $token =
$root.config[0]["system.web"].rolemanager.providers.add[0].type

$root.config[0]["system.web"].rolemanager.providers.add[0].SetAttribute("type",
$token.replace($ship,$debug)) | Out-Null
        }
        if ($j -eq 0) {
            # Update the CA web config
            if (-not $xmlo.SelectSingleNode("/configuration/connectionStrings")) {

$xmlo.configuration["system.web"].membership.ParentNode.RemoveChild($xmlo.configur
ation["system.web"].membership) | Out-Null

$xmlo.configuration["system.web"].roleManager.ParentNode.RemoveChild($xmlo.configu
ration["system.web"].roleManager) | Out-Null

$xmlo.SelectSingleNode("/configuration").AppendChild($xmlo.ImportNode($root.config
[0]["connectionStrings"], $true)) | Out-Null

$xmlo.SelectSingleNode("/configuration/system.web").AppendChild($xmlo.ImportNode($
root.config[0]["system.web"].membership, $true)) | Out-Null

$xmlo.SelectSingleNode("/configuration/system.web/membership/providers").PrependCh
ild($xmlo.ImportNode($perf, $true)) | Out-Null

$xmlo.SelectSingleNode("/configuration/system.web").AppendChild($xmlo.ImportNode($
root.config[0]["system.web"].rolemanager, $true)) | Out-Null
```

```
$xmlo.SelectSingleNode("/configuration/system.web/roleManager/providers").PrependC
hild($xmlo.ImportNode($perf, $true)) | Out-Null
            }
        }
        elseif ($j -eq 1) {
            # Update the STS web config
            if (-not $xmlo.SelectSingleNode("/configuration/system.web")) {

$xmlo.SelectSingleNode("/configuration").AppendChild($xmlo.ImportNode($root.config
[0]["connectionStrings"], $true)) | Out-Null

$xmlo.SelectSingleNode("/configuration").AppendChild($xmlo.ImportNode($root.config
[0]["system.web"], $true)) | Out-Null
            }
        }
        else {
            # Update the FBA web config
            if ($vroot -ne "APP") {
                if ($type.contains("PPL")) {$provider=1} elseif
($type.contains("SUN")) {$provider=2} elseif ($type.contains("IBM")) {$provider=3}
elseif ($type.contains("NVL")) {$provider=4} elseif ($type.contains("SQL"))
{$provider=5} else {$provider=0}

$xmlo.SelectSingleNode("/configuration").AppendChild($xmlo.ImportNode($root.config
[0]["connectionStrings"], $true)) | Out-Null

$xmlo.SelectSingleNode("/configuration/system.web/membership/providers").PrependCh
ild($xmlo.ImportNode($root.config[0]["system.web"].membership.providers.add[$provi
der], $true)) | Out-Null

$xmlo.SelectSingleNode("/configuration/system.web/membership/providers").PrependCh
ild($xmlo.ImportNode($perf, $true)) | Out-Null

$xmlo.SelectSingleNode("/configuration/system.web/roleManager/providers").PrependC
hild($xmlo.ImportNode($root.config[0]["system.web"].rolemanager.providers.add[$pro
```

```
vider], $true)) | Out-Null

$xmlo.SelectSingleNode("/configuration/system.web/roleManager/providers").PrependC
hild($xmlo.ImportNode($perf, $true)) | Out-Null
            }
        }
        $xmlo.Save($fileo)
    }
}
function Manage-SnapIns ([int]$action) {
    #The OWSTimer process always causes an update conflict (known bug) while
    #creating multiple web apps; let's temporarily shut it down until we're done
    if ($action -eq 1) { Stop-Service "SPTimerV4" }

    # We need to do this only if we're running on ISE so check it
    if ($host.name.contains("ISE")) {
        if ($action -eq 1) {
            write-host -Fore Yellow "Detecting host and loading dependent snap-
ins..."
            # Add-PSSnapIn WebAdministration (later!)
            Add-PSSnapIn Microsoft.Sharepoint.PowerShell
        }
        else {
            write-host -Fore Yellow "Unloading dependent snap-ins loaded earlier
on..."
            # Remove-PSSnapIn WebAdministration (later!)
            Remove-PSSnapIn Microsoft.Sharepoint.PowerShell
        }
    }
    if ($action -eq 0) {Start-Service "SPTimerV4"; write-host -Fore Yellow "`nAll
done; if there were errors please research PS database for known issues!`n"}
}
main
```

6. Start IIS Manager by typing **INETMGR** at a command prompt.
7. Go to the **Claims Web Application** site in IIS.
8. In the left pane, right-click **Claims Web Application**, and select **Edit Bindings**.
9. Select **https** and click **Edit**.
10. Under **SSL Certificate**, select any listed certificate. Consider using a self-signed certificate.
11. Import the Windows Live ID public certificate to the **Local computer**, SharePoint Server 2010, and **Trusted People** folders.
12. Perform IIS reset and browse the site URL.

Grant permissions to all Window Live ID authenticated users

Use the procedures in this section to grant permissions to all Windows Live Id authenticated users.

To grant permissions to all Windows Live ID authenticated users

1. Browse to the SharePoint Server 2010 site that you created and log on using the administrator account.
2. On the **Site Actions** menu click **Site Settings**.
3. In the **Users and Permissions** section, click **Site Permissions**.
4. Click **Site Name Visitors** group, where **Site Name** is the name of the site.
5. Click **New**, and then click **Add Users**.
6. In the **Grant Permissions** window, click the browse icon.
7. In the **Select People and Groups** window, click **All Users**, and then click **All Users (LiveIDSTS)** in the right pane.
8. Click **Add**.
9. Click **OK**.
10. Verify that **All Users (LiveIDSTS)** is now part of the visitor's group. You should now be able to log on to the SharePoint Server 2010 site with any other Live ID user's credentials.

See Also

Understanding WS-Federation (http://go.microsoft.com/fwlink/?LinkId=192377)
(*http://go.microsoft.com/fwlink/?LinkId=192377*)

Configure Kerberos authentication (SharePoint Server 2010)

In this article:

- About Kerberos authentication
- Before you begin
- Configure Kerberos authentication for SQL communications
- Create Service Principal Names for your Web applications using Kerberos authentication
- Deploy the server farm
- Configure services on servers in your farm
- Create Web applications using Kerberos authentication
- Create a site collection using the Collaboration Portal template in the portal site Web application
- Confirm successful access to the Web applications using Kerberos authentication
- Confirm correct Search Indexing functionality
- Confirm correct Search Query functionality
- Configuration limitations
- Additional resources and troubleshooting guidance

About Kerberos authentication

Kerberos is a secure protocol that supports ticketing authentication. A Kerberos authentication server grants a ticket in response to a client computer authentication request, if the request contains valid user credentials and a valid service principal name (SPN). The client computer then uses the ticket to access network resources. To enable Kerberos authentication, the client and server computers must have a trusted connection to the domain Key Distribution Center (KDC). The KDC distributes shared secret keys to enable encryption. The client and server computers must also be able to access Active Directory Domain Services (AD DS). For AD DS, the forest root domain is the center of Kerberos authentication referrals.

To deploy a server farm running Microsoft SharePoint Server 2010 using Kerberos authentication, you must install and configure a variety of applications on your computers. This article describes an example server farm running SharePoint Server 2010 and provides guidance for deploying and configuring the farm to use Kerberos authentication to support the following functionality:

- Communication between SharePoint Server 2010 and Microsoft SQL Server database software.
- Access to the SharePoint Central Administration Web application.
- Access to other Web applications, including a portal site Web application and a My Site Web application.

Before you begin

This article is intended for administrative-level personnel who have an understanding of the following:

- Windows Server 2008
- Active Directory
- Internet Information Services (IIS) 6.0 (or IIS 7.0)
- Microsoft SharePoint Foundation 2010
- SharePoint Server 2010
- Windows Internet Explorer
- Kerberos authentication, as implemented in Active Directory Domain Services (AD DS) for Windows Server 2008
- Network Load Balancing (NLB) in Windows Server 2008
- Computer accounts in an Active Directory domain
- User accounts in an Active Directory domain
- IIS Web sites and their bindings and authentication settings
- IIS application pool identities for IIS Web sites
- The SharePoint Products Configuration Wizard
- SharePoint Foundation 2010 and SharePoint Server 2010 Web applications
- Central Administration pages
- Service principal names (SPNs) and how to configure them in an Active Directory domain

Important:
To create SPNs in an Active Directory domain, you must have domain administrative-level permissions.

This article does not provide an in-depth examination of Kerberos authentication. Kerberos is an industry-standard authentication method that is implemented in AD DS.

This article does not provide detailed, step-by-step instructions for installing SharePoint Server 2010 or using the SharePoint Products Configuration Wizard.

This article does not provide detailed, step-by-step instructions for using Central Administration to create SharePoint Server 2010 Web applications.

Software version requirements

The guidance provided in this article, and the testing performed to confirm this guidance, are based on results using systems running Windows Server 2008 and Internet Explorer with the latest updates applied from the Windows Update site (*http://go.microsoft.com/fwlink/?LinkID=101614&clcid=0x409*).

The following software versions were installed:

- Windows Server 2008, with the latest updates from the Windows Update site (*http://go.microsoft.com/fwlink/?LinkID=101614&clcid=0x409*)
- Internet Explorer
- The released version of SharePoint Server 2010

You should also make sure that your Active Directory domain controllers are running Windows Server 2008, with the latest updates applied from the Windows Update site (*http://go.microsoft.com/fwlink/?LinkID=101614&clcid=0x409*).

Known issues

SharePoint Server 2010 can crawl Web applications configured to use Kerberos authentication if those Web applications are hosted on IIS virtual servers that are bound to default ports (TCP port 80 and Secure Sockets Layer (SSL) port 443). However, SharePoint Server 2010 Search cannot crawl SharePoint Server 2010 Web applications that are configured to use Kerberos authentication if the Web applications are hosted on IIS virtual servers that are bound to non-default ports (ports other than TCP port 80 and SSL port 443). Currently, SharePoint Server 2010 Search can only crawl SharePoint Server 2010 Web applications hosted on IIS virtual servers bound to non-default ports that are configured to use either NTLM authentication or Basic authentication.

For end-user access using Kerberos authentication, if you need to deploy Web applications that can only be hosted on IIS virtual servers that are bound to non-default ports, and if you want end users to get search query results, then:

- The same Web applications must be hosted on other IIS virtual servers on non-default ports.
- The Web applications must be configured to use either NTLM or Basic authentication.
- Search Indexing must crawl the Web applications using NTLM or Basic authentication.

This article provides guidance for:

- Configuring the Central Administration Web application using Kerberos authentication hosted on an IIS virtual server bound to non-default ports.
- Configuring portal and My Site applications, using Kerberos authentication hosted on IIS virtual servers bound to default ports and with an IIS host header binding.
- Ensuring that Search Indexing successfully crawls SharePoint Server 2010 Web applications using Kerberos authentication.
- Ensuring that users accessing Kerberos-authenticated Web applications can successfully get search query results for those Web applications.

Additional background

It is important to understand that when you use Kerberos authentication, accurate authentication functionality is dependent in part on the behavior of the client that is attempting to authenticate using Kerberos. In a SharePoint Server 2010 farm deployment using Kerberos authentication, SharePoint

Server 2010 is not the client. Before you deploy a server farm running SharePoint Server 2010 using Kerberos authentication, you must understand the behavior of the following clients:

- The browser (in the context of this article, the browser is always Internet Explorer)
- The Microsoft .NET Framework

The browser is the client used when browsing to a Web page in a SharePoint Server 2010 Web application. When SharePoint Server 2010 performs tasks such as crawling the local SharePoint Server 2010 content sources, the .NET Framework is functioning as the client.

For Kerberos authentication to work correctly, you must create SPNs in AD DS. If the services to which these SPNs correspond are listening on non-default ports, the SPNs should include port numbers. This is to ensure that the SPNs are meaningful. It is also required to prevent the creation of duplicate SPNs.

When a client attempts to access a resource using Kerberos authentication, the client must construct an SPN to be used as part of the Kerberos authentication process. If the client does not construct an SPN that matches the SPN that is configured in AD DS, Kerberos authentication will fail, usually with an "Access denied" error.

There are versions of Internet Explorer that do not construct SPNs with port numbers. If you are using SharePoint Server 2010 Web applications that are bound to non-default port numbers in IIS, you might have to direct Internet Explorer to include port numbers in the SPNs that it constructs. In a farm running SharePoint Server 2010, the Central Administration Web application is hosted, by default, in an IIS virtual server that is bound to a non-default port. Therefore, this article addresses both IIS Web sites that are port-bound and IIS Web sites that are bound to host-headers.

By default, in a farm running SharePoint Server 2010, the .NET Framework does not construct SPNs that contain port numbers. This is the reason why Search cannot crawl Web applications using Kerberos authentication if those Web applications are hosted on IIS virtual servers that are bound to non-default ports.

Server farm topology

This article targets the following SharePoint Server 2010 server farm topology:

- Two computers running Windows Server 2008 that are acting as front-end Web servers, with Windows NLB configured.
- Three computers running Windows Server 2008 that are acting as application servers. One of the application servers hosts the Central Administration Web application. The second application server is running Search Query, and the third application server is running Search Indexing.
- One computer running Windows Server 2008 that is used as the SQL host for the farm running SharePoint Server 2010. For the scenario described in this article, you can use either Microsoft SQL Server 2008.

Active Directory Domain Services, computer naming, and NLB conventions

The scenario described in this article uses the following Active Directory, computer-naming, and NLB conventions:

Server role	Domain name
Active Directory Domain Services	mydomain.net
A front-end Web server running SharePoint Server 2010	mossfe1.mydomain.net
A front-end Web server running SharePoint Server 2010	mossfe2.mydomain.net
SharePoint Server 2010 Central Administration	mossadmin.mydomain.net
Search Indexing running SharePoint Server 2010	mosscrawl.mydomain.net
Search Query running SharePoint Server 2010	mossquery.mydomain.net
SQL Server host running SharePoint Server 2010	mosssql.mydomain.net

An NLB VIP is assigned to mossfe1.mydomain.net and mossfe2.mydomain.net as a result of configuring NLB on these systems. A set of DNS host names that point to this address is registered in your DNS system. For example, if your NLB VIP is 192.168.100.200, you have a set of DNS records that resolve the following DNS names to this IP address (192.168.100.200):

- kerbportal.mydomain.net
- kerbmysite.mydomain.net

Active Directory domain account conventions

The example in this article uses the naming conventions listed in the following table for service accounts and application pool identities used in the farm running SharePoint Server 2010.

Domain account or application pool identity	Name
Local administrator account • On all servers running SharePoint Server 2010 (but not on the host computer running SQL Server) • For SharePoint Server 2010 setup and for the SharePoint Products Configuration Wizard run-as user	mydomain\pscexec

Domain account or application pool identity	Name
Local administrator account on the SQL Server host computer	mydomain\sqladmin
SQL Server service account used to run the SQL Server service	mydomain\mosssqlsvc
SharePoint Server 2010 farm administrator account	mydomain\mossfarmadmin This is used as the application pool identity for Central Administration and as the service account for the SharePoint Timer Service.
SharePoint Server 2010 application pool identity for the portal site Web application	mydomain\portalpool
SharePoint Server 2010 application pool identity for the My Site Web application	mydomain\mysitepool
SharePoint Foundation 2010 search service account	mydomain\wsssearch
SharePoint Foundation 2010 search content access account	mydomain\wsscrawl
SharePoint Server 2010 search service account	mydomain\mosssearch
SharePoint Server 2010 content access account	mydomain\mosscrawl

Preliminary configuration requirements

Before you install SharePoint Server 2010 on the computers in your server farm, make sure you have performed the following procedures:

- All servers used in the farm, including the SQL host, are set up with Windows Server 2008, including the latest updates applied from the Windows Update site (*http://go.microsoft.com/fwlink/?LinkID=101614&clcid=0x409*).
- All servers in the farm have Internet Explorer (and the latest updates for it) installed from the Windows Update site (*http://go.microsoft.com/fwlink/?LinkID=101614&clcid=0x409*).
- SQL Server 2008 is installed and running on the SQL host computer, and the SQL Server service is running as the account, mydomain\sqlsvc. A default instance of SQL Server is installed and is listening on TCP port 1433.
- The SharePoint Products Configuration Wizard run-as user has been added:

- As a SQL Login on your SQL host.
- To the SQL Server DBCreators role on your SQL host.
- To the SQL Server Security Administrators role on your SQL host.

Configure Kerberos authentication for SQL communications

Configure Kerberos authentication for SQL communications before installing and configuring SharePoint Server 2010 on your servers running SharePoint Server 2010. This is necessary because Kerberos authentication for SQL communications has to be configured, and confirmed to be working, before your computers running SharePoint Server 2010 can connect to your SQL Server.

The process of configuring Kerberos authentication for any service installed on a host computer running Windows Server 2008 includes creating an SPN for the domain account used to run the service on the host. SPNs are made up of the following parts:

- A Service Name (for example, MSSQLSvc or HTTP)
- A host name (either real or virtual)
- A port number

The following list contains examples of SPNs for a default instance of SQL Server running on a computer named mosssql and listening on port 1433:

- MSSQLSvc/mosssql:1433
- MSSQLSvc/mosssql.mydomain.com:1433

These are the SPNs that you will create for the instance of SQL Server on the SQL host that will be used by the farm described in this article. You should always create SPNs that have both a NetBIOS name and a full DNS name for a host on your network.

There are different methods that you can use to set an SPN for an account in an Active Directory domain. One method is to use the SETSPN.EXE utility that is part of the resource kit tools for Windows Server 2008. Another method is to use the ADSIEDIT.MSC snap-in on your Active Directory domain controller. This article addresses using the ADSIEDIT.MSC snap-in.

There are two core steps for configuring Kerberos authentication for SQL Server:

- Create SPNs for your SQL Server service account.
- Confirm Kerberos authentication is used to connect servers running SharePoint Server 2010 to servers running SQL Server.

Create the SPNs for your SQL Server service account

1. Log on to your Active Directory domain controller using the credentials of a user that has domain administrative permissions.
2. In the **Run** dialog box, type **ADSIEDIT.MSC**.
3. In the management console dialog box, expand the domain container folder.

4. Expand the container folder containing user accounts, for example CN=Users.
5. Locate the container for the SQL Server Service account, for example CN=mosssqlsvc.
6. Right-click this account, and then click **Properties**.
7. Scroll down the list of properties in the **SQL Server Service account** dialog box until you find **servicePrincipalName**.
8. Select the **servicePrincipalName** property and click **Edit**.
9. In the **Value to Add field**, in the **Multi-Valued String Editor** dialog box, type the SPN **MSSQLSvc/mosssql:1433** and click **Add**. Next, type the SPN **MSSQLSvc/mosssql.mydomain.com:1433** in this field and click **Add**.
10. Click **OK** on the **Multi-Valued String Editor** dialog box, and then click **OK** on the properties dialog box for the SQL Server service account.

Confirm Kerberos authentication is used to connect servers running SharePoint Server 2010 to SQL Server

Install the SQL Client Tools on one of your servers running SharePoint Server 2010, and use the tools to connect from your server running SharePoint Server 2010 to those running SQL Server. This article does not address the steps for installing the SQL Client Tools on one of your servers running SharePoint Server 2010. The confirmation procedures are based on the following assumptions:

- You are using SQL Server 2008 on your SQL host.
- You have logged on to one of your servers running SharePoint Server 2010, using the account mydomain\pscexec, and have installed the SQL 2005 Client Tools on the server running SharePoint Server 2010.

1. Run the SQL Server 2005 Management Studio.
2. When the **Connect to Server** dialog box appears, type the name of the SQL host computer (in this example, the SQL host computer is mosssql), and click **Connect** to connect to the SQL host computer.
3. To confirm that Kerberos authentication was used for this connection, run the event viewer on the SQL host computer and examine the Security event log. You should see a Success Audit record for a Logon/Logoff category event that is similar to the data shown in the following tables:

Event Type	Success Audit
Event Source	Security
Event Category	Logon/Logoff
Event ID	540
Date	10/31/2007

Time	4:12:24 PM
User	MYDOMAIN\pscexec
Computer	MOSSSQL
Description	

An example of a successful network logon is depicted in the following table.

User Name	pscexec
Domain	MYDOMAIN
Logon ID	(0x0,0x6F1AC9)
Logon Type	3
Logon Process	Kerberos
Workstation Name	
Logon GUID	{36d6fbe0-2cb8-916c-4fee-4b02b0d3f0fb}
Caller User Name	
Caller Domain	
Caller Logon ID	
Caller Process ID	
Transited Services	
Source Network Address	192.168.100.100
Source Port	2465

Examine the log entry to confirm that:

1. The user name is correct. The mydomain\pscexec account logged on over the network to the SQL host.
2. The logon type is 3. A type 3 logon is a network logon.
3. The logon process and authentication package both use Kerberos authentication. This confirms that your server running SharePoint Server 2010 is using Kerberos authentication to communicate with the SQL host.
4. The Source Network Address matches the IP address of the computer from which the connection was made.

If your connection to the SQL host fails with an error message similar to **Cannot generate SSPI context**, it is likely that there is an issue with the SPN being used for your instance of SQL Server. To troubleshoot and correct this, please refer to the article How to troubleshoot the "Cannot generate SSPI context" error message (*http://go.microsoft.com/fwlink/?LinkId=76621*) from the Microsoft Knowledge Base.

Create Service Principal Names for your Web applications using Kerberos authentication

As far as Kerberos authentication is concerned, there is nothing special about IIS-based SharePoint Server 2010 Web applications—Kerberos authentication treats them as just another IIS Web site.

This process requires knowledge of the following items:

- The Service Class for the SPN (in the context of this article, for SharePoint Server 2010 Web applications, this is always HTTP).
- The URL for all of your SharePoint Server 2010 Web applications using Kerberos authentication.
- The host name portion of the SPN (either real or virtual; this article addresses both).
- The port number portion of the SPN (in the scenario described in this article, both IIS port-based and IIS host-header-based SharePoint Server 2010 Web applications are used).
- The Windows Active Directory accounts for which your SPNs must be created.

The following table lists the information for the scenario described in this article:

URL	Active Directory account	SPN
http://mossadmin.mydomain.net:10000	mossfarmadmin	• HTTP/mossadmin.mydomain.net:10000 • HTTP/mossadmin.mydomain.net:10000
http://kerbportal.mydomain.net	portalpool	• HTTP/kerbportal.mydomain.net • HTTP/kerbportal
http://kerbmysite.mydomain.net	mysitepool	• HTTP/kerbmysite.mydomain.net • HTTP/kerbmysite

Notes for this table:

- The first URL listed above is for Central Administration, and uses a port number. You don't have to use port 10000. This is just an example used for consistency throughout this article.
- The next two URLs are for the portal site and My Site, respectively.

Use the guidance provided above to create the SPNs you need in AD DS to support Kerberos authentication for your SharePoint Server 2010 Web applications. You need to log on to a domain controller in your environment using an account that has domain administrative permissions. To create the SPNs, you can use either the SETSPN.EXE utility mentioned previously, or you can use the ADSIEDIT.MSC snap-in mentioned previously. If using the ADSIEDIT.MSC snap-in, please refer to the instructions provided earlier in this article for creating the SPNs. Be sure to create the correct SPNs for the correct accounts in AD DS.

Deploy the server farm

Deploying the server farm includes the following steps:

1. Set up SharePoint Server 2010 on all of your servers running SharePoint Server 2010.
2. Run the SharePoint Products Configuration Wizard and create a new farm. This step includes creating a SharePoint Server 2010 Central Administration Web application that will be hosted on an IIS virtual server bound to a non-default port and use Kerberos authentication.
3. Run the SharePoint Products Configuration Wizard and join the other servers to the farm.
4. Configure Services on Servers in your farm for:
    - SharePoint Foundation 2010 Search service
    - SharePoint Server 2010 Search Indexing
    - SharePoint Server 2010 Search Query
5. Create Web applications that are used for the portal site and My Site, using Kerberos authentication.
6. Create a site collection using the Collaboration Portal template in the portal site Web application.
7. Confirm successful access to the Web applications using Kerberos authentication.
8. Confirm correct Search Indexing functionality.
9. Confirm correct Search Query functionality.

Install SharePoint Server 2010 on all of your servers

This is the straightforward process of running SharePoint Server 2010 setup to install the SharePoint Server 2010 binaries on your servers running SharePoint Server 2010. Log on to each of your computers running SharePoint Server 2010 using the account mydomain\pscexec. No step-by-step instructions are provided for this. For the scenario described in this article, do a **Complete** installation of SharePoint Server 2010 on all servers that require SharePoint Server 2010.

Create a new farm

For the scenario described in this article, run the SharePoint Products Configuration Wizard from the MOSSADMIN Search Indexing server first, so that MOSSADMIN hosts the SharePoint Server 2010 Central Administration Web application.

On the server named MOSSCRAWL, when setup completes, a **Setup Complete** dialog box appears with a check box selected to run the SharePoint Products Configuration Wizard. Leave this check box selected and close the setup dialog box to run the SharePoint Products Configuration Wizard.

When running the SharePoint Products Configuration Wizard on this computer, create a new farm using the following settings:

- Provide the database server name (in this article, it is the server named MOSSSQL).
- Provide a configuration database name (you can use the default, or stipulate a name of your choice).
- Provide the database access (farm administrator) account information. Using the scenario in this article, that account is mydomain\mossfarmadmin.
- Provide the information required for the SharePoint Server 2010 Central Administration Web application. Using the scenario in this article, that information is:
    - Central Administration Web application port number: 10000
    - Authentication Method: Negotiate

When you have provided all the required information, the SharePoint Products Configuration Wizard should finish successfully. If it completes successfully, confirm that you can access the SharePoint Server 2010 Central Administration Web application home page using Kerberos authentication. To do this, perform the following steps:

1. Log on to a different server running SharePoint Server 2010 or another computer in the domain mydomain as mydomain\pscexec. You should not verify correct Kerberos authentication behavior directly on the computer hosting the SharePoint Server 2010 Central Administration Web application. This should be done from a separate computer in the domain.
2. Start Internet Explorer on this server and attempt to go to the following URL: http://mossadmin.mydomain.net:10000. The home page of Central Administration should render.
3. To confirm that Kerberos authentication was used to access Central Administration, go back to the computer named MOSSADMIN and run the event viewer and look in the security log. You should see a Success Audit record that looks similar to the following table:

Event Type	Success Audit
Event Source	Security
Event Category	Logon/Logoff

Event ID	540
Date	11/1/2007
Time	2:22:20 PM
User	MYDOMAIN\pscexec
Computer	MOSSADMIN
Description	

An example of a successful network logon is depicted in the following table.

User Name	pscexec
Domain	MYDOMAIN
Logon ID	(0x0,0x1D339D3)
Logon Type	3
Logon Process	Kerberos
Authentication Package	Kerberos
Workstation Name	
Logon GUID	{fad7cb69-21f8-171b-851b-3e0dbf1bdc79}
Caller User Name	
Caller Domain	
Caller Logon ID	
Caller Process ID	
Transited Services	
Source Network Address	192.168.100.100
Source Port	2505

Examination of this log record shows the same type of information as in the previous log entry:

- Confirm that the user name is correct; it is the mydomain\pscexec account that logged on over the network to the server running SharePoint Server 2010 that is hosting Central Administration.
- Confirm that the logon type is 3; a logon type 3 is a network logon.

- Confirm that the logon process and authentication package both use Kerberos authentication. This confirms that Kerberos authentication is being used to access your Central Administration Web application.
- Confirm that the Source Network Address matches the IP address of the computer from which the connection was made.

If the Central Administration home page fails to render and instead an **unauthorized** error message is displayed, Kerberos authentication is failing. There are usually only two causes for this failure:

- The SPN in AD DS was not registered for the correct account. It should have been registered for mydomain\mossfarmadmin.
- The SPN in AD DS does not match the SPN being constructed by Internet Explorer or is otherwise invalid. You might have omitted the port number from the SPN that you registered in AD DS. Ensure that this is corrected and that Central Administration is working, using Kerberos authentication, before proceeding.

Note:
A diagnostic aid you could use to see what is going on over the network is a network sniffer, such as Microsoft Network Monitor, to take a trace during browsing to Central Administration. After the failure, examine the trace and look for KerberosV5 Protocol packets. Find a packet with an SPN constructed by Internet Explorer. If the SPN in the trace looks correct, either the SPN in AD DS is invalid, or it has been registered for the wrong account.

Join the other servers to the farm

Now that your farm has been created and you can successfully access Central Administration using Kerberos authentication, you need to run the SharePoint Products Configuration Wizard and join the other servers to the farm.

On each of the other four servers running SharePoint Server 2010 (mossfe1, mossfe2, mossquery, and mosscrawl), SharePoint Server 2010 installation should have completed, and the setup completion dialog box should appear with the SharePoint Products Configuration Wizard check box selected. Leave this check box selected and close the setup completion dialog box to run the SharePoint Products Configuration Wizard. Perform the procedure to join each of these servers to the farm.

Upon completion of the SharePoint Products Configuration Wizard on each server you add to the farm, verify that each of these servers can render Central Administration, which is running on the server, MOSSADMIN. If any of these servers fail to render Central Administration, take the appropriate steps to solve the problem before you proceed.

Configure services on servers in your farm

Configure specific SharePoint Foundation 2010 and SharePoint Server 2010 services to run on specific servers running SharePoint Foundation 2010 and SharePoint Server 2010 in the farm, using the accounts indicated in the following sections.

Note:
This section does not provide an in-depth description of the user interface. Only high-level instructions are provided. You should be familiar with Central Administration and how to perform the required steps before you proceed.

Access Central Administration and perform the following steps to configure the services on the servers indicated, using the accounts indicated.

Windows SharePoint Services Search

On the Services on Server page in Central Administration:

1. Select the server MOSSQUERY.
2. In the list of services that appears, close to the middle of the page, locate the SharePoint Foundation 2010 Search service, and then click **Start** in the **Action** column.
3. On the subsequent page, provide the credentials for the SharePoint Foundation 2010 search service account and for the SharePoint Foundation 2010 Content Access account. In the scenario in this article, the SharePoint Foundation 2010 search service account is mydomain\wsssearch, and the SharePoint Foundation 2010 content access account is mydomain\wsscrawl. Type the account names and passwords in the appropriate locations on the page, and then click **Start**.

Index server

On the Services on Server page in Central Administration:

1. Select the server MOSSCRAWL.
2. In the list of services that appears close to the middle of the page, locate the SharePoint Server 2010 Search service, and then click **Start** in the **Action** column.

On the subsequent page, check the **Use this server for indexing content** check box and then provide the credentials for the SharePoint Server 2010 search service account. In the scenario in this article, the SharePoint Server 2010 search service account is mydomain\mosssearch. Type the account names and passwords in the appropriate locations on the page, and then click **Start**.

Query server

On the Services on Server page in Central Administration:

1. Select the server MOSSQUERY.

2. In the list of services that appears close to the middle of the page, locate the SharePoint Server 2010 Search service, and then click the service name in the Service column.

On the subsequent page, check the **Use this server for serving search queries** check box and click **OK**.

Create Web applications using Kerberos authentication

In this section, create Web applications that are used for the portal site and a My Site in your farm.

Note:
This section does not provide an in-depth description of the user interface. Only high-level instructions are provided. You should be familiar with Central Administration and how to perform the required steps before you proceed.

Create the portal site Web application

1. On the Application Management page in Central Administration, click **Create or extend Web application**.
2. On the subsequent page, click **Create a new Web application**.
3. On the subsequent page, make sure **Create a new IIS Web site** is selected.
   - In the **Description** field, type **PortalSite**.
   - In the **Port** field, type **80**.
   - In the **Host Header** field, type **kerbportal.mydomain.net**.
4. Make sure **Negotiate** is selected as the authentication provider for this Web application.
5. Create this Web application in the Default zone. Do not modify the zone for this Web application.
6. Make sure **Create new application pool** is selected.
   - In the **Application Pool Name** field, type **PortalAppPool**.
   - Make sure **Configurable** is selected. In the **User name** field, type the account **mydomain\portalpool**.
7. Click **OK**.
8. Confirm that the Web application is successfully created.

Note:
If you want to use an SSL connection and bind the Web application to port 443, type **443** in the **Port** field and select **Use SSL** on the Create New Web Application page. In addition, you must install an SSL wildcard certificate. When using an IIS host header binding on an IIS Web site configured for SSL, you must use an SSL wildcard certificate. For more information about SSL host headers in IIS, see Configuring SSL Host Headers (IIS 6.0) (*http://go.microsoft.com/fwlink/?LinkId=111285&clcid=0x409*).

Create the My Site Web application

1. On the Application Management page in Central Administration, click **Create or extend Web application**.
2. On the subsequent page, click **Create a new Web application**.
3. On the subsequent page, make sure **Create a new IIS Web site** is selected.
   - In the **Description** field, type **MySite**.
   - In the **Port** field, type **80**.
   - In the **Host Header** field, type **kerbmysite.mydomain.net**.
4. Make sure **Negotiate** is selected as the authentication provider for this Web application.
5. Create this Web application in the Default zone. Do not modify the zone for this Web application.
6. Make sure **Create new application pool** is selected.
   - In the **Application Pool Name** field, type **MySiteAppPool**.
   - Make sure **Configurable** is selected. In the **User name** field, type the account **mydomain\mysitepool**.
7. Click **OK**.
8. Confirm that the Web application is successfully created.

Note:
If you want to use an SSL connection and bind the Web application to port 443, type **443** in the **Port** field and select **Use SSL** on the Create New Web Application page. In addition, you must install an SSL wildcard certificate. When using an IIS host header binding on an IIS Web site configured for SSL, you must use an SSL wildcard certificate. For more information about SSL host headers in IIS, see Configuring SSL Host Headers (IIS 6.0) (*http://go.microsoft.com/fwlink/?LinkId=111285&clcid=0x409*).

Create a site collection using the Collaboration Portal template in the portal site Web application

In this section, you create a site collection on the portal site in the Web application that you created for this purpose.

Note:
This section does not provide an in-depth description of the user interface. Only high-level instructions are provided. You should be familiar with Central Administration and how to perform the required steps before you proceed.

1. On the Application Management page in Central Administration, click **Create site collection**.
2. On the subsequent page, make sure you select the correct Web application. For the example in this article, select **http://kerbportal.mydomain.net**.
3. Provide the title and description you want to use for this site collection.

4. Leave the Web site address unchanged.
5. In the **Template Selection** section under **Select a Template**, click the **Publishing** tab and select the **Collaboration Portal** template.
6. In the **Primary Site Collection Administrator** section, type **mydomain\pscexec**.
7. Specify the Secondary Site Collection Administrator you want to use.
8. Click **OK**.
9. Confirm that the portal site collection is successfully created.

Confirm successful access to the Web applications using Kerberos authentication

Confirm that Kerberos authentication is working for the recently created Web applications. Start with the portal site.

To do this, perform the following steps:

1. Log on to a server running SharePoint Server 2010 rather than either of the two front-end Web servers that are configured for NLB as mydomain\pscexec. You should not verify correct Kerberos authentication behavior directly on one of the computers hosting the load-balanced Web sites using Kerberos authentication. This should be done from a separate computer in the domain.
2. Start Internet Explorer on this other system and attempt to go to the following URL: http://kerbportal.mydomain.net.

The home page of the Kerberos-authenticated portal site should render.

To confirm that Kerberos authentication was used to access the portal site, go to one of the load-balanced front-end Web servers and run the event viewer and look in the security log. You should see a Success Audit record, similar to the following table, on one of the front-end Web servers. Note that you may have to look on both front-end Web servers before you find this, depending on which system handled the load-balanced request.

Event Type	Success Audit
Event Source	Security
Event Category	Logon/Logoff
Event ID	540
Date	11/1/2007
Time	5:08:20 PM
User	MYDOMAIN\pscexec
Computer	mossfe1
Description	

An example of a successful network logon is depicted in the following table.

User Name	pscexec
Domain	MYDOMAIN
Logon ID	(0x0,0x1D339D3)
Logon Type	3
Logon Process	Kerberos authentication
Workstation Name	
Logon GUID	{fad7cb69-21f8-171b-851b-3e0dbf1bdc79}
Caller User Name	
Caller Domain	
Caller Logon ID	
Caller Process ID	
Transited Services	
Source Network Address	192.168.100.100
Source Port	2505

Examination of this log record shows the same type of information as in the previous log entry:

- Confirm that the user name is correct; it is the mydomain\pscexec account that logged on over the network to the front-end Web server running SharePoint Server 2010 that is hosting the portal site.
- Confirm that the logon type is 3; a logon type 3 is a network logon.
- Confirm that the logon process and authentication package both use Kerberos authentication. This confirms that Kerberos authentication is being used to access your portal site.
- Confirm that the Source Network Address matches the IP address of the computer from which the connection was made.

If the home page of the portal site fails to render, and displays an “unauthorized” error message, then Kerberos authentication is failing. There are usually only a couple of causes for this:

- The SPN in AD DS was not registered for the correct account. It should have been registered for mydomain\portalpool, for the Web application of the portal site.
- The SPN in AD DS does not match the SPN being constructed by Internet Explorer or is invalid for another reason. In this case, because you are using IIS host headers without explicit port numbers, the SPN registered in AD DS differs from the IIS host header specified when you extended the Web application. You need to correct this to get Kerberos authentication working.

Note:
A diagnostic aid you could use to see what is going on over the network is a network sniffer such as Microsoft Network Monitor to take a trace during browsing to Central Administration. After the failure, examine the trace and look for KerberosV5 Protocol packets. You should find a packet with an SPN constructed by Internet Explorer. If the SPN in the trace looks correct, then either the SPN in AD DS is invalid or the SPN has been registered for the wrong account.

After you have Kerberos authentication working for your portal site, go to your Kerberos-authenticated My Site, using the following URL:

- http://kerbmysite.mydomain.net

Note:
The first time you access the My Site URL, it will take some time for SharePoint Server 2010 to create a My Site for the logged-on user. However, it should succeed, and the My Site page for that user should render.

This should work correctly. If it does not work, refer to the preceding troubleshooting steps.

Confirm correct Search Indexing functionality

Confirm that Search Indexing is successfully crawling the content hosted on this farm. This is the step you must take prior to confirming the Search Query results for users accessing the sites using Kerberos authentication.

Note

- This section does not provide an in-depth description of the user interface. Only high-level instructions are provided. You should be familiar with Central Administration and how to perform the required steps before you proceed.
- To confirm Search Indexing functionality, access a Web application and start a full crawl. Wait for the crawl to complete. If the crawl fails, you must investigate and correct the failure, and then run a full crawl. If the crawl fails with "access denied" errors, it is either because the crawling account does not have access to the content sources, or because Kerberos authentication has failed. Whatever the cause, this error must be corrected before proceeding to subsequent steps.

You must complete a full crawl of the Kerberos-authenticated Web applications before proceeding.

Confirm correct Search Query functionality

To confirm that Search Query returns results for users accessing the portal site that uses Kerberos authentication:

1. Start Internet Explorer on a system in mydomain.net and go to http://kerbportal.mydomain.net.
2. When the home page of the portal site renders, type a search keyword in the **Search** field and press **ENTER**.
3. Confirm that Search Query results are returned. If they are not, confirm that the keyword you have entered is valid in your deployment, that Search Indexing is running correctly, that the Search service is running on your Search Indexing and Search Query servers, and that there are no problems with search propagation from your Search Index server to your Search Query server.

Configuration limitations

The host name portion of the new-format SPNs that are created will be the NetBIOS name of the host running the service, for example: MSSP/kerbtest4:56738/SSP1. This is because the host names are fetched from the SharePoint Server 2010 configuration database, and only NetBIOS computer names are stored in the SharePoint Server 2010 configuration database. This might be ambiguous in certain scenarios. Currently, the Stsadm command-line tool to rename a server running SharePoint Server 2010 cannot be successfully used to rename a server running SharePoint Server 2010, so there is no workaround for this issue.

Additional resources and troubleshooting guidance

Product/technology	Resource
Kerberos scenarios white paper	Configuring Kerberos authentication for SharePoint 2010 Products (*http://go.microsoft.com/fwlink/?LinkId=197178*) End-to-end guidance for configuring Kerberos in nine specific scenarios, including core deployment, three Microsoft SQL Server solutions, and scenarios that use Excel Services, PowerPivot, Visio Services, PerformancePoint Services, and Business Connectivity Services. Also includes conceptual information about identity and Kerberos authentication.
SQL Server	How to make sure that you are using Kerberos authentication when you create a remote connection to an instance of SQL Server 2005 (*http://go.microsoft.com/fwlink/?LinkId=85942&clcid=0x409*)
SQL Server	How to troubleshoot the "Cannot generate SSPI context" error message (*http://go.microsoft.com/fwlink/?LinkId=82932&clcid=0x409*)
.NET Framework	AuthenticationManager.CustomTargetNameDictionary Property (*http://go.microsoft.com/fwlink/?LinkId=120460&clcid=0x409*)

Product/technology	Resource
Internet Explorer	Error message in Internet Explorer when you try to access a Web site that requires Kerberos authentication on a Windows XP-based computer: "HTTP Error 401 - Unauthorized: Access is denied due to invalid credentials" (*http://go.microsoft.com/fwlink/?LinkId=120462&clcid=0x409*)
Kerberos authentication	Kerberos Authentication Technical Reference (*http://go.microsoft.com/fwlink/?LinkId=78646&clcid=0x409*)
Kerberos authentication	Troubleshooting Kerberos Errors (*http://go.microsoft.com/fwlink/?LinkId=93730&clcid=0x409*)
Kerberos authentication	Kerberos Protocol Transition and Constrained Delegation (*http://go.microsoft.com/fwlink/?LinkId=100941&clcid=0x409*)
IIS	Configuring SSL Host Headers (IIS 6.0) (*http://go.microsoft.com/fwlink/?LinkId=120463&clcid=0x409*)

Configuring Kerberos authentication for SharePoint 2010 Products (white paper)

This document gives you information that will help you understand the concepts of identity in Microsoft SharePoint 2010 Products, how Kerberos authentication plays a very important role in authentication and delegation scenarios, and the situations where Kerberos authentication should be used or may be required in solution designs. Scenarios include business intelligence implementations which secure access to external data sources such as SQL Server.

The document also shows how to configure Kerberos authentication end-to-end within your environment, including scenarios that use various service applications in Microsoft SharePoint Server. Additional tools and resources are described to help you test and validate Kerberos configuration. The "Step-by-Step Configuration" sections of this document cover the following scenarios for SharePoint Server 2010.

- Scenario 1: Core Configuration
- Scenario 2: Kerberos Authentication for SQL OLTP
- Scenario 3: Identity Delegation for SQL Analysis Services
- Scenario 4: Identity Delegation for SQL Reporting Services
- Scenario 5: Identity Delegation for Excel Services
- Scenario 6: Identity Delegation for PowerPivot for SharePoint
- Scenario 7: Identity Delegation for Visio Services
- Scenario 8: Identity Delegation for PerformancePoint Services
- Scenario 9: Identity Delegation for Business Connectivity Services

Two options

The information about Configuring Kerberos authentication for SharePoint 2010 Products is available in two forms:

- You can download the white paper in .doc format: Configuring Kerberos Authentication for Microsoft SharePoint 2010 Products and Technologies (*http://go.microsoft.com/fwlink/?LinkId=196600*) (7.3 MB)
- All the same information is available as a set of articles here in the TechNet Library. To begin, see: Overview of Kerberos authentication for Microsoft SharePoint 2010 Products (*http://technet.microsoft.com/library/33241bc8-8be1-43f9-b395-9e2471d61df8(Office.14).aspx*)

Create a site collection (SharePoint Server 2010)

A site collection is a group of Web sites that have the same owner and share administration settings, for example, permissions. When you create a site collection, a top-level site is automatically created in the site collection. You can then create one or more subsites below the top-level site.

A site collection must exist within a Web application. You can create a site collection based on an existing Web application, or you can create a Web application and then create a site collection within that application. For more information, see Create a Web application (SharePoint Server 2010).

If your Web application is for a single project or for use by a single team, you should use a single site collection to avoid the overhead of managing multiple sites. However, complex solutions benefit from multiple site collections because it is easier to organize content and manage permissions for each site collection. For example, because there is no built-in navigation from one site collection to another, having multiple site collections can provide an additional layer of security for site content.

SharePoint provides site templates in the following categories: collaboration, meetings, enterprise, publishing, and custom. When you create a site collection, you select the template that matches what you want the site to do. For example, choose the Publishing Portal template if you want to create a large intranet site that has many more readers than contributors.

Before you create a site collection, ensure that the following prerequisites are available:

- A Web application in which to create the site collection.
- A quota template, if you plan to define values that specify how much data can be stored in a site collection and the storage size that triggers an e-mail alert to the site collection administrator. For more information, see Create, edit, and delete quota templates (SharePoint Server 2010) (*http://technet.microsoft.com/library/c2eda191-1814-423b-882f-1fdafe9df6c9(Office.14).aspx*).
- A custom managed wildcard path, if you plan to create the site collection somewhere other than under the root (/) directory or the /sites/ directory. For more information, see Define managed paths (SharePoint Server 2010) (*http://technet.microsoft.com/library/0f95a1e6-7044-487e-8681-b1d717caabb3(Office.14).aspx*).

In this article:

Create a site collection by using Central Administration

Create a site collection by using Windows PowerShell

Create a site collection by using Central Administration

You typically use the Central Administration Web site to create a site collection in a stand-alone deployment.

To create a site collection by using Central Administration

1. Verify that you have the following administrative credentials:
   - To create a site collection, you must be a member of the Farm Administrators SharePoint group on the computer that is running the SharePoint Central Administration Web site.
2. On the Central Administration Web site, in the **Application Management** section, click **Create site collections**.
3. On the Create Site Collection page, in the **Web Application** section, if the Web application in which you want to create the site collection is not selected, on the **Web Application** menu click **Change Web Application**, and then click the Web application in which you want to create the site collection.
4. In the **Title and Description** section, type the title and description for the site collection.
5. In the **Web Site Address** section, select the path to use for your URL (for example, a wildcard inclusion path such as /sites/, or the root directory (/).

   If you select a wildcard inclusion path, you must also type the site name to use in your site's URL.
6. In the **Template Selection** section, in the **Select a template** list, select the template that you want to use for the top-level site in the site collection, or click the **Custom** tab to create an empty site and apply a template later.
7. In the **Primary Site Collection Administrator** section, type the user name (in the form DOMAIN\username) for the user who will be the site collection administrator.
8. In the **Secondary Site Collection Administrator** section, type the user name for the secondary administrator of the site collection.

   Designating a secondary site collection administrator is a best practice to ensure that someone can manage the site collection when a primary site collection administrator is not present.
9. If you are using quotas to manage storage for site collections, in the **Quota Template** section, click a template in the **Select a quota template** list.
10. Click **OK**.

Create a site collection by using Windows PowerShell

You typically use Windows PowerShell to create a site collection when you want to automate the task, which is common in enterprises.

To create a site collection by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. From the Windows PowerShell command prompt (that is, PS C:\>), type the following command and press ENTER:

```
Get-SPWebTemplate

$template = Get-SPWebTemplate "STS#0"

New-SPSite -Url "<URL for the new site collection>" -OwnerAlias "<domain\user>" -
Template $template
```

   This example retrieves a list of all available site templates and then creates a site collection by using the Team Site template. For more information, see New-SPSite (*http://technet.microsoft.com/library/ebdadc86-0cda-49b7-a84a-5cfc6b4506b3(Office.14).aspx*) and Get-SPWebTemplate (*http://technet.microsoft.com/library/dfd10bac-c304-4f3f-bea9-eb0af5f96df5(Office.14).aspx*).

   We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

Uninstall SharePoint Server 2010

This article describes how to uninstall Microsoft SharePoint Server 2010.

You remove SharePoint Server 2010 by uninstalling it from Control Panel. When you uninstall SharePoint Server 2010, most files and subfolders located in the installation folders are removed. However, Web.config files, index files, log files, and customizations that you might have are not automatically removed when you uninstall SharePoint Server 2010. Also, Microsoft SQL Server databases are detached, but are not removed from the database server. If you uninstall a single server that has a built-in database, Microsoft SQL Server Express is not removed. When you uninstall SharePoint Server 2010, all user data remains in the database files.

Before you perform this procedure, confirm that you have uninstalled any language packs that you might have on the server.

Uninstall SharePoint Server 2010

Use this procedure to uninstall SharePoint Server 2010.

To uninstall SharePoint Server 2010

1. Verify that you are a member of the Farm Administrators group or a member of the Administrators group on the local computer.
2. On the computer that runs SharePoint Server 2010, log on as a local or domain administrator.
3. Click **Start**, and then click **Control Panel**.
4. In the **Programs** area, click **Uninstall a program**.
5. In the **Uninstall or change a program** dialog box, click **Microsoft SharePoint Server 2010**.
6. Click **Change**.
7. On the Change your installation of Microsoft SharePoint Server 2010 page, click **Remove**, and then click **Continue**. A confirmation message appears.
8. Click **Yes** to remove SharePoint Server 2010. A warning message appears.
9. Click **OK** to continue. A confirmation message appears.
10. Click **OK**.

    You might be prompted to restart the server.

Note:

If you did not remove the language template packs before you uninstalled and then reinstalled SharePoint Server 2010, you must run **Repair** from the SharePoint Products Configuration Wizard for each language template pack on the server. After the repair operation is complete, you must restart the server. Finally, complete the language template pack configuration by running the SharePoint Products Configuration Wizard.

See Also

Add a Web or application server to the farm (SharePoint Server 2010)
(*http://technet.microsoft.com/library/0926f63d-8dae-44c0-9e91-51209aa4c3ef(Office.14).aspx*)

Remove a server from the farm (SharePoint Server 2010)
(*http://technet.microsoft.com/library/acb2d589-6c19-4b6b-9cd1-5e334ba4f2cd(Office.14).aspx*)

Hardware and software requirements (SharePoint Server 2010)

Deploy a single server with a built-in database (SharePoint Server 2010)

Deploy a single server with SQL Server (SharePoint Server 2010)

Multiple servers for a three-tier farm (SharePoint Server 2010)

Deploy language packs (SharePoint Server 2010)

Deploy customizations - overview (SharePoint Server 2010)

The articles in this chapter describe how to deploy site elements that have been customized by developers or Web designers in a Microsoft SharePoint Server 2010 environment.

In this article:

- Process overview
- Before you begin
- About the two kinds of customizable site elements
- Deploying developed site elements
- Deploying authored site elements

Process overview

Deploying customizations can be quite complex, particularly because there are many deployment methods available in SharePoint Server 2010, and the advantages of using one method over another are not always obvious.

You deploy these different types of site elements, or *artifacts*, by using different methods. You cannot deploy the full range of customizable site elements by using a single deployment method. There are other unique deployment considerations that apply to each type of element because they are likely to originate from different groups of designers, and because they are subject to different upgrade considerations. The different kinds of site elements are described in About the two kinds of customizable site elements, later in this article.

For specific deployment tasks and related considerations, see the following articles:

- Deploy solution packages (SharePoint Server 2010)
- Deploy authored site elements (SharePoint Server 2010)
- Deploy site elements by using Features (SharePoint Server 2010)
- Deploy templates (SharePoint Server 2010)
- Workflow deployment process (SharePoint Server 2010)

Before you begin

Before you deploy any custom code to the environment, you should establish a baseline of the environment's performance so that you can analyze how customizations affect performance. After you have established a performance baseline, test the custom code thoroughly in a test or integration environment and compare the results with the baseline. Make sure that you thoroughly test all customizations before you deploy them to the production environment.

You should also test any code that you acquire from third parties before you deploy it to the production environment, even if you acquire it from a trusted source.

The descriptions and guidance in these articles apply to a SharePoint Server environment that has been deployed and configured to meet the requirements in Plan for server farms and environments (SharePoint Server 2010) (*http://technet.microsoft.com/library/fd081152-5fdd-4e47-9afb-c887fbf1ce22(Office.14).aspx*).

About the two kinds of customizable site elements

Developed site elements are solution artifacts and are typically created by developers. A solution can include assemblies, which are SharePoint components that are written in Microsoft .NET Framework–based languages and compiled before being deployed. Developed site elements, except timer jobs assemblies and site definitions, are typically grouped into Features and deployed as part of a solution package. Developed site elements include:

- Web Parts
- Workflows
- Site and list definitions
- Document converters
- Event receivers
- Timer jobs
- Assemblies

Authored site elements, which are typically created by Web designers, are not explicitly compiled and reside in a content database. Authored site elements include:

- Master pages
- Cascading style sheets
- Forms
- Layout pages

These two kinds of customizable site elements are differentiated by:

- Where the files are stored in a SharePoint Server 2010 farm.
- Which team in the organization is responsible for administering the site element.
- What deployment mechanism the site element requires.

Some elements can be either solution artifacts or authored artifacts. For example, a content type can be defined in an XML file as a developed solution artifact, or created through a browser as an authored artifact. Site elements that can be solution artifacts or authored artifacts include site columns and list instances. Also, solution artifacts can be used to provision files into Web sites and set to be cached in memory on the front-end Web server.

Deploying developed site elements

Developed site elements can be generally defined as site elements that are created in a code-development environment and are deployed directly to front-end Web servers and application servers. These site elements are customized typically by developers by using Microsoft Visual Studio 2010 Tools for SharePoint 2010, Microsoft Office SharePoint Designer, or XML editing tools. For more information, see SharePoint Foundation Development Tools (*http://go.microsoft.com/fwlink/?LinkId=183360*).

Note:
This article does not discuss the deployment of developed site elements that are deployed as sandboxed solutions. Sandboxed solutions are solutions that can access a subset of the server object model and a subset of feature elements that site collection administrators can deploy. For more information, see Sandboxed solutions overview (SharePoint Server 2010) (*http://technet.microsoft.com/library/cf2b1b11-8e35-4269-a8c7-96fde034947f(Office.14).aspx*).

A best practice is to use solution packages and Windows PowerShell to deploy developed site elements. The SharePoint Server Solution Framework simplifies and standardizes the process of deploying new and upgrade site elements across the farm and synchronizing a front-end Web server so that its state is consistent with the state of other servers in the farm. For example, solution packages simplify the process of rebuilding a farm. Deploying site elements by manually handling code and files can lead to inconsistencies in the upgrade process and can result in servers that are not synchronized with other servers. You can use solution packages to deploy developed site elements from developer environments to integration farms and then to staging, pilot, and production farms.

You can use Windows PowerShell cmdlets to create, import, export and provision solution packages, which leverage the Solution Framework to distribute developed site element customizations. Windows PowerShell cmdlets are useful for deployment of site customizations in most environments because they are included with both SharePoint Server 2010 and SharePoint Foundation 2010, and you can use them alone or in conjunction with other methods. You can use Windows PowerShell cmdlets to deploy both artifacts and developed site elements. You can also use cmdlets to activate Features that are deployed in a solution package.

Deploying authored site elements

Authored site elements differ from developed site elements in that they are stored in the content database, although they can depend on resources that exist in the file system of Web servers or — less typically — application servers. In some cases, authored site elements do not function because they require that developed site elements be deployed first.

In environments where customization deployments are entirely automated, the required deployment order can be enforced by the system to eliminate synchronization issues. However, if customization deployment is partially or wholly manual, you must ensure that all required resources are in place on the Web servers and application servers before you deploy any content that relies on those resources.

You deploy authored site elements from authoring environments to staging, pilot, and production farms by using one or more of several different systems. The following table describes these systems and their associated interfaces and usage scenarios.

Deployment system	Usage scenario
SharePoint Central Administration Web site	In environments where source and destination farms are connected by a network, you can use the content deployment features in Central Administration to create a content deployment package on the source farm and export the package to another farm. This method is easy to configure and use, and can be used to automate deployment of authored site elements with very little setup time and maintenance.
Content Migration object model	Depending on the method you use (programming by using the deployment namespace APIs, using Simple Object Access Protocol (SOAP) calls to a Web service, or moving a whole site by using Windows PowerShell cmdlets), you can control what content is migrated and how. Using the API to import and export content is the only supported method that retains globally unique identifiers (GUIDs). For more information, see Content Migration (*http://go.microsoft.com/fwlink/?LinkId=183372*).
Windows PowerShell	You can use Windows PowerShell cmdlets to perform import and export operations against the whole site, preserving time stamps, security information, and user information. Windows PowerShell cmdlets are most useful when you want to move basic content from a whole Web site. Windows PowerShell is useful for deployment of site customizations in most environments because it is included with SharePoint 2010 Products, and you can use it alone or with other methods. You can use Windows PowerShell cmdlets to deploy both artifacts and developed site elements. For more information, see SharePoint 2010 Products administration by using Windows PowerShell (*http://technet.microsoft.com/library/ae4901b4-505a-42a9-b8d4-fca778abc12e(Office.14).aspx*).
Custom Web service	You can create a custom Web service that automates content migration and deployment. You can write custom scripts and Windows applications to execute specific tasks within this process. For more information about programmatic methods for writing a custom Web service, see the following resources in the Microsoft SharePoint 2010 Software Development Kit (SDK): • Sites Methods (*http://go.microsoft.com/fwlink/?LinkID=183373*)

Deployment system	Usage scenario
	• Sites.ExportWeb Method (*http://go.microsoft.com/fwlink/?LinkId=183377*) • Sites.ImportWeb Method (*http://go.microsoft.com/fwlink/?LinkId=183378*)
Manual code handling	In smaller, disconnected environments, or in environments in which authored site elements are not continually customized, you can manually deploy site elements and related resources. In smaller connected environments, consider using the content deployment features in Central Administration to deploy authored site element customizations.
Solution packages and Features	Elements such as page layouts, master pages, forms, and style sheets, can be grouped and deployed in Features as part of a solution package. Features deployed from a solution package can be activated on the scopes where authored elements need to be provisioned. For more information, see Deploy site elements by using Features (SharePoint Server 2010).
Custom templates	A user can save an existing site, with or without its specific content, as a custom template. This provides a means for reusing customized sites. A custom site template is stored as a .wsp file. Site templates are saved in the Solution Gallery of the top-level site in a site collection, where they become available for subsite creation on all Web sites in the site collection. Site templates can be downloaded and moved to other site collection galleries.

See Also

Deploy solution packages (SharePoint Server 2010)

Deploy authored site elements (SharePoint Server 2010)

Deploy templates (SharePoint Server 2010)

Deploy templates (SharePoint Server 2010)

Deploy solution packages (SharePoint Server 2010)

This article describes solution packages and the role they play in deploying authored and developed customizations in Microsoft SharePoint Server 2010. It includes procedures for importing and deploying solution packages, and an example for building and deploying a solution package by using Microsoft Visual Studio 2010.

In this article:

- What is a solution package?
- Deploying site elements by using solution packages
- Creating and deploying a custom Web Part solution package by using Visual Studio 2010

What is a solution package?

A *solution package* is a distribution package that delivers your custom SharePoint Server 2010 development work to the Web servers or the application servers in your server farm. Use solutions to package and deploy custom Features, site definitions, templates, layout pages, Web Parts, cascading style sheets, and assemblies.

This article does not discuss the deployment of sandboxed solutions. You can deploy a Microsoft SharePoint Server 2010 solution directly onto your SharePoint Server farm, or you can deploy the solution into a *sandbox*. A sandbox is a restricted execution environment that enables programs to access only certain resources, and that keeps problems that occur in the sandbox from affecting the rest of the server environment. For more information, see Sandboxed solutions overview (SharePoint Server 2010) (*http://technet.microsoft.com/library/cf2b1b11-8e35-4269-a8c7-96fde034947f(Office.14).aspx*).

A solution package is a CAB file with a .wsp file name extension and a manifest file. It is recommended that you use Visual Studio 2010 Tools for SharePoint 2010 to develop and package SharePoint solutions. You can also create solution packages manually by using a tool such as Makecab.exe.

Components that can be packaged in a solution include:

- .NET Framework assemblies, typically Web part assemblies and event receiver assemblies.
- Deployment files such as resource files, pages, or other helper files.
- Features, which allow you to activate and deactivate code in a Web site and provide functionality that include elements such as custom lists, libraries, fields, and content types.
- New templates and site definitions.

- Configurations that must be performed at the Web-server level — for example, deploying customizations to the Web.config files for the registration of Web Parts. You can also modify these configurations with a Feature that is distributed with a Feature.
- Web content such as Web pages and images that are called by Web pages. If you must deploy Web content in a disconnected environment, you should use a content deployment package.

Deploying site elements by using solution packages

In this section:

- When to use solution packages
- Deploying farm solutions
- Adding a solution package
- Deploying a solution package
- About creating a solution package

When to use solution packages

A best practice for deploying customizations is to use solution packages as part of a straightforward, safe, and consistent application lifecycle management process. Solution packages make it simpler to change the Features and functionality of the Web sites after the sites are created.

You can use solution packages to deploy new solutions and upgrade existing solutions across the farm. You can package all your SharePoint Server entities as one file, add the file to the solution store, and deploy it to the front-end Web servers in the farm. Use solution packages to synchronize a front-end Web server so that its state is consistent with the state of other Web servers in the farm.

You can use solution packages to deploy authored site element customizations from an integration farm to authoring, pilot, or production farm. In SharePoint Server, users can save a customized site as a template. This creates a solution package with a .wsp file name extension that can be deployed on another farm.

You can use solution packages to deploy customizations between these environments:

- From developer workstations, to an integration farm or a software configuration management system
- From an integration farm and authoring client workstations, to pilot or production farms

Deploying farm solutions

Farm solutions are deployed either locally or by using a timer service. Both local and timer-based deployments can be triggered either by using command-line instructions or programmatically by using the object model.

Local deployment

In a local deployment, solution files are deployed only to the computer from which the deployment operation was initiated. The solution is not marked as "deployed" in the configuration database until the solution files are deployed to all applicable servers in the server farm. Then solution features are installed, and schema and definition files are committed to the configuration store.

Timer Service deployments

In deployments that use the timer service, the deployment creates a timer job. This timer job is picked up by the timer service on each Web server in the server farm. Initially, manifest and feature manifests are parsed to find assembly and _layouts files, which are copied to the appropriate locations. All other files contained within a feature directory are copied to the feature directory. After solution files are copied to the target computers, a configuration reset is scheduled for all front-end Web servers; the reset then deploys the files and restarts Microsoft Internet Information Services (IIS). Solution features are then registered, and schema and definition files are committed to the configuration store.

For more information about the solution store, deployment, and synchronization, see Deploying a Solution (*http://go.microsoft.com/fwlink/?LinkId=186995*) in the Microsoft SharePoint 2010 Software Development Kit (SDK)

Adding a solution package

Before you can deploy a solution package, you must add it to the solution database of a SharePoint Server farm.

Important:
You must be a member of the Administrators group on any computer on which you run Windows PowerShell.

To import a solution package by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin..
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

```
Add-SPSolution -LiteralPath <SolutionPath>
```

The solution is added to the farm's solution store. To use the solution, follow the procedure in the next section in this article. For more information, see Add-SPSolution (*http://technet.microsoft.com/library/0c64c1ac-39c0-4d5e-923f-27d0c48b006a(Office.14).aspx*).

Deploying a solution package

You can deploy imported solutions by using the Central Administration Web site or by using Windows PowerShell. After a solution has been added to the solution store by using the Windows PowerShell**Add-SPSolution** cmdlet, it must be deployed to a site before it can be accessed.

Note:
You cannot add a solution to the solution store by using the Solution Management page in Central Administration.

The following procedures show how to deploy an imported solution to a site in the farm by using either the Central Administration Web site or Windows PowerShell.

To deploy a solution by using Central Administration

1. On the Central Administration Home page, click **System Settings**.
2. In the **Farm Management** section, click **Manage farm solutions**.
3. On the Solution Management page, click the solution that you want to deploy.
4. On the Solution Properties page, click **Deploy Solution**.
5. On the Deploy Solution page, in the **Deploy When** section, select one of the following:
   - **Now**
   - **At a specified time**. If you select this option, specify a time by using the date and time boxes. We recommend that you select a time when the load on the destination servers is low.
6. In the **Deploy To?** section, in the **A specific web application** list, click either **All web applications** or select a specific Web application.
7. Click **OK**.

To deploy a solution package to a single Web application by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

   ```
   Install-SPSolution -Identity <SolutionName> -WebApplication <URLname>
   ```

   Where:

   - *<SolutionName>* is the name of the solution.
   - *<URLname>* is the URL of the Web application to which you want to deploy the imported solution.

   By default, the solution is immediately deployed. You can also schedule the deployment by

using the **time** parameter. For more information, see Install-SPSolution (*http://technet.microsoft.com/library/0133c53b-70c4-4dff-a2ae-3c94759ed25d(Office.14).aspx*).

To deploy a solution package to all Web applications by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

```
Install-SPSolution -Identity <SolutionName> -AllWebApplications -time
<TimeToDeploy> -GACDeployment -CASPolicies
```

   Where:

   - **GACDeployment** is the parameter that enables SharePoint Server 2010 to deploy the assemblies in the global assembly cache.
   - **CASPolicies** enables the creation of a custom code access security (CAS) policy file and the activation of it in the Web.config file of the targeted site collection.

   The solution is immediately deployed by default. You can also schedule the deployment by using the **time** parameter.

About creating a solution package

SharePoint Server 2010 does not include a tool for creating solution packages. This section describes ways to create solution packages that contain developed site elements and artifacts.

Visual Studio 2010

You can use Visual Studio 2010 to group related SharePoint elements into a Feature, and then bundle multiple Features, site definitions, assemblies, and other files into a single package (.wsp file) to deploy to servers running SharePoint Server 2010. You can use Visual Studio 2010 to debug and test the .wsp file on the server running SharePoint Server 2010. You can also customize the deployment steps on the development computer.

Developers can build their SharePoint solutions in Visual Studio 2010 and produce .wsp files by using the automated build process. Source code of the Visual Studio SharePoint project that is used to generate the .wsp file can also be added to the source code control system by using Visual Studio 2010 integration. Visual Studio 2010 can import .wsp files and create projects to further extend them and to create new .wsp files. The primary source of .wsp files that are imported into Visual Studio 2010 is templates saved from sites by using the Save As Template command on SharePoint Server 2010 sites. These templates can be used to save all of site customizations to a SharePoint solution.

For more information, see SharePoint Development in Visual Studio (*http://go.microsoft.com/fwlink/?LinkId=187000*).

Makecab

Solution packages can be created manually by using tools such as Makecab.exe. The Makecab.exe tool takes a pointer to a .ddf file, which describes the structure of the CAB file. The format of a .ddf file is similar to that of an .inf file — that is, you declare a standard header and then enumerate, one file per line, the set of files by where they are located on disk and where they should be located in the CAB file.

The Makecab.exe tool is available for download in the Microsoft Cabinet Software Development Kit (*http://go.microsoft.com/fwlink/?LinkId=107292*).

About customizing solution packages

If you need to make any of the following customizations in SharePoint Server 2010 solutions, it is recommended that you use Visual Studio 2010 to customize solution packages. You can also make these customizations by manually creating SharePoint solution packages.

- Deploy .NET Framework assemblies in the private application folder instead of the global assembly cache.
- Add code access security permissions to the solution that must be applied during the deployment.
- Deviate from the names used by default for the Feature folders.
- Localize the solution.
- Associate Feature event handlers to certain types of SharePoint Foundation 2010 solutions, such as Web Part solutions.
- Add resources (XML files, pictures, .dll files, and assemblies) to the solution package.

Manually create a solution file

In most SharePoint Server 2010 development scenarios, we recommend that you use Visual Studio 2010 Tools for SharePoint 2010 to develop and package SharePoint solutions. In Visual Studio 2010, the deployment process copies the .wsp file to the server running SharePoint Server 2010, installs the solution, and then activates the Features.

You can also manually create a solution file. The following are the basic steps of creating a solution file:

1. Collect all individual solution files in a folder. There are no concrete guidelines about how you should do this, but a best practice is to separate the different types of solution files into their own subfolders.
2. Create a manifest.xml file that lists the components of the solution.
3. Create a .ddf file that defines the structure of the solution file. This file contains the list of individual solution files that determine the output .wsp file.
4. Execute Makecab.exe with the .ddf file as input and the .wsp file as output.

About the solution manifest file

The solution manifest (always called manifest.xml) is stored at the root of a solution file. This file defines the list of Features, site definitions, resource files, Web Part files, and assemblies to process. It does

not define the file structure — if files are included in a solution but not listed in the manifest.xml file, they are not processed in any way.

Following is an example of the structure of a manifest.xml file, shown in XML.

```
<?xml version="1.0" encoding="utf-8" ?>
<Solution xmlns="http://schemas.microsoft.com/sharepoint/"
SolutionId="{79d1a62e-3627-11db-963e-00e08161165f}"
ResetWebServer="TRUE">

    <Assemblies>
        <Assembly DeploymentTarget="GlobalAssemblyCache"
Location="Example.Sharepoint.Webparts\
Example.SharePoint.WebParts.dll">
            <SafeControls>
                <SafeControl Assembly="Example.Sharepoint.Webparts,
Version=1.0.0.0, Culture=Neutral, PublicKeyToken=63cce650e8605f5d"
Namespace="Example.Sharepoint.Webparts" TypeName="*"/>
            </SafeControls>
        </Assembly>
        <Assembly DeploymentTarget="GlobalAssemblyCache"
Location="Example.Sharepoint.Timer/Example.Sharepoint.Timer.dll"/>
    </Assemblies>

    <FeatureManifests>
        <FeatureManifest Location="Example.Sharepoint.Timer\Feature.xml"/>
        <FeatureManifest Location="Example.CustomType\Feature.xml"/>
        <FeatureManifest Location="Example.ExampleLibrary\Feature.xml"/>
        <FeatureManifest Location="Example.Columns\Feature.xml"/>
        <FeatureManifest Location="Example.Workflow.ProcessExample\Feature.xml"/>
        <FeatureManifest Location="Example.Workflow.ProvisionExample\Feature.xml"/>
    </FeatureManifests>

    <SiteDefinitionManifests>
        <SiteDefinitionManifest Location="EXAMPLE">
            <WebTempFile Location="1033\XML\WEBTEMPExample.XML"/>
        </SiteDefinitionManifest>
```

```
    </SiteDefinitionManifests>
</Solution>
```

In addition, you can add a **DwpFiles** element to specify .webpart or .dwp files, or a **ResourceFiles** element to specify resource files, site definitions, application resources, and code access security policies.

Optionally, annotate your Feature.xml files by using **<ElementFile>** tags.

If your solution contains Features, in the **<ElementManifests>** tag in your Feature.xml file, add `<ElementFile Location="..."/>` for all the extra files in your feature, such as ASP.NET pages (for example, allitems.aspx) or master pages, and so on.

For more information about solution manifest files that define the constituent parts of a solution, see Solution Schema (*http://go.microsoft.com/fwlink/?LinkID=183466*).

Creating and deploying a custom Web Part solution package by using Visual Studio 2010

For an example walkthrough that shows you how to use Visual Studio 2010 to create, customize, debug, and deploy a SharePoint list definition to track project tasks, see Walkthrough: Deploying a Project Task List Definition (*http://go.microsoft.com/fwlink/?LinkID=189612*) in the MSDN Library.

This walkthrough illustrates the following tasks:

- Creating a SharePoint list definition project that contains tasks.
- Adding the list definition to a SharePoint Feature.
- Adding an event receiver to the list.
- Creating and customizing a SharePoint package to deploy your Feature.
- Building and deploying your SharePoint solution.

When you build the sample project in this walkthrough, Visual Studio 2010 automatically deploys the solution to the server running SharePoint Server 2010 on your development computer for testing and debugging. You can also create a solution package file that you can add and deploy on another computer. For more information, see How to: Deploy a SharePoint Solution (*http://go.microsoft.com/fwlink/?LinkID=187004*). You can use the **Add-SPSolution**Windows PowerShell cmdlet to import the solution to another computer.

You can use the **Solution Management** page in Central Administration to deploy the solution package. Alternatively, you can use the **Install-SPSolution**Windows PowerShell cmdlet to deploy the solution package.

In the walkthrough, the scope of the project list feature is Web. To activate the Feature, on the Web site, expand the **Site Actions** menu, and then click **Site Settings**. Under **Site Actions**, click **Manage site features**. On the Features page, next to the Feature name, click **Activate**.

Deploy authored site elements (SharePoint Server 2010)

This article discusses the deployment of authored site element customizations in Microsoft SharePoint Server 2010, including deployment procedures, general considerations, and best practices related to deploying custom content.

In this article:

- About deploying authored site elements
- Before you begin
- Deploy content by using content deployment jobs
- Deploy content by using the Content Migration API
- Create a content deployment package by using Windows PowerShell

About deploying authored site elements

Authored site elements can be thought of as the "content" in your sites. They are the Web pages, images, layout pages, cascading style sheets, and other resources that compose your SharePoint Server 2010 Web site. Authored site elements include:

- **Artifacts** These are site elements — typically authored by using a design tool such as Microsoft SharePoint Designer 2010 — that compose the framework in which your site's content appears. Examples of artifacts include master pages and layouts.
- **Web content** These are site elements — typically authored directly in the Web browser or in a client authoring program such as Word 2010 — that supply the content of your site. Examples of Web content include Web pages and images.

This article does not discuss deployment of developed site elements such as Web Parts and other code. For more information, see Deploy solution packages (SharePoint Server 2010) and Deploy templates (SharePoint Server 2010).

Authored site elements can be deployed by various methods:

- Content deployment is a built-in system that enables you to create content deployment jobs in the Central Administration Web site. Content deployment is intended for regularly updating content or moving content to a destination site collection.
- Use the object model to handle scenarios such as writing scripts to automate common tasks and setting custom properties for export and import that tailor the deployment. The object model provides the most control over your data migration scenarios.
- Content deployment packages are intended for a one-time move or migration of content to a destination site collection. Content deployment packages are CAB files that can contain part or all

of the authored site elements in a Web site, and can be deployed in a disconnected environment. Windows PowerShell cmdlets are used to create content deployment packages.

Note:
This article does not discuss using solution packages to deliver your custom SharePoint Server 2010 development work to the front-end Web servers or the application servers in your server farm. By using solution packages, you can deploy artifacts in a disconnected environment, and you can deploy artifacts and developed site elements in the same package. For more information, see Deploy solution packages (SharePoint Server 2010).

When to use the content deployment feature

The content deployment feature in SharePoint Server 2010 provides a simple and effective method of automating the deployment of content from one farm to another. You can also use content deployment to move artifacts and content between a staging environment and the production environment in a scenario where content is staged before it is published.

You can use the content deployment feature to deploy authored site elements in one or both of the following scenarios:

- **Connected environments** The content deployment feature can only be used in a connected environment — that is, in an environment in which the source and destination farms can communicate over a network.
- **Automated environments** If you want to automate the deployment of authored site elements from one server farm to another, the content deployment feature provides a built-in system for setting up automated deployment jobs.

The content deployment feature can be used to deploy authored site element customizations between an authoring site and multiple target sites. A best practice is to deploy customizations between the authoring site and the staging site or pilot environment. If you verify through testing that all site elements function as expected in network and security conditions that match the production environment, you can deploy the customizations from the authoring site to the production environment.

Note:
Content deployment is always one-way. It is a "single master" system that always moves from a source farm or site collection to a destination farm or site collection.

For general guidance about how to plan to use content deployment with your SharePoint Server 2010 sites, see Plan content deployment (SharePoint Server 2010) (*http://technet.microsoft.com/en-us/library/cc263428(office.14).aspx*).

When to use a content deployment package

You can use content deployment packages to deploy authored site elements in one or more of the following scenarios:

- **One-time content migration** Use a content deployment package to move content to a destination site collection only once. If you plan to update content regularly on a destination site collection, use the content deployment feature or the Content Migration API.
- **Disconnected environments** If the farms are disconnected, you can create a content deployment package for asynchronous transfer to the integration farm.
- **Sample content** If authored site element customizations need to be deployed from the authoring environment to the integration environment to be used as samples for development purposes, you can use a content deployment package to simplify this process.

Before you begin

To eliminate potential synchronization issues, you must often deploy developed site elements before you deploy authored site elements. Farm solutions and Web application solutions must be installed and deployed to the destination farm prior to content deployment. Also be aware that you must install on the destination server any language packs that are in use on the source server; if you fail to install the required language packs, content deployment will fail.

Before performing the procedures in this article, familiarize yourself with the concepts related to the deployment of site element customizations. For more information about planning and designing sites and site collections, see Fundamental site planning (SharePoint Server 2010) (*http://technet.microsoft.com/library/d59e6849-84af-4ad0-abed-68fae92574c6(Office.14).aspx*). Also, download an Excel version of the Content deployment planning worksheet (*http://go.microsoft.com/fwlink/?LinkID=167835*).

Deploy content by using content deployment jobs

You can deploy content by using content deployment jobs between farms that are connected by a network. In many customization environments, content deployment jobs are run at regular intervals between farms. For example, every night a content deployment job may deploy all customizations that were developed on the authoring farm to the integration farm. This frees the authors and authoring farm administrators from the responsibility of manually migrating customizations, and you can time the migration to coincide with deployment of customizations from the developer workstations to the integration farm to ensure that customizations are deployed in the appropriate order.

Perform the following procedure to configure automated content deployment.

To configure automated content deployment

1. On the destination server farm, create an empty site to receive the initial content deployment job. Do not apply a template to the site that you create. In SharePoint Server 2010, you cannot deploy content to a site that has had any blank template applied to it.

2. On the destination farm, on the Content Deployment Settings page in SharePoint Central Administration, configure the farm to accept incoming deployment jobs, assign a front-end Web server as the import server to manage incoming deployment jobs, and specify whether to require encryption on the connection between the source and destination farms.
3. On the source farm, on the Content Deployment Settings page, assign a Web server as the export server to manage outgoing deployment jobs.
4. On the source farm, on the Manage Content Deployment Paths and Jobs page in Central Administration, create one or more deployment paths.
5. On the source farm, on the Manage Content Deployment Paths and Jobs page, create one or more deployment jobs for each path.
6. Run the initial deployment job to initiate the content on the destination farm.
7. If the path does not deploy all security information, you should replicate the source farm's users, SharePoint groups, and permissions on content and sites on the destination farm.

For more information about how to plan for automated content deployment, see Plan content deployment (SharePoint Server 2010) (*http://technet.microsoft.com/library/edcdacca-8013-460e-95a0-d2b83b6cc7ef(Office.14).aspx*).

Deploy content by using the Content Migration API

Most deployment scenarios can be accomplished by using Central Administration without the need for scripts. However, you can use the object model to handle other scenarios, such as writing scripts to automate common tasks and setting custom properties for export and import that you cannot configure you set up a deployment by using the SharePoint Central Administration site. You can also create code that exports and imports a content package in situations where connectivity between a source farm and a destination farm may be limited or unavailable.

For more information about content migration and the content migration APIs, see Content Migration Overview (*http://go.microsoft.com/fwlink/?LinkId=187033*). For an overview of the content deployment feature and the background and resources necessary to build and implement custom deployment solutions, see Deploying Content Between Servers (*http://go.microsoft.com/fwlink/?LinkID=181466*). For a code example that shows how to use the object model to create paths and jobs that deploy content between site collections, see How to: Deploy Content Between Servers (*http://go.microsoft.com/fwlink/?LinkId=187034*). For a code sample and information about how to export and import a content package by using the Content Migration API, see How to: Customize Deployment for Disconnected Scenarios (*http://go.microsoft.com/fwlink/?LinkID=181076*).

Create a content deployment package by using Windows PowerShell

You can use Windows PowerShell to create a content deployment package that contains the authored site elements for a whole site (including all the content in the site) or a list or a document library.

Note:
Use content deployment packages for a one-time migration of content to a destination site collection. Use the content deployment feature or the Content Migration API to periodically move content from a source site collection to a destination site collection.

Content deployment packages are implemented as CMP (Content Migration Package) files. You export this package from the source server, and then import it into the destination server. You can use this method of content deployment in both connected and disconnected environments.

If you are using a software configuration management system, follow the steps for exporting the content deployment package, and then use the procedure appropriate to your software configuration management system to save the exported file.

To create a content deployment package by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

```
Export-SPWeb -Identity <URLname> -path <ExportFileName> -IncludeUserSecurity -
IncludeVersions 4 -NoFileCompression
```

   Where:

   - *<URLname>* is the site to export. This site is written to the content deployment package together with all the subsites beneath it.
   - **IncludeUserSecurity** causes the new site to have the same permissions as the original site.
   - **IncludeVersions** is set to 4 to specify that all versions should be included.
   - **NoFileCompression** causes the content deployment package to be output to an uncompressed folder instead of a single CAB file. This makes it more complicated to deploy the deployment package to a different server, but makes it easier to edit the individual files.

   For more information, see Export-SPWeb (*http://technet.microsoft.com/library/cd85bf19-6f24-4f13-bd9c-37bbf279ea2b(Office.14).aspx*).

Note:

We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

To import a content deployment package by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

   ```
   Import-SPWeb -Identity <URLname> -path <ImportFileName> -IncludeUserSecurity
   ```

   Where:

   - *<URLname>* is the site that will be imported, together with all the subsites beneath it.

For more information, see Import-SPWeb (*http://technet.microsoft.com/library/2ecc5b6e-1b23-4367-a966-b7bd3377db3a(Office.14).aspx*).

Note:

We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

Deploy site elements by using Features (SharePoint Server 2010)

This article describes how to deploy developed site element customizations by using Features. By using Features, you can control the scope within which the site customization can be activated and deactivated, and easily deploy the customizations across multiple server farms.

In this section:

- What is a Feature?
- When to use Features
- Create a Feature
- Install and activate a Feature by using Windows PowerShell

What is a Feature?

A *Feature* is a container of various defined extensions for SharePoint Server 2010, and is composed of a set of XML files that are deployed to front-end Web servers and application servers. You can deploy a Feature as part of a solution package, and you can individually activate a Feature in SharePoint Server sites.

Features reduce the complexity involved in making simple site customizations. Features eliminate the need to copy large chunks of code to change simple functionality, and therefore they reduce versioning and inconsistency issues that can 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 turning on or turning off a particular Feature in the user interface.

An element is an atomic unit within a Feature. The **Feature** element is used in a Feature.xml file to define a Feature and to specify the location of assemblies, files, dependencies, or properties that support the Feature. A Feature includes a Feature.xml file and any number of files describing individual elements. Another Feature element from a different schema is used in an Onet.xml file to specify that a Feature be activated when a site is created from the site definition.

A Feature.xml file typically points to one or more XML files whose top-level **<Elements>** tag contains definitions for elements that support the Feature. Elements in SharePoint Server 2010 often correspond to what were discrete nodes in the Onet.xml or Schema.xml file of Microsoft Office SharePoint Portal Server 2003. There are several types of elements —for example, a custom menu item or an event handler.

- For more information about the capabilities of Features, see Using Features (*http://go.microsoft.com/fwlink/?LinkId=183450*).

- For specific information about the file format and XML elements used in the Feature.xml file, see Feature.xml Files (*http://go.microsoft.com/fwlink/?LinkId=183451*).
- For information about how features affect the file format of the Onet.xml file in a site definition, see Site Definition (Onet.xml) Files (*http://go.microsoft.com/fwlink/?LinkId=183454*).
- For more information about Feature element types, see Element Types (*http://go.microsoft.com/fwlink/?LinkId=183455*).

When to use Features

Features are the recommended method for deploying pieces of functionality, customizations, or configuration changes to front-end Web servers. Features are a flexible way to manage functionality through its lifecycle, including activation, upgrade, and eventually deactivation.

You can use Features to deploy developed site elements in one or more of the following scenarios:

- **Need for activation and deactivation** When you deploy site element customizations in a Feature, you can install, activate, and deactivate the Feature by using Windows PowerShell or by using the object model. You can also activate and deactivate a Feature by using the Central Administration Web site.
- **Flexibility of scope** You can activate a Feature for a single scope, including farm, Web application, site collection, or Web site.
- **Ease of distributed deployment** A Feature is easy to deploy to multiple server farms as part of a solution.
- **Control through the Feature object model** The Feature object model enables you to specify the list of installed features within a given scope and to control whether features are enabled at the farm and site levels.

Use solution packages to package Features to deploy to different environments. For example, use a solution package to deploy customizations between developer workstations and an integration farm, and also between either an integration farm or authoring client workstations, and pilot or production farms.

Create a Feature

When you create a custom Web page in SharePoint Server 2010 by using the browser or SharePoint Designer, the ASPX page can belong only to the root site collection of the server that is running SharePoint Server 2010. To create a page under a site collection that is available to the whole farm and in all site collections, use a solution to deploy the page under the \14\Template\Layouts folder (by using the **TemplateFiles** element in the solution manifest file).

A best practice on a farm is to deploy Features by using a solutions package. If a server must be rebuilt or another server is added to the farm, the Feature will not have to be manually added to each front-end Web server. By using solutions packages, you can deploy new and upgraded Features across the farm

and synchronize a front-end Web server so that its state is consistent with the state of other servers in the farm.

To control the availability of a custom page in a site collection or a Web site, deploy the custom Web page as a SharePoint Feature as part of a solution. Use the module element in the Feature.xml file to deploy a Web page by using a scope of site collection and Web site. Modules are frequently used to implement a Web Part Page in the site. A Feature that is deployed as part of a solution is installed automatically. If you manually deploy a Feature, you must install and activate it. See Install and activate a Feature by using Windows PowerShell (*http://technet.microsoft.com/library/b0f3e532-48e3-4429-aacd-20750cb9f0df.aspx#InstallAndActivate*), later in this article.

To create and deploy a custom Feature

1. Create a Feature.xml file. The following is an example Feature.xml file, which is necessary for giving the feature a unique ID and pointing to the Module.xml file.

```
<?xml version="1.0"?>

<Feature Id="8C4DD0CB-5A94-44da-9B7F-E9ED49C2B2DC" Title=

"Custom Web page"

Description="This simple example feature adds an ASPX page

with a hosted XmlFormView control"

Version="1.0.0.0" Scope="Web"

xmlns="http://schemas.microsoft.com/sharepoint/">

<ElementManifests>

    <ElementManifest Location="Module.xml"/>

</ElementManifests>

</Feature>
```

2. Create a Module.xml file. The following is an example Module.xml file, which contains information about the page or pages that are part of the solution.

```
<?xml version="1.0"?>

<Elements xmlns="http://schemas.microsoft.com/sharepoint/">

    <module name="file" url="" path="">

        <file url="XmlFormViewPage.aspx" type="ghostable"> </file>

    </module>

</Elements>
```

3. Change the file **url** value to the name of your ASPX page.
4. Add a subfolder for the Feature definition within the Features setup directory on the server computer, typically located at %COMMONPROGRAMFILES%\Microsoft shared\Web server extensions\14\TEMPLATE\FEATURES.

Important:

A best practice is to use detailed, qualified names for the subfolders that you create for Feature definitions. This practice minimizes the likelihood that you will add multiple Features that have the same names and overwrite the Feature.xml file for another Feature. For example, use **HR_Contract** and **Finance_Contract** rather than **Contract**.

5. Add your custom .aspx page to this subfolder for the Feature definition.
6. Add Feature.xml and Module.xml files to the same location.
7. Add the Feature to a solution package.

   You can use Visual Studio 2010 to add the Feature to a solution, or you can manually add a **FeatureManifests** element to the solution Manifest.xml file.
8. Create the solution package.

   You can use Visual Studio 2010 to build the solution package. You can also use the Makecab.exe tool to create the solution package.
9. Import and deploy the solution package.

   Add the solution to the solution store by using the Windows PowerShell**Add-SPSolution** cmdlet, and then deploy the solution from the solution store by using the Central Administration Web site or by using Windows PowerShell.

For more information about using Visual Studio 2010 to add Features to a solution packages, see Creating SharePoint Solution Packages (*http://go.microsoft.com/fwlink/?LinkId=187035*). For more information about manually creating a solution package or using the Makecab.exe tool to make the package, see Creating a Solution (*http://go.microsoft.com/fwlink/?LinkId=187036*).. For more information about deploying solutions, see Deploy solution packages (SharePoint Server 2010).

Install and activate a Feature by using Windows PowerShell

You can install and activate a Feature by using Windows PowerShell or by using the object model. You can also activate a Feature by using the Manage Web Applications Features page or the Features page of the site collection or site on which you want to activate the Feature. Installing a Feature makes its definition and elements known throughout a server farm, and activating the Feature makes the Feature available at a particular scope.

Note:

Features that are deployed as part of a solution package are installed by the deployment and manual installation is not required.

You install Features in the 14\Template\Features folder, with each Feature in its own subfolder. At the root of this folder, a Feature.xml file defines the contents of the Feature. You must install individual Features before you can use them, and —unless the Feature is scoped to the farm — you must

activate them after you install them. If a Feature is scoped to the farm or Web application, it is activated automatically.

To uninstall a Feature so that its definition is no longer available within a server farm, you first must deactivate the feature by using the Windows PowerShellDisable-SPFeature (*http://technet.microsoft.com/library/c10fbc69-088c-4e49-9005-fde54c035f23(Office.14).aspx*) cmdlet, unless the Feature is scoped for Web applications or farms. After you deactivate the Feature, you can use the **Uninstall-SPFeature** cmdlet to uninstall it. For more information, see Uninstall-SPFeature (*http://technet.microsoft.com/library/2f3831e4-b964-4e0e-bcc5-02659fdc0bb7(Office.14).aspx*). After uninstalling a Feature, reset Internet Information Services (IIS) so that the changes can take effect.

To deactivate a Feature so that it becomes inactive at its originally assigned scope without uninstalling it, you can use the **Disable-SPFeature** cmdlet. For more information, see Disable-SPFeature (*http://technet.microsoft.com/library/c10fbc69-088c-4e49-9005-fde54c035f23(Office.14).aspx*).

Use the following procedures to install and activate a Feature.

To install a Feature by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.
5. At the Windows PowerShell command prompt, type the following command:

   ```
   Install-SPFeature -path <Path> [-force]
   ```

   Where:

   - *<Path>* is a valid file path; for example, MyFeature. The path to the feature must be a literal path to the 14\Template\Features folder name. The Feature.xml file name is implied and does not need to be provided.

   For more information, see Install-SPFeature (*http://technet.microsoft.com/library/a1093d30-68a1-4c84-8454-967bda8d68b9(Office.14).aspx*).

   Note:
   We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

To activate a feature by using Windows PowerShell

1. Verify that you meet the following minimum requirements: See Add-SPShellAdmin.
2. On the **Start** menu, click **All Programs**.
3. Click **Microsoft SharePoint 2010 Products**.
4. Click **SharePoint 2010 Management Shell**.

5. At the Windows PowerShell command prompt, type the following command:

```
Enable-SPFeature -Identity <FeatureID> [-url] <URLname> [-force]
```

Where:

- <*FeatureID*> is the name of the Feature folder located in the 14\Template\Features folder. It must be a valid file path; for example, MyCustom.
- <*URLname*> is the Feature parent URL of the Web application, site collection, or Web site for which the Feature is being activated; for example, http://somesite.

For more information, see Enable-SPFeature (*http://technet.microsoft.com/library/9b68c192-b640-4cb8-8a92-a98008169b27(Office.14).aspx*).

Note:
We recommend that you use Windows PowerShell when performing command-line administrative tasks. The Stsadm command-line tool has been deprecated, but is included to support compatibility with previous product versions.

Deploy templates (SharePoint Server 2010)

This article describes how to create a custom site definition and deploy it by using a solution package.

In this article:

- What are site definitions?
- Site definitions and configurations
- Create a custom site definition and configuration
- Deploy a site definition by using a solution package

What are site definitions?

In Microsoft SharePoint Server 2010, a user creates a site through the user interface (UI) by selecting a site definition configuration or custom site template that defines how to instantiate the site. A site definition is a template that determines, for example, the lists, files, Web Parts, Features, or settings with which to provision a new SharePoint site.

A site definition is a family of site definition configurations. Each site definition specifies a name and contains a list of the site definition configurations. In SharePoint Server 2010, a site definition consists of a set of XML files that can be applied to provision new sites. The files are located on Web servers.

Site definitions consist primarily of multiple XML and ASPX files stored on a front-end Web server in folders under the %ProgramFiles%\Common Files\Microsoft Shared\web server extensions\14\TEMPLATE\SiteTemplates folder.

A site created from a site definition adds to, but does not repeat, the structural and content information from the original site definition. Throughout their lifecycle, sites continue to depend on the site definition that is their ultimate foundation. For this reason, Microsoft does not support changing or removing a site definition after sites have been created from it. Such changes may cause sites created from the definition to stop working properly or may prevent the creation of new sites based directly, or indirectly, on the site definition. To customize a site definition, developers can add a Feature that includes the changes to the site definition. The site definition itself is not modified.

For more information about what kinds of customizations of site definitions are supported by Microsoft, see Supported and unsupported scenarios for working with custom site definitions and custom area definitions in Windows SharePoint Services, in SharePoint Portal Server 2003, and in Office SharePoint Server 2007 (*http://go.microsoft.com/fwlink/?LinkId=187678*).

In the object model, an SPWebTemplate represents a site definition (and configuration). For more information about site templates and site definitions, see Site Templates and Definitions (*http://go.microsoft.com/fwlink/?LinkID=184756*).

Note:
The STP format of a custom site template (.stp file) is deprecated in SharePoint Server 2010 and replaced with WSP format site templates. In Office SharePoint Server 2007, users can save an existing site as a custom site template. The site template is stored in the database as a model, and users can select the site template as a foundation that defines how to instantiate the site. In SharePoint Server 2010, users can save an existing site as a template. The template is saved as a .wsp file in the Solution Gallery of the top-level site in a site collection, where it becomes available for subsite creation on all Web sites in the site collection..

For more information about site definitions, see Site Definitions and Configurations (*http://go.microsoft.com/fwlink/?LinkId=183458*).

Site definitions and configurations

A site definition defines a specific SharePoint site. There are five site definitions natively installed in SharePoint Server 2010. A site definition can include more than one site definition configuration. SharePoint Web sites are based on specific site definition configurations that include the following:

- STS includes the site definition configurations for Team Site, Blank Site, and Document Workspace.
- MPS includes the site definition configurations for Basic Meeting Workspace, Blank Meeting Workspace, Decision Meeting Workspace, Social Meeting Workspace, and Multipage Meeting Workspace.
- CENTRALADMIN provides a site definition configuration for Central Administration Web sites.
- WIKI provides a site definition configuration for Web sites that support community content by using wiki technology.
- BLOG provides a site definition configuration for blogs.

Each site definition consists of files that are placed in the \\Program Files\Common Files\Microsoft Shared\web server extensions\14\TEMPLATE\SiteTemplates subfolders of front-end Web servers during installation of SharePoint Server 2010. Site definition files include .xml, .aspx, .ascx, and .master page files, in addition to document template files — such as .dot and .htm — and content files, such as .gif and .doc.

Uncustomized pages and page customization

Site definition files are cached in memory on the server at process startup of Microsoft Internet Information Services (IIS). This allows uncustomized pages to be reused across sites. The information contained in these files is pulled from the cache at run time. Pages and list schemas are read from the site definition files but appear to be actual files within a site. New Web Part pages are also considered to be uncustomized.

When site pages are customized — excluding browser-based customizations such as modifications to Web Parts — their contents are stored in the content database, and the customized site page is used instead of the original page from the site definition. Uploaded .aspx files are automatically considered to be customized.

For more information about ghosting and page customization, see Site Definitions and Configurations (*http://go.microsoft.com/fwlink/?LinkId=183458*).

Core schema files

The following table lists the core XML files that can be modified for a site definition and shows their locations in the file system.

WebTemp.xml	Identifies the site definitions and provides information about their configurations. Located in: \TEMPLATE\1033\XML
Onet.xml	Defines the navigation areas, specifies the list definitions available, specifies document templates and their files, defines the base types for lists, and defines configurations and modules for site definitions. Located in: \TEMPLATE\SiteDefinitions*site_type*\XML
Schema.xml	Defines the views, forms, toolbar, and special fields in a list definition. Each definition has its own Schema.xml file. Located in: \TEMPLATE\FEATURES\List_Definition_Name
DocIcon.xml	Each front-end Web server in a SharePoint Server deployment contains a DocIcon.xml file that maps file programmatic identifiers (ProgIDs) and file name extensions of document types to specific icons and to controls for opening each type. Changes to DocIcon.xml are global to a SharePoint Server deployment and affect all site definitions on the front-end Web server. Located in: \TEMPLATE\XML

These XML files use Collaborative Application Markup Language (CAML) (*http://go.microsoft.com/fwlink/?LinkID=183464*) for defining aspects of a site. For more information about these core XML files that you can use to customize site definitions, see Site Definitions and Configurations (*http://go.microsoft.com/fwlink/?LinkID=183458*).

Create a custom site definition and configuration

You can create custom site definitions by manually copying an existing site definition or by importing a .wsp file into Visual Studio 2010.

Import items from an existing SharePoint site

This method requires saving a site as a template from SharePoint Server to generate a .wsp file, and then importing the .wsp file into Visual Studio 2010 by using the solution import project template. The Import SharePoint Solution Package project template lets you reuse elements such as content types, list definitions, and fields from existing SharePoint sites in a new Visual Studio SharePoint solution. For more information about importing items from an existing SharePoint site into a Visual Studio SharePoint project, see Importing Items from an Existing SharePoint Site (*http://go.microsoft.com/fwlink/?LinkID=187040*). This chapter includes a walkthrough that demonstrates the following tasks:

1. Customizing a SharePoint site by adding a custom site column.
2. Exporting a SharePoint site to a .wsp file.
3. Importing the .wsp file into Visual Studio SharePoint project by using the .wsp Import project.

Copy an existing SharePoint site

This method involves copying an existing site definition, modifying the copy, and changing two schema files: the copy of a WebTemp.xml file, and the copy of an Onet.xml file.

Warning:
Do not modify the originally installed WebTemp.xml file.

1. Copy an existing site definition folder located in the Local_Drive:\Program Files\Common Files\Microsoft Shared\web server extensions\14\TEMPLATE\SiteTemplates\ directory. Your copy should be a peer of the original, and you can give it any name that contains no spaces.

   For example, to create a custom site definition that derives from the team site definition for Microsoft SharePoint Server, copy the \sts folder.

2. Make a copy of the WebTemp.xml file. This file is located in Local_Drive:\Program Files\Common Files\Microsoft Shared\web server extensions\14\TEMPLATE\1033\XML.

   Give the file a unique name by appending a string to the name of the original file; for example, WebTempAction.xml. At run time, the compiler merges information contained in this file with the information contained in the original file to specify which site definition configurations are available for creating new sites.

3. Customize the contents of the new WebTemp file.

   Each WebTemp.xml file contains a collection of **Template** elements and **Configuration** subelements, which identify to the compiler all the site definition configurations that can be instantiated. The **Configuration** element defines, for example, a title, a description, the URL for the image displayed in the user interface (UI), and a display category that specifies the tab on which to display the template in the **Template Selection** section of the Create Site Collection page.

   Important:
   In each **Template** element defined in the WebTemp file, the **Name** attribute must contain the same name that is assigned to the new folder.To avoid conflict with IDs already used in SharePoint Server 2010, use unique values greater than 10,000 for the ID attribute.

The following example uses two **Configuration** elements in the WebTemp.xml file to define different site definition configurations for instantiating a site, one for a Research Collaboration site and the other for a Research Document Workspace site. This example uses only two configurations within a single site definition, but you can include multiple site definitions, each with multiple configurations, within a single WebTemp.xml file. Each site definition references a different site definition folder and its Onet.xml file.

```
<?xml version="1.0" encoding="utf-8" ?>

<Templates xmlns:ows="Microsoft SharePoint">

  <Template Name="RESEARCH" ID="10001">

    <Configuration ID="0" Title="Research Collaboration site"

      Hidden="FALSE" ImageUrl="_layouts/images/stsprev.jpg"

      Description="This definition creates a site for the Research

      team to create, organize, and share general information."

      DisplayCategory="Collaboration">

    </Configuration>

    <Configuration ID="1" Title="Research Workspace" Hidden="FALSE"

      ImageUrl="_layouts/images/dwsprev.jpg" Description="This

      definition creates a site for Research team colleagues to

      work together on specific documents."

      DisplayCategory="Collaboration">

    </Configuration>

  </Template>

</Templates>
```

As indicated by the value of the **Name** attribute in the **Template** element, this example assumes that a site definition directory named "RESEARCH" exists. If a WebTemp*.xml file specifies more than one site definition, the definitions are distinguished by their unique **ID** values.

Each **Configuration** element also contains an **ID** attribute. The combination of this **ID** and the value of the **Name** attribute in the **Template** element provides a reference to the contents of a specific **Configuration** element in a specific Onet.xml file. In the example, the **Name** attribute contains RESEARCH and the **ID** attributes contain 0 and 1, which reference the RESEARCH site definition and configurations with IDs of 0 or 1 in Onet.xml.

You may need to reset IIS to cause the new definition configuration to appear as an option in the UI. To do this, enter iisreset at a command prompt.

For more information about defining each site definition configuration in Onet.xml, see How to: Use Site Definition Configurations (*http://go.microsoft.com/fwlink/?LinkId=183465*).

Deploy a site definition by using a solution package

To deploy a custom site definition by using a solution package, add a **SiteDefinitionManifest** element to the manifest file of the solution package. Add the **TemplateFiles** element to define the template files that must be deployed in a subfolder of the \14\Template folder

Add a SiteDefinitionManifest element

The **SiteDefinitionManifest** element has a **Location** attribute that picks up all the files in the specified folder and creates the required folder in the \14\Template\SiteTemplates folder. The **WebTempFile** child element deploys the **webtemp*.xml** file to make the template known to SharePoint 2010 Products, as shown in the following example:

```
<SiteDefinitionManifests>
  <SiteDefinitionManifest Location="LitwareSiteTemplate">
    <WebTempFile Location="1033\xml\webtempLitware.xml" />
  </SiteDefinitionManifest>
</SiteDefinitionManifests>
```

Add a TemplateFile element

The **TemplateFile** element in a solution manifest file is used to define the template files that must be deployed in a subfolder of the \14\Template folder. An example of the kind of file you can deploy in this way is the fldtypes*.xml file, which defines the details of a custom field type. Use the **Location** attribute to specify the relative path to the file, which is indicated by the string "Text" in the following example:

```
<TemplateFiles
  <TemplateFile
    Location="Text"/>
    ...
</TemplateFiles>
```

For more information about how to deploy solutions, see Deploy solution packages (SharePoint Server 2010).

Workflow deployment process (SharePoint Server 2010)

After you prepare a workflow for use in Microsoft SharePoint Server 2010, deployment of the workflow varies depending on whether you use a predefined workflow, a Microsoft Office SharePoint Designer workflow, or a Microsoft Visual Studio custom workflow.

This article contains information and procedures about how to deploy workflows in SharePoint Server 2010.

In this article:

- Overview
- Before you begin
- Deploying workflows
- Verification

Overview

SharePoint Server 2010 provides a set of predefined workflows that can be used to guide and track common tasks such as document review or approval.

You can also use Office SharePoint Designer to define your own workflows, or you can use Visual Studio to create code-based custom workflows. Workflows are built on Windows Workflow Foundation. For more information, see Windows Workflow Foundation (*http://go.microsoft.com/fwlink/?LinkId=166185*). The workflows run in SharePoint sites and are integrated with and available across the Microsoft Office system, as follows:

- SharePoint Server 2010: Used to host workflows. After a workflow is deployed to the host, you can activate, configure, start, participate in, and track the workflow.
- SharePoint Designer 2010: Used to create user-defined workflows.
- Visual Studio: Used by developers to create workflows.
- Microsoft Visio 2010: Used together with SharePoint Designer 2010 to visualize and create user-defined workflows.
- Microsoft Word, Microsoft PowerPoint, Microsoft Excel, and InfoPath Forms Services: Used to start and participate in workflows.
- Microsoft Outlook: Used to receive workflow notifications and track tasks.

Important:
The last two items in this list that relate to integration with the Microsoft Office 2010 suites and InfoPath Forms Services are available only in SharePoint Server 2010. They are not available in SharePoint Foundation 2010.

Note:
To use Visio 2010 for workflow visualization, you must be running the Enterprise version of SharePoint Server 2010 and Visio 2010 Premium.

Before you begin

Before you perform the deployment procedures in this article, confirm that the server is running the Standard or Enterprise version of SharePoint Server 2010.

Deploying workflows

A number of predefined workflows have already been installed as SharePoint Features in SharePoint Server 2010.

Workflows in a SharePoint Web site are stored as workflow templates. As an alternative to using the predefined workflows, you can create SharePoint workflow templates in Office SharePoint Designer and Visual Studio, and then deploy them to a SharePoint Server 2010 Web site. Use SharePoint administration tools as needed to add the template to libraries or lists on a SharePoint Server 2010 Web site.

To deploy a predefined workflow, you activate it for the site, associate it with a list, library, content type, or site, and then start the workflow.

To deploy workflows in SharePoint Server 2010, use the appropriate procedure from the following:

- Deploy predefined workflows
- Deploy SharePoint Designer workflows
- Deploy Visual Studio workflows

Deploy predefined workflows

Activate the workflow

As described earlier in this article, before you can use a predefined workflow, it must be active in the site or site collection. Only active workflows can be associated with the lists and libraries on the site or site collection.

Many of the predefined workflows are active by default when a site or site collection is created. Because workflows can be deactivated, you can check the site or site collection to determine whether the predefined workflow that you want to use is active. Use the following procedure to determine whether the predefined workflow that you want to use is active, and then activate the predefined workflow as necessary.

Note:

In SharePoint Foundation 2010 there is only one predefined workflow, the Three-state workflow, and it is active by default when a site or site collection is created in SharePoint Foundation 2010.

To activate a workflow that is deployed as a feature, such as the predefined workflows, see Activate or deactivate a workflow (SharePoint Server 2010) (*http://technet.microsoft.com/library/3fc9047e-82be-4c81-bc12-7b5dd8b0386e(Office.14).aspx*).

Add the workflow association to a list, library, content type, or site

When you add a workflow, you associate the workflow with a list, library, content type, or site. You configure the workflow by specifying parameters such as the workflow name, start options, participants, and completion options.

To add a workflow association, see Add a workflow association (SharePoint Server 2010) (*http://technet.microsoft.com/library/4a0d02c0-5ecb-4912-a7f3-2f7db70df775(Office.14).aspx*).

Start the workflow

After you activate a workflow and add it to a list, library, content type, or site, an authenticated user can run the workflow on an item in the list, on a document in the library, or on a site in the case of a site workflow. When you add the workflow, you specify whether you want the workflow to run automatically or manually. If the workflow is configured to start automatically, the default settings are always used when the workflow begins. If the workflow is configured to start manually, a user can modify the default settings, such as specifying workflow participants and specifying a due date. The workflow runs on items in the list or documents in the library with which the workflow is associated.

The procedure for starting a workflow depends on whether it was configured to start manually or automatically and whether you start it from SharePoint Foundation 2010 or SharePoint Server 2010, or from the Office system.

Note:
To enable users to start workflows in the Office 2010 suites, you must be running SharePoint Server 2010. SharePoint Foundation 2010 does not enable users to start workflows in the Office 2010 suites.

For more information, see Start a workflow instance (SharePoint Server 2010) (*http://technet.microsoft.com/library/a629fa2e-04a7-42e1-942c-48059f255642(Office.14).aspx*).

Deploy SharePoint Designer workflows

When user-defined workflows are enabled, users can deploy Office SharePoint Designer workflows on their sites.

Enable user-defined workflows

To allow users to create and run SharePoint Designer 2010 workflows, you must ensure that user-defined workflows are enabled for the site collection. By default, this setting is enabled. When this setting is enabled, users can define workflows in a declarative workflow editor such as the SharePoint Designer 2010 workflow editor. A *declarative* workflow is a workflow that is built from conditions and actions that are assembled into rules and steps, and that sets the parameters for the workflow without writing code. Unlike code-centric workflows such as those that are created by using Visual Studio, declarative workflows are not deployed to SharePoint Server 2010 as compiled code. Instead they are compiled at runtime.

Because the capability to use declarative workflows on the Web application can be turned off, you can check the Web application to determine whether declarative workflows are active.

For information, see Enable or disable declarative workflows (SharePoint Server 2010) (*http://technet.microsoft.com/library/fe4b55a2-ce50-4ba9-9619-63924e58033f(Office.14).aspx*).

Create a SharePoint Designer workflow

By using the Workflow Designer wizard in SharePoint Designer 2010, you can create workflows that add application logic to the site or site collection without writing custom code. The Workflow Designer incorporates the tasks of creating the workflow, activating the workflow, and adding it to the list, library, or site. You do not have to perform any manual configuration tasks outside the designer to deploy the workflow. However, if you publish a workflow template to a SharePoint site collection, you can download that template as a WSP file and then deploy it to other site collections. For more information, see Deploy a workflow as a WSP file (SharePoint Server 2010) (*http://technet.microsoft.com/library/b88e395b-c9c1-4e38-811a-cf834b2c6b07(Office.14).aspx*).

Start the workflow

Because SharePoint Designer 2010 can automatically activate the workflow and add it to a list, library, or site, an authenticated user can then run the workflow on an item in the list, on a document in the library, or on a site in the case of a site workflow. When you create the workflow in SharePoint Designer 2010, you specify whether you want the workflow to run automatically or manually. If the workflow is configured to start automatically, the default settings are always used when the workflow begins. If the workflow is configured to start manually, a user can modify the default settings, such as specifying workflow participants and specifying a due date, as allowed by the workflow template. When started, the workflow runs on items in the list, on documents in the library, or on the site with which the workflow is associated.

The procedure for starting a workflow depends on whether it was configured to start manually or automatically and whether you start it from SharePoint Server 2010, or from the Office system.

Note:
To enable users to start workflows in the Microsoft Office 2010 suites, you must be running SharePoint Server 2010. SharePoint Foundation 2010 does not enable users to start workflows in the Office 2010 suites.

For more information, see Start a workflow instance (SharePoint Server 2010) (*http://technet.microsoft.com/library/a629fa2e-04a7-42e1-942c-48059f255642(Office.14).aspx*).

Deploy Visual Studio workflows

After a Visual Studio custom workflow is created and installed, the processing to deploy it resembles that of a predefined workflow.

Create a custom workflow

When a custom workflow is created by using Visual Studio, it is packaged as a SharePoint Feature. Feature packaging is a way of encapsulating SharePoint solutions and functionality for ease of deployment. After the development team has created a workflow and packaged it as a Feature, deploy the workflow using the **Install-SPFeature**Windows PowerShell command as described in the following section.

Install the custom workflow

You install Features in the \Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\Template\Features directory. Each Feature is in its own subdirectory. At the root of this folder, a Feature.xml file defines the contents of the Feature. Use the Windows PowerShell**Install-SPFeature** cmdlet to install the Feature. For details, see Install-SPFeature (*http://technet.microsoft.com/library/a1093d30-68a1-4c84-8454-967bda8d68b9(Office.14).aspx*).

Important:
To run Windows PowerShell, you must be a member of the Administrators group on the local computer. Also, Windows PowerShell must be enabled as a feature on the server on which you are installing a workflow.

Activate the workflow

Before you can use a Visual Studio workflow that was deployed as a feature, you must activate it for the site collection. Only active workflows can be associated with the lists, libraries, content types, and sites.

To activate a workflow that is deployed as a feature, see Activate or deactivate a workflow (SharePoint Server 2010) (*http://technet.microsoft.com/library/3fc9047e-82be-4c81-bc12-7b5dd8b0386e(Office.14).aspx*).

Add the workflow to a list, library, content type, or site

When you add a workflow, you associate the workflow with a list, library, content type, or site, and you configure the workflow by specifying parameters such as the workflow name, start options, participants, and completion options.

To add a workflow association, see Add a workflow association (SharePoint Server 2010) (*http://technet.microsoft.com/library/4a0d02c0-5ecb-4912-a7f3-2f7db70df775(Office.14).aspx*).

Start the workflow

After you activate a workflow and add it to a list, library, content type, or site, an authenticated user can run the workflow on an item in the list or a document in the library or in the case of a site workflow, on a site. When you add the workflow, you specify whether you want the workflow to run automatically or

manually. If the workflow is configured to start automatically, the default settings are always used when the workflow begins. If the workflow is configured to start manually, the user can modify the default settings, such as specifying workflow participants and specifying a due date. The workflow runs on items in the list or documents in the library with which the workflow is associated.

The procedure for starting a workflow depends on whether it was configured to start manually or automatically and whether you start it from SharePoint Foundation 2010 or SharePoint Server 2010, or from the Office system.

Note:
To enable users to start workflows in the Office 2010 suites, you must be running SharePoint Server 2010. SharePoint Foundation 2010 does not enable users to start workflows in the Office 2010 suites.

For more information about starting a workflow, see Start a workflow instance (SharePoint Server 2010) (*http://technet.microsoft.com/library/a629fa2e-04a7-42e1-942c-48059f255642(Office.14).aspx*).

Verification

After you deploy a workflow, we recommend that you test the workflow to ensure that it is functioning correctly, sending e-mail notifications to the correct workflow participants at the correct stages in the workflow, and delivering the expected results.

Deploy software updates for SharePoint Server 2010

Microsoft periodically releases software updates for Microsoft SharePoint Server 2010. It is important to understand what these updates are and how to deploy them to servers or server farms. This section describes the software update process for SharePoint Server.

In this section:

- Software updates overview (SharePoint Server 2010)

  This article provides an overview of the software update process for SharePoint Server.

- Prepare to deploy software updates (SharePoint Server 2010)

  This article helps you determine which approach to use to update the servers or server farms in your environment, and lists the steps that you must take before you can start to install the update.

- Install a software update (SharePoint Server 2010)

  This article contains instructions for installing a software update and upgrading your content to that level.

Software updates overview (SharePoint Server 2010)

This article provides an overview of deploying software updates on a Microsoft SharePoint Server 2010 farm.

In this article:

- Improvements and new features
- Intended audience and scope
- Software update process
- Software update strategy
- Software update deployment cycle

Improvements and new features

SharePoint Server 2010 introduces improvements and new features that facilitate a better end-to-end software update experience. Some of these features are as follows:

- There is support for backward compatibility between update versions on different servers, which enables you to install the update binary files and postpone update completion to a later time.
- You can update multiple Microsoft SharePoint Server servers concurrently to shift the workload to the database servers.
- There is full support for automatic updates that use Windows Server Update Services (WSUS), Windows Update, and Microsoft Update.

  Note:
  An automatic update will install the binary files on the farm servers, but you must complete the software update by running the upgrade on the servers.
- Administrators can monitor the status of the update by using the Central Administration Web site or Windows PowerShell.

For more information about SharePoint Server improvements and new features, see What's new in upgrade (SharePoint Server 2010) (*http://technet.microsoft.com/library/e5362d08-06a2-448f-9b4e-8c459d4583bf(Office.14).aspx*).

Intended audience and scope

The information that is provided about the software update process is intended for all IT professionals who maintain SharePoint Server 2010. However, the specific instructions for installing a software

update are intended for IT professionals who have to deploy software updates on a SharePoint Server server farm.

The information in this article applies to the following products:

- SharePoint Server 2010
- SharePoint Server 2010 language pack
- Microsoft Filter Pack

Note:
The process for installing software updates in stand-alone environments of SharePoint Server is a simpler process than the process for installing software updates in a server farm and does not require all the steps that are required for a server farm.

Software update process

It is important to understand that deploying updates in a SharePoint Server 2010 environment is a two-phase process: patching and upgrading. The term *patch* is used in this article to differentiate between updating the software and upgrading the software.

Each phase has specific steps and results. It is possible to postpone the upgrade phase.

Caution:
Inconsistent farm behavior may result from postponing the upgrade for more than several days. The longer the postponement, the larger the risk is that farm behavior issues will occur.

Update phase

The patch phase has two steps, the patching step and the deployment step. During the patching step, new binary files are copied to the Central Administration server. Any services that are using files that have to be replaced are temporarily stopped. Stopping services reduces the requirement to restart the server to replace files that are being used. However, there are some instances when you must restart the server.

The second step in the patch phase is the deployment step. In this step, the installer copies support files to the appropriate directories on the server that is running SharePoint Server. This step ensures that all the Web applications are running the correct binary files and will function correctly after the update is installed. The update phase is complete after the deployment step.

The next and final phase to deploy software updates is the upgrade phase.

Upgrade phase

After you finish the patch phase, you must complete the update installation by starting the upgrade phase. The upgrade phase is task intensive and, therefore, takes the most time to finish. The first action is to upgrade all the SharePoint Server processes that are running. After the processes are upgraded,

the databases are crawled and upgraded. Because the upgrade process can run on a single server, the other servers in the farm can continue to serve requests.

For more information about upgrades, see Upgrade process overview (SharePoint Server 2010) (*http://technet.microsoft.com/library/4d7a8038-4b27-4bd8-a855-585db4e924a8(Office.14).aspx*).

Software update strategy

The update strategy that you select will be based primarily on one of the following factors:

- The amount of downtime that is acceptable for installing the update.
- The additional staff and computing resources that are available to reduce downtime.

When you are determining your update strategy, consider how the strategy enables you to manage and control the update.

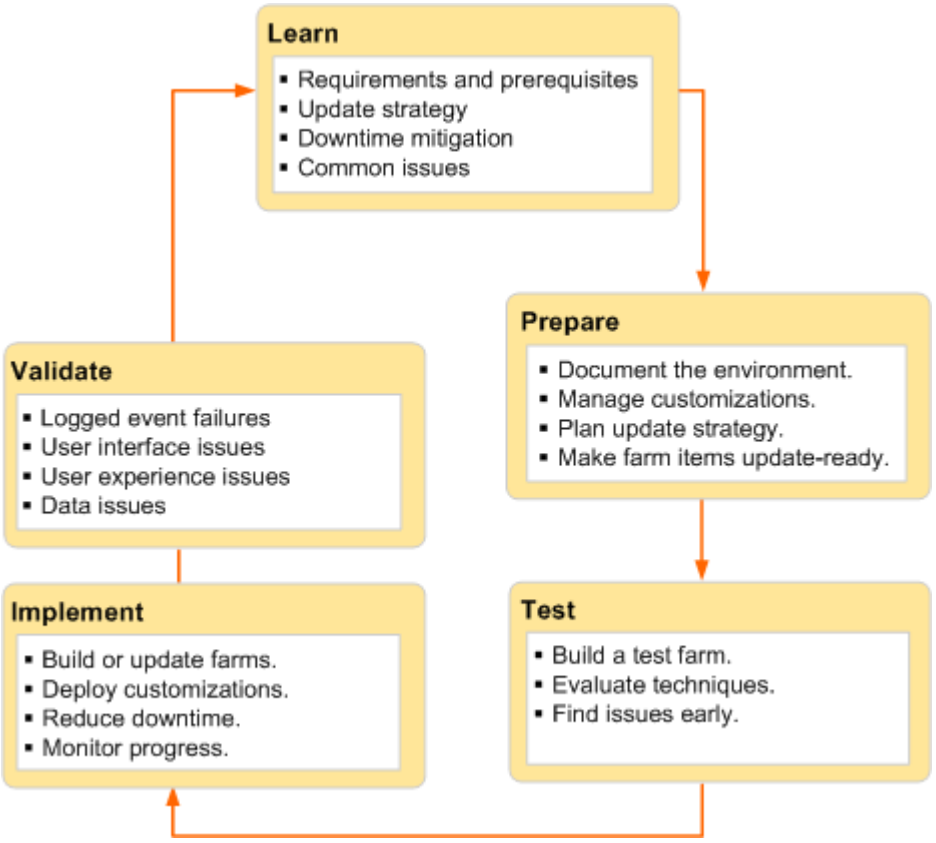
In terms of downtime reduction, the following options, ordered from most to least downtime, are available:

- Install the update and do not postpone the upgrade phase.
- Install the update and postpone the upgrade phase.
- Install the update with the shortest possible downtime and postpone the upgrade phase.

Software update deployment cycle

The cycle that is used for upgrading SharePoint Server farms and servers also applies to deploying software updates, which are a subset of an upgrade.

We recommend that you use the update cycle that is shown in the following illustration as a guide to deploy software updates.



Learn

During this phase of the cycle the purpose is to learn what is required to install the update. This information also affects new servers that you want to update and then add to the farm.

Requirements and prerequisites

First, ensure that the system can be provisioned as a farm server. For more information, see Hardware and software requirements (SharePoint Server 2010). Ensure that any server that you plan to update is running the same version of the operating system as the other farm servers. This includes updates, service packs, and security hotfixes.

Update strategy

Determine which strategy you want to use to update the farm. Depending on your requirements, you can use one of the following strategies:

- In-place
- Database attach

You can use either of the previous strategies to create a hybrid approach that is tailored to your environment. For more information, see Determine upgrade approach (SharePoint Server 2010) (*http://technet.microsoft.com/library/f11e6c4f-dc2a-4d17-a2c8-9455792b4b9b(Office.14).aspx*).

Downtime reduction

Research and assess the options that are available for reducing downtime. The first thing to check for is missing dependencies, which may extend the amount of downtime. Identify all the dependencies for the update and either address these dependencies before you start to deploy the update, or factor the time cost into your schedule. Consider using read-only content databases and doing parallel upgrades to reduce downtime.

Important:
We strongly advise against using alternate access mapping URL redirection (AAM) with database attach as an option for downtime reduction. AAM was not designed to deploy software updates. For more information, see Using AAM URL redirection as part of the upgrade process (SharePoint Server 2010) (white paper) (*http://technet.microsoft.com/library/a5cbf0f4-6eba-4982-9b6f-2c98cd379f0c(Office.14).aspx*).

Common issues

Identify and address common issues such as missing or out-of-date dependencies and lack of space on the servers where the update will be installed.

Prepare

Prepare for the software update by documenting the environment and planning an update strategy to ensure that the update will go as planned in the expected downtime window.

Document the environment

The purpose of documenting the environment is to determine what is unique in your farm. You can use several techniques to gather information about your farm, such as manual inspection, comparisons by using WinDiff, and Windows PowerShell commands.

Document, as appropriate, the following elements of the environment:

- Farm topology and site hierarchy
- Language packs and filter packs that are installed
- Customizations that could be affected by the update

Manage customizations

Customizations are typically one of the top issues during a farm upgrade or software update. Identify your farm customizations and determine whether they might be affected by the update. If in doubt, err on the side of caution and determine how you will manage the customizations. You must ensure that customizations will work after the software update. You can use the Stsadm command, **ExportIPFSAdminObjects**, to collect and export customizations.

For more information, see Determine how to handle customizations (SharePoint Server 2010) (*http://technet.microsoft.com/library/be1de0da-addb-4e6a-852c-0e68072d7f95(Office.14).aspx*).

Plan the update strategy

During the Learn phase of the update cycle, you should have determined an update strategy and the required downtime minimization. In addition to determining hardware, space, and software requirements, you must include the following in your update strategy:

- The update sequence for the farm servers
- The order of operations
- The downtime limits and how you plan to reduce downtime
- A rollback process if there is a major problem

Tip:
Clean up the farm environment before you deploy the update. The benefits of a cleanup are improved update installation performance and the elimination of potential issues during and after the software update. For more information, see Cleaning up your environment before upgrade (SharePoint Server 2010) (*http://technet.microsoft.com/library/bad0e5f7-44e1-40c7-8450-4c7069d4984b(Office.14).aspx*).

The two final requirements for the update strategy are a communication plan and an update schedule.

It is very important to communicate with site owners and users about what to expect during an upgrade. The administrator should inform them about downtime and the risk that the upgrade may take longer than expected or that some sites may need some rework after upgrade. For more information, see Create a communication plan (SharePoint Server 2010) (*http://technet.microsoft.com/library/dcf8e99b-fd8e-4a91-b281-bd673afee3b7(Office.14).aspx*).

Create a benchmark update operations schedule that contains the start times of operations related to the update deployment. At a minimum, the plan should include the following operations:

- Back up the farm.
- Start the update of the farm servers.
- Start the upgrade of the farm databases.
- Start a rollback of the environment, if it is required.
- Resume the upgrade, if it is required.
- Verify that the environment is completely working, either as the original version if you rolled back or the new version if you completed the upgrade.

Make farm items update-ready

Ensure that farm items are ready for the update. Farm items are ready if they are backed up, documented, or updated to ensure that the update can be installed. Verify that the following aspects of a farm are update-ready:

- Solutions
- Features
- Site definitions
- Web Parts

Test

The rigor, thoroughness, and detail of your tests determine the success or failure of the software update deployment. In a production computer environment there are no safe shortcuts, and there are consequences from insufficient testing. For more information, see Use a trial upgrade to find potential issues (SharePoint Server 2010) (*http://technet.microsoft.com/library/2b5d3825-adba-4185-84f2-ef59e8110fac(Office.14).aspx*).

Build a test farm

Build a test farm that is representative of the production environment. We recommend that you use a copy of the production data to determine potential problem areas and monitor overview system performance during the upgrade. The key indicator is the length of time it takes from the beginning to the end of the deployment process. This should include backup and validation. You can incorporate this information into the update schedule.

If possible, use hardware in the test environment that has equivalent performance capabilities to the production servers.

Tip:
Consider the use of a test farm in a virtual environment. After you finish the tests, you can shut down the virtual farm and use it later for future updates.

Evaluate techniques

A test farm also enables you to evaluate the techniques that you plan to use to update the production environment. In addition to testing and assessing your downtime reduction strategy, you can refine update monitoring. This is especially important in the areas of validating and troubleshooting the software update.

Implement

The update strategy that you use will determine whether you have to build a new farm or deploy the update on the current farm servers.

Build or update farms

Whether you build a new farm or do an in-place update, the most important farm elements to consider are as follows:

- Content
- Services
- Service applications

Deploy customizations

Use solutions whenever possible so that you can quickly deploy any customizations.

Reduce downtime

Reduce downtime by using techniques such as read-only databases and update parallelism. For more information, see Determine upgrade approach (SharePoint Server 2010) (*http://technet.microsoft.com/library/f11e6c4f-dc2a-4d17-a2c8-9455792b4b9b(Office.14).aspx*).

Monitor progress

The refined techniques that you use to monitor the software update in the test environment carry over to deploying the update in the production environment. Use the Upgrade and Migration page in Central Administration to monitor the status indicators that are available. This feature enables live monitoring and provides a single location to view the patch status for all farm servers. Additionally, you can use the Upgrade and Migration page to view the update status for individual servers and the status and type of farm databases. Finally, a valuable aspect of monitoring by using Central Administration is identifying farm servers that must be updated.

The following tables describe the status information that is available in Central Administration.

Status value	Description	Hyperlink
No action required	Farm server does not currently require any action to be taken by the administrator.	No hyperlink
Installation required	Farm server is missing an .msi file that is set to mandatory for all farm servers, or has a patch level below the individual farm-wide effective patch version.	Hyperlink to the Patch Deployment State page
Upgrade in progress	Farm server is currently undergoing an upgrade operation.	Hyperlink to the Upgrade Status page
Upgrade available	Farm server is running in backward-compatibility mode.	Hyperlink to the Upgrade and Migration page

Status value	Description	Hyperlink
Upgrade required	Farm server is outside the backward-compatibility mode range with one or more databases.	Hyperlink to the Upgrade and Migration page
Upgrade blocked	If an upgrade is available and any farm server requires installation, the remaining servers that do not require installation will be set to this status unless they are currently undergoing an upgrade.	Hyperlink to the Patch Deployment State page

Status value	Description
Installed	Indicates that no action is required
Missing/Required	Displayed if a product is required on each server or if a patch for a specific .msi file is located on one server but not on the server for which this status is shown
Missing/Optional	Displayed if a product is not required on each server
Superseded	Displayed if an update is no longer required on a server because a newer patch supersedes it

Other tools to monitor the update process are log files and Windows PowerShell commands.

Important:
Remember to monitor the length of time that the update is taking. Compare current update processes against the benchmark schedule to determine whether the update will meet the downtime window. If not, you should communicate this information to the farm users.

Validate

You can start to validate the success of the update during the implementation phase and continue validation after the update is implemented.

Logged event failures

Review the event logs to discover any issues that occurred during the deployment. Resolve these issues and then resume or restart the update as appropriate.

User interface or experience issues

Any user interface or user experience issues will surface on site pages.

Look for the following issues:

- Ghosting
- User interface version mismatch
- HTML and XHTML compliance

Additional issues may include missing templates, user identifiers, and content issues such as large lists.

Data issues

Data issues result from the condition of the farm databases and can include all or some of the following:

- Connectivity issues to data sources
- Database corruption
- Orphaned items
- Hidden column data

In some cases there may be minor issues that you can troubleshoot and then resume or restart the update. Be prepared to roll back the update as soon as there are issues that cannot be easily resolved.

Prepare to deploy software updates (SharePoint Server 2010)

This article describes the required and recommended tasks that have to be completed before you install software updates on servers in a Microsoft SharePoint Server 2010 farm.

In this article:

- Verify account permissions and security settings
- Determine the update approach
- Back up the environment
- Document the environment
- Determine whether related items need to be updated
- Obtain the software update and prepare the installation source (optional)

Verify account permissions and security settings

Verify that you have the required account permissions and know which security settings are in place on the farm. For more information, see Account permissions and security settings (SharePoint Server 2010) (*http://technet.microsoft.com/library/55b99d80-3fa7-49f0-bdf4-adb5aa959019(Office.14).aspx*).

Determine the update approach

There are two basic options for deploying a software update on a farm: in-place and database attach. Additionally, these options can be combined to use one of the hybrid approaches that are described in Determine upgrade approach (SharePoint Server 2010) (*http://technet.microsoft.com/library/f11e6c4f-dc2a-4d17-a2c8-9455792b4b9b(Office.14).aspx*).

Note:
Because installing a software update is a subset of a software upgrade, documentation about software upgrades applies to deploying software updates.

The differences between the two update approaches are as follows:

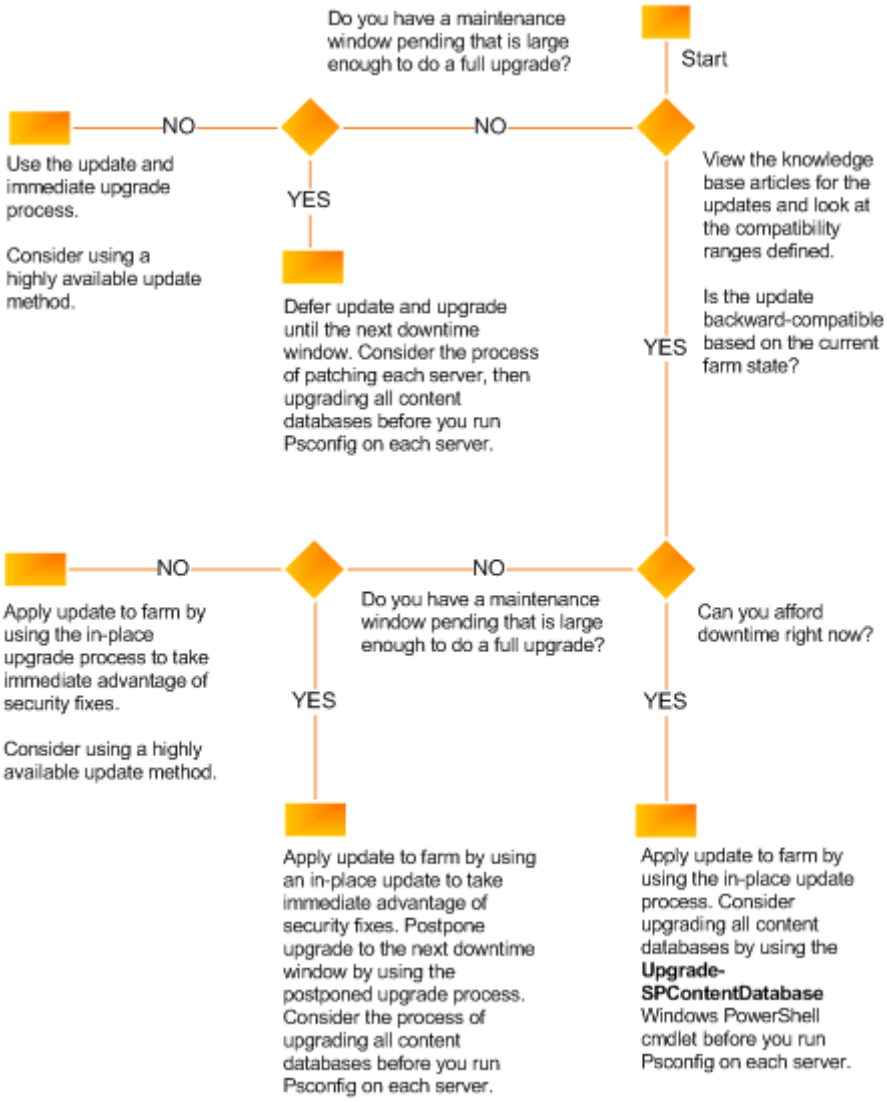
- In-place update: This approach is the easiest to do. With this method, the amount of downtime required is directly related to the size and complexity of the farm. You have two choices for an in-place update:
  - In-place without backward compatibility – The update is installed on all the farm servers at the same time and the content is upgraded without using backward compatibility. No downtime reduction is available for this method.

- In-place with backward compatibility to reduce downtime – The update is installed in stages and uses postponed upgrade with backward compatibility to reduce downtime.

- Database attach: This approach is more complex than an in-place update, and it costs more in person time and hardware resources. This update method uses two farms to provide high availability for existing content.

When you use either the in-place with backward compatibility method or the database attach method, you can use a postponed upgrade so that you can choose to upgrade the content first and then the farm and servers afterward.

We recommend that you use the following flowchart, which presents the key decision points and subsequent actions, for determining which update approach to use.



Whichever method you choose to use for updating your servers, you can use either the SharePoint Products Configuration Wizard or Windows PowerShell cmdlets to upgrade your content.

Back up the environment

To ensure that you can recover the existing environment in case something goes wrong during the update deployment process, we recommend that you back up the SharePoint Server 2010 environment before you start to install the update. A failed software update can be caused by factors other than the update process, such as the following:

- Media failure
- User errors (such as deleting a file by mistake)
- Hardware failures (such as a damaged hard disk or permanent loss of a server)
- Power failures
- Natural disaster

You can back up all or part of a farm. The following list summarizes the farm components that can be backed up individually:

- Configuration settings
- Web applications
- Service applications
- Search
- Secure store service
- Site collections
- Logs

For more information about how to determine what you need to back up and which method to use to do so, see Plan for backup and recovery (SharePoint Server 2010) (*http://technet.microsoft.com/library/01abe8d2-33f8-48fe-af76-40522a5afe08(Office.14).aspx*). After you determine which farm elements you will back up, refer to the articles listed in Backup (SharePoint Server 2010) (*http://technet.microsoft.com/library/79d47308-a90a-4c51-a1ae-93567e978236(Office.14).aspx*). These articles provide detailed instructions and guidance for backing up all or part of a farm.

Important:
Test the farm backups before you start to deploy the software update. You have to be sure that these backups are valid so that you can recover if there is a hardware failure or data corruption during the update process.

Document the environment

Be sure to document the farm, including any custom components in the farm, in case you need to rebuild.

For more information about creating an inventory of customizations, see Identify and install customizations (*http://technet.microsoft.com/library/2b5d3825-adba-4185-84f2-ef59e8110fac.aspx#Customizations*) in the Upgrade guide. In addition, document unique things about your farm, such as the following:

- Any large lists
- Any sites with large access control lists (ACLs)
- Any sites that are critical to your organization

Having a list of these items will help you more quickly validate your environment after you apply an update.

Determine whether related items need to be updated

Consider whether the following related items need to be updated when you update your farm:

- Filter packs
- Office Web applications
- Language packs

All these items are updated separately from SharePoint Server 2010. Check to see if any updates to these items are available, and evaluate whether you want to apply the updates to your farm when you apply the updates for SharePoint Server 2010. Language packs are usually only updated when service packs (SP1, SP2, and so on) are released.

Obtain the software update and prepare the installation source (optional)

If the servers on which you want to install SharePoint Server 2010 are isolated from the Internet, it is usually necessary to install software updates from an offline location. Even if the servers are not isolated, if you install software updates from an offline central location, you can ensure farm server consistency by installing a well-known and controlled set of images. Use the following procedure to prepare a software update for installation on a farm server.

You do not need to perform this procedure if you are downloading and installing the update directly to your servers.

To prepare an installation source

1. Download the software update that you want to install.

2. Extract the software update to a shared location by using the following command:

   <package> **/extract:<**path>

   The **/extract** switch prompts you to provide a folder name for the files. An example of a folder name for x64 systems is as follows:

   sps-kb999999-x64-fullfile-en-us.exe /extract:<\\computername\updateshare\Updates>

3. Copy the extracted files from the shared location to an Updates folder that you create on the computer where you want to start to install the update.

   Note:
   You must use the name Updates for this updates folder. If you use the **SupdateLocation="path-list"** property to specify a different location, Setup stops responding.

You can now use this location as an installation point, or you can create an image of this source that you can burn to physical media or save as an ISO file.

Slipstream package

In server farm deployments, all the Web servers must have the same software update version applied. This means that, before you add a new Web server to an existing server farm, the new Web server must have the same software updates as the rest of the Web servers in your server farm. Likewise, when you create a farm, all servers in the farm must have the same software updates. To make sure that all new servers have the same software updates applied, we recommend that you create an installation source that contains a copy of the release version of the software, together with software updates that match those installed on your server farm (also known as a slipstreamed installation source). When you run Setup from this updated installation source, the new Web server will have the same software update version as the rest of the Web servers in your server farm. For more information, see Create an installation source that includes software updates (Office SharePoint Server 2007) [*http://technet.microsoft.com/en-us/library/cc261890(office.12).aspx*].

Install a software update (SharePoint Server 2010)

This article describes how to install a software update on servers in a Microsoft SharePoint Server 2010 farm. Additionally, three example scenarios are discussed and an update procedure is provided for each scenario.

In this article:

- Verify the update strategy
- Monitor installation progress
- Handle update failures
- Review update scenarios
- Use the in-place method without backward compatibility
- Use the in-place method with backward compatibility
- Use the database attach method for high availability of existing content
- Verify update completion and success

Verify the update strategy

Before you start to deploy the software update, verify that the update strategy that you plan to use is optimal for your Microsoft SharePoint Server environment. There are several factors, such as downtime reduction, cost, and complexity that determine which strategy to use to deploy a software update. Use the flowchart in the "Determine Update Strategy" section of Prepare to deploy software updates (SharePoint Server 2010) to verify the update strategy that you want to use: in-place, database attach, or a hybrid.

Monitor installation progress

Monitor the update deployment process during the update to verify that the update is proceeding as planned. There may be issues that will block the update or that will result in an updated farm that has elements that do not work as expected. Pay extra attention to database synchronization and customizations.

We recommend that you use the Upgrade and Migration view in Central Administration as the primary tool for viewing product and patch installation status, data status, and upgrade status in real time.

After Setup runs, you can also view the log files and use Windows PowerShell to obtain the current results of the installation progress.

Handle update failures

SharePoint Server 2010 provides an improved approach to handling upgrade failures after the patching phase finishes. If an update fails and you are running in backward compatibility mode, you can restore the SharePoint Server database and continue to run in backward compatibility mode. After the update issue is resolved for the site, you can resume the upgrade. Any tasks that were completed are not run again. For more information, see Testing and troubleshooting upgrade (SharePoint Server 2010) (*http://technet.microsoft.com/library/ca15deea-3645-4a49-b39d-395ff87195fa(Office.14).aspx*).

If an update failed in earlier SharePoint Products and Technologies environments, you usually had to uninstall the product, install the older version, and then restore from a backup.

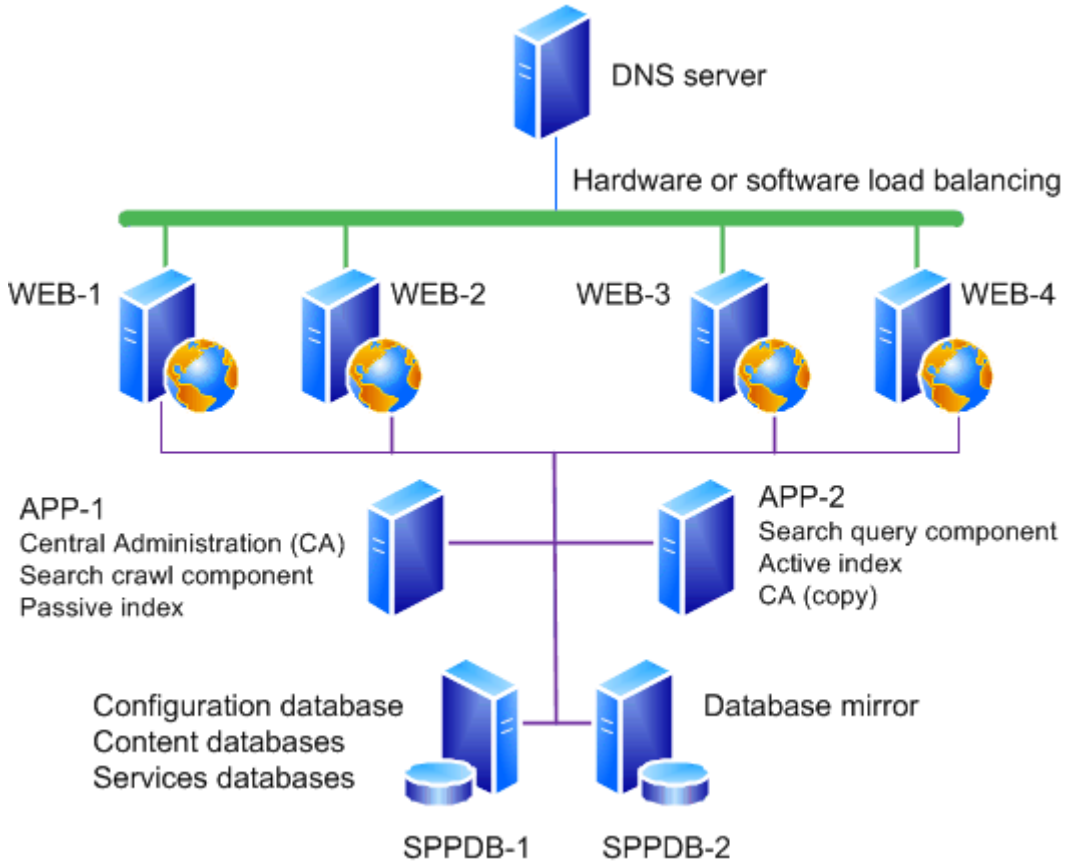
Review update scenarios

The following software update scenarios are discussed in this article:

- In-place without backward compatibility – The update is installed on all the farm servers at the same time and the content is upgraded without using backward compatibility.
- In-place with backward compatibility to reduce downtime – The update is installed in stages and uses deferred upgrade with backward compatibility to reduce downtime.
- Database attach for high content availability – This update uses two farms to provide high availability for existing content.

For more information about how the in-place and database attach processes work, see the diagrams in the following article: Upgrade process overview (SharePoint Server 2010) (*http://technet.microsoft.com/library/4d7a8038-4b27-4bd8-a855-585db4e924a8(Office.14).aspx*). Note that these articles are about how to upgrade across software versions, not how to install software updates. However, the general process is very similar.

The following illustration shows the farm topology that is used as an example for each patching scenario that is described in this article.



Initial state and required conditions

The preceding illustration shows the initial state of the farm before you install the update. Verify that the following conditions are true:

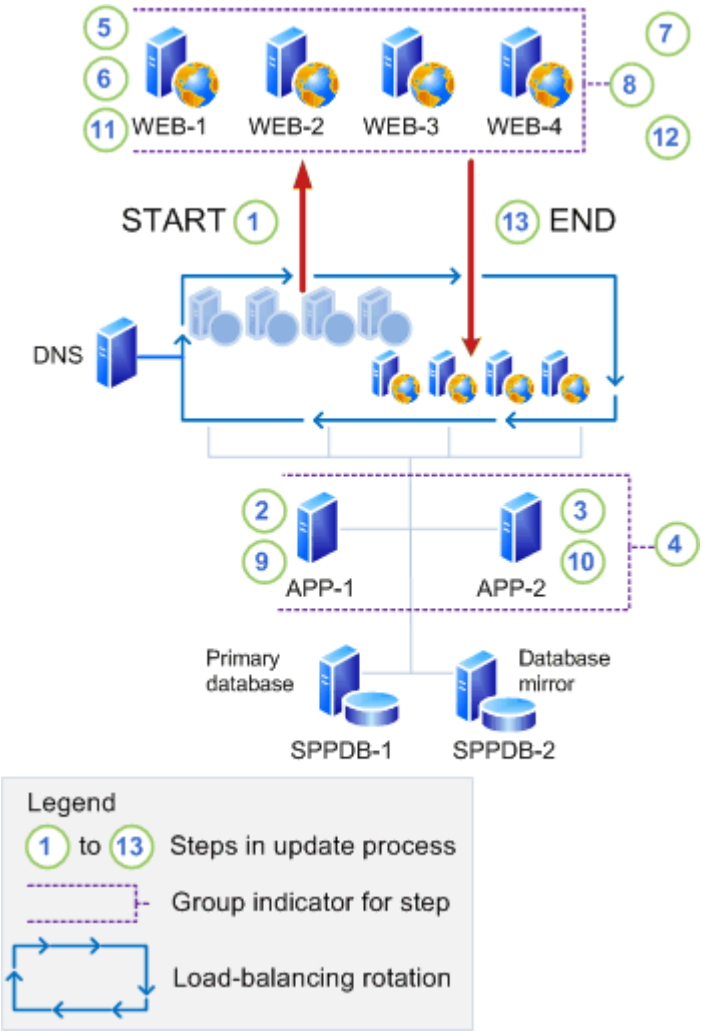
- All the front-end Web servers are load balanced together and are in rotation with the load balancer.
- All the farm servers are operating correctly.
- All the databases are active and operating correctly.

Do not start the software if any of the preceding conditions are not true. Resolve all issues before you continue.

Use the in-place method without backward compatibility

In this scenario the complete farm is shut down by disabling incoming requests to the front-end Web servers and then installing the update on all the farm servers. This strategy combines the update and the upgrade phase described in the "Software Update Process" section in Software updates overview (SharePoint Server 2010).

The following illustration shows the sequence of steps to follow to install the update on the farm.



Use the preceding illustration as a guide for using the recommended steps in the following procedure.

To install an update without backward compatibility

1. Remove the Web servers (WEB-1 to WEB-4) from rotation in the load balancer, or pause the load balancer to stop incoming requests to the servers.
2. Run the executable file to install the update on the application server that hosts Central Administration (APP-1).
3. Run the executable file to install the update on the application server that hosts the search query component (APP-2).
4. Verify that all the servers were updated successfully.
5. Log on to the first Web server (WEB-1).
6. Run the executable file to install the update on the Web server.
7. Run the executable file to install the update on the remaining Web servers (WEB-2, WEB-3, and WEB-4).
8. Verify that all the servers were updated successfully.
9. Run the SharePoint Products Configuration Wizard on the Central Administration server (APP-1) to upgrade the configuration database and upgrade each content database serially.
10. Run the SharePoint Products Configuration Wizard on the application server that hosts the search query component (APP-2).
11. Run the SharePoint Products Configuration Wizard on the first Web server (WEB-1).

    Note:
    Run the configuration wizard to ensure that if the update fails for a specific server, the error is not propagated to the other Web servers. For example, a failed upgrade for one server could make the upgrade fail for one or more site collections.
12. Repeat the preceding step for each remaining Web server.
13. Add the Web servers (WEB-1 to WEB-4) to the rotation in the load balancer, or start the load balancer to enable incoming requests to the servers.
14. Verify update completion and success. For more information, see Verify update completion and success.

Use the in-place method with backward compatibility

This scenario takes advantage of the backward compatibility of SharePoint Server 2010 and the deferred upgrade feature to reduce the downtime that is required to deploy a software update. This software update scenario uses two phases to install the update on farm servers. These phases are as follows:

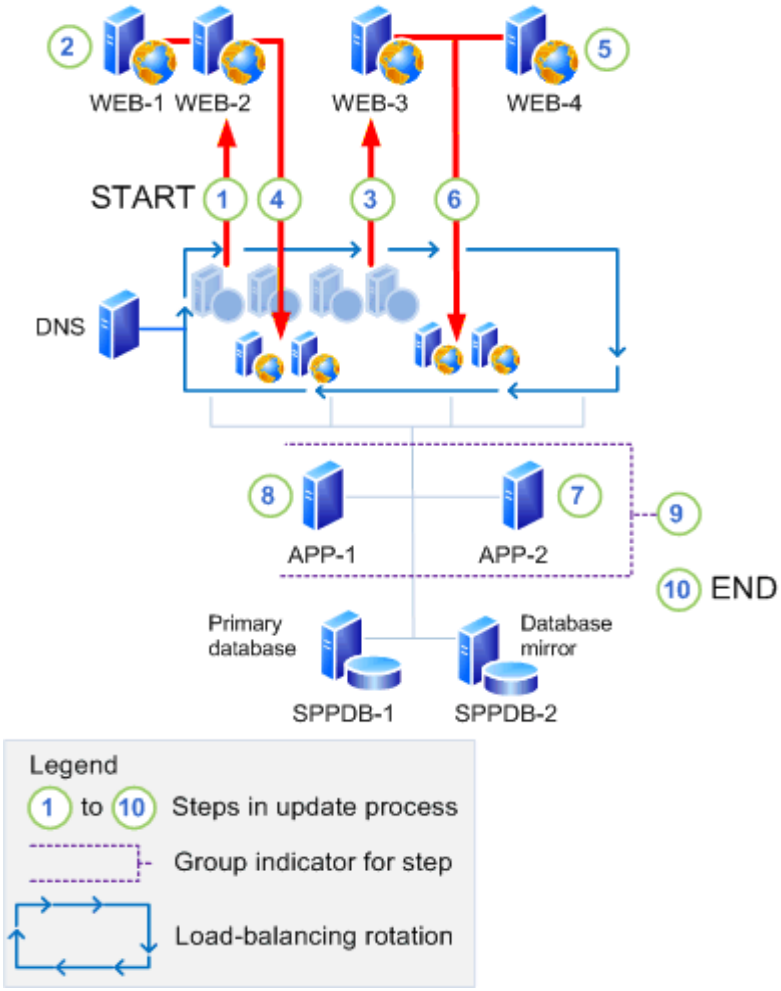
- Update to install the update on the farm servers.

- Upgrade to complete the patching process.

For more information about the software update process, see "The Software Update Process" section in Software updates overview (SharePoint Server 2010).

Update phase

The following illustration shows the sequence of steps that are required to install the update on the farm.



Use the preceding illustration as a guide for using the recommended steps in the following procedure.

To install the update on farm servers

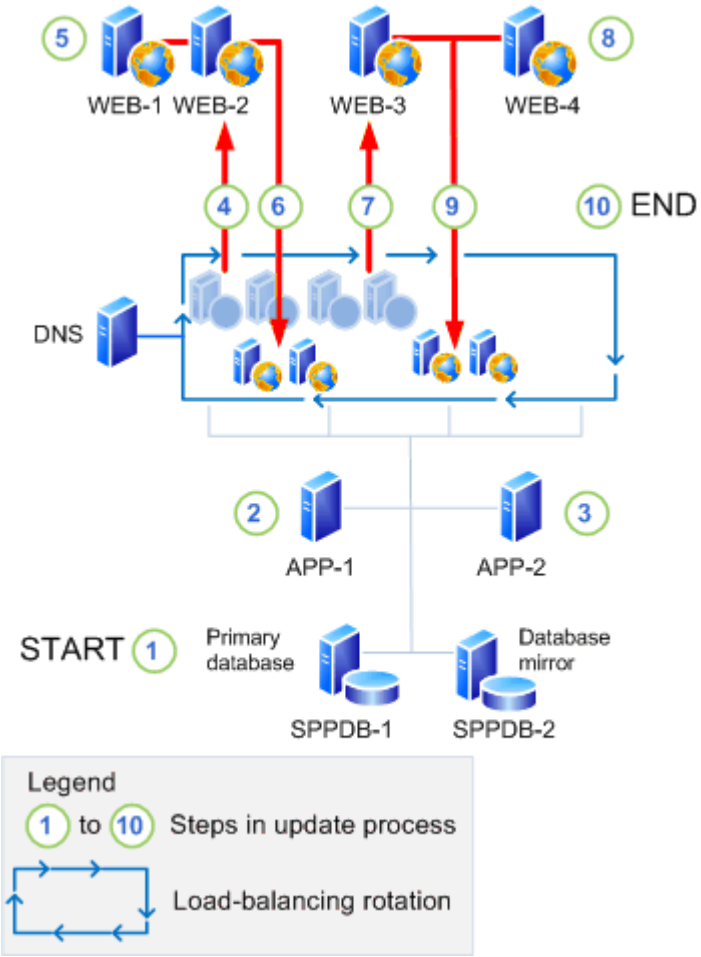
1. Remove half of the Web servers (WEB-1 and WEB-2) from rotation in the load balancer, or pause the load balancer to stop incoming requests to the servers.

2. Run the executable file to install the update on each Web server that is out of the load-balancing rotation (WEB-1 and WEB-2). Do not run the SharePoint Products Configuration Wizard on either of these servers. Verify that both of the Web servers were updated successfully.
3. Remove the remaining Web servers (WEB-3 and WEB-4) from rotation in the load balancer, or pause the load balancer to stop incoming requests to the servers. At this point none of the front-end Web servers are receiving requests for the farm.
4. Add the updated Web servers (WEB-1 and WEB-2) back into the load-balancing rotation.
5. Run the executable file to install the update on each Web server that is still out of the load-balancing rotation. Do not run the SharePoint Products Configuration Wizard on either of these servers. Verify that both of the Web servers were updated successfully.
6. Add the updated Web servers (WEB-3 and WEB-4) back into the load-balancing rotation.
7. Run the executable file to install the update on the application server that hosts the search query component (APP-2). Do not run the SharePoint Products Configuration Wizard on this server.
8. Run the executable file to install the update on the Central Administration server (APP-1). Do not run the SharePoint Products Configuration Wizard on this server.
9. Verify that both of the application servers (APP-1 and APP-2) were updated successfully.
10. Verify update completion and success. For more information, see Verify update completion and success.

At this point in the process, the databases and other components such as settings, features, and site-level data must still be upgraded because the SharePoint Products Configuration Wizard was not run on any of the farm servers.

Upgrade phase

The following illustration shows the sequence of steps that are required to finish the patching process by upgrading the farm servers.



Use the preceding illustration as a guide for using the recommended steps in the following procedure.

Important:
Monitor the status of the upgrade on each server before you upgrade the next server in the sequence.

To upgrade the farm servers

1. Use the Windows PowerShell **Upgrade-SPContentDatabase** cmdlet to upgrade each content database.

   You must run this cmdlet for each database. You can run it from any of the upgraded Web

servers or application servers. Note that the content for each database will be unavailable while this process is running on that database.

Note:
Some updates might also require you to run additional Windows PowerShell cmdlets to upgrade specific service applications.

2. Run the SharePoint Products Configuration Wizard on the Central Administration server (APP-1).

Note:
The SharePoint Products Configuration Wizard also starts an immediate upgrade of the configuration database and any other databases that are not already upgraded. Because the content databases are the only databases that are already upgraded, all the service application databases are also upgraded in this step.

3. Run the SharePoint Products Configuration Wizard on the remaining application server (APP-2).
4. Remove half of the Web servers (WEB-1 and WEB-2) from rotation in the load balancer, or pause the load balancer to stop incoming requests to the servers.
5. Run the SharePoint Products Configuration Wizard on the Web servers that are no longer in the load-balancing rotation (WEB-1 and WEB-2).
6. Add the upgraded Web servers (WEB-1 and WEB-2) back into rotation in the load balancer.
7. Remove the Web servers that have not been upgraded (WEB-3 and WEB-4) from rotation in the load balancer, or pause the load balancer to stop incoming requests to the servers.
8. Run the SharePoint Products Configuration Wizard on the Web servers that are no longer in the load-balancing rotation (WEB-3 and WEB-4).
9. Add the upgraded Web servers (WEB-3 and WEB-4) back into rotation in the load balancer.
10. Verify update completion and success. For more information, see Verify update completion and success.

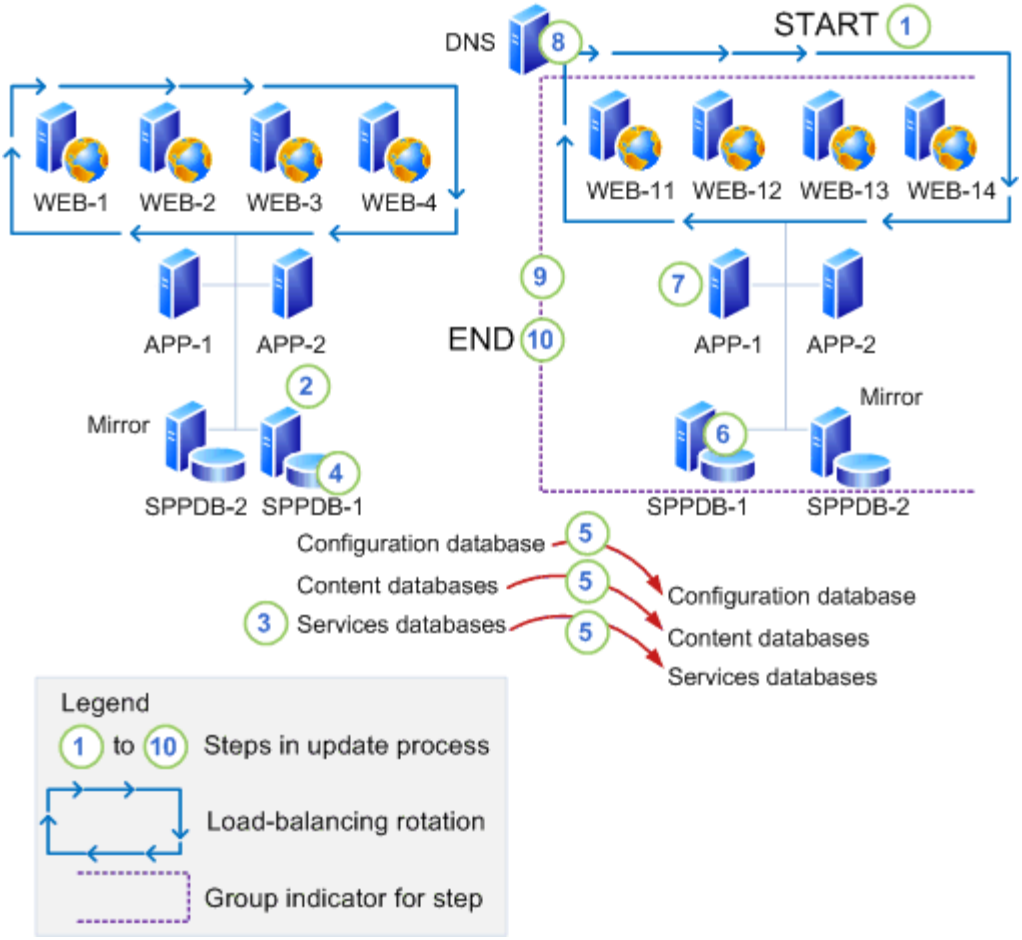
Note:

Steps 4-8 can also be performed on each server sequentially, instead of doing half of the servers and then the other half. The goal with upgrading half of the servers at a time is to maintain access during the upgrade process.

Use the database attach method for high availability of existing content

To ensure high availability for existing content, this scenario uses read-only databases on the existing farm. The update is installed on a new farm and user traffic is rerouted to this farm.

The following illustration shows the sequence of steps to follow to install the update on a new farm by using the database attach method. For more information, see Attach databases and upgrade to SharePoint Server 2010 (*http://technet.microsoft.com/library/d0a50e40-5322-4f7a-a3a4-c4660c879666(Office.14).aspx*).



Use the preceding illustration as a guide for using the recommended steps in the following procedure.

To install the update by using database attach

1. Create a new farm where you will install the software update. This farm does not require front-end Web servers. For more information, see Prepare the new SharePoint Server 2010 environment for a database attach upgrade (*http://technet.microsoft.com/library/99684f0b-f617-4c10-8420-d4d0adea3687(Office.14).aspx*).

   Note:
   If the original farm uses a database mirror, you must configure mirroring after you finish deploying the software update on the new farm.

2. Configure the databases on the existing farm so that they are in a read-only state.

   Note:
   If the existing farm is mirrored, you must pause mirroring before setting the databases to read-only.

   For more information about how to configure read-only databases, see the "Set the Previous Version Databases to Be Read-Only (Database Attach with Read-Only Databases)" section in Attach databases and upgrade to SharePoint Server 2010 (*http://technet.microsoft.com/library/d0a50e40-5322-4f7a-a3a4-c4660c879666(Office.14).aspx*) and Run a farm that uses read-only databases (SharePoint Server 2010) (*http://technet.microsoft.com/library/8b91dc0a-c37d-4ec8-aa75-deb3f268fb97(Office.14).aspx*).
3. Configure the service application databases on the existing farm so that they are in a read-only state. This prevents unexpected changes to service applications.
4. If you are patching the User Profile Service service application database, then you must export the User Profile Synchronization Service encryption key from the old database and then import the key into the new database. This key is also known as the Microsoft Identity Integration Server (MIIS) key, the Synchronization Service encryption key, and the Forefront Identity Manager 2010 (FIM 2010) key. If you do not export and then import the key correctly, the Synchronization Service will not start. To export the encryption key, complete these steps:
   a. Use farm administrator credentials to log on to the computer that contains the old User Profile Service service application database.
   b. Open the Command Prompt window, and then change to the following folder:

      %Program Files%\Microsoft Office Servers\14.0\Synchronization Service\Bin\
   c. Type the following command, and then press Enter:

      miiskmu.exe /e <*Path*>

      Where <*Path*> is the full path of the file to which you want to export the key, and the user name and password are the credentials for the farm administrator.
5. Back up the content databases on the existing farm. For more information, see Backup and recovery (SharePoint Server 2010) (*http://technet.microsoft.com/library/71abd06e-6730-442e-b2c1-e3ba9c04d497(Office.14).aspx*).
6. To import the encryption key, complete these steps:
   a. Use farm administrator credentials to log on to the computer that contains the new User Profile Service service application database.
   b. Attempt to start the User Profile Synchronization service. Because you have not yet imported the encryption key, the service will not start. Confirm that the service did not start by using the ULS log or by making sure that the status of the service is **Stopped**.
   c. Open the Command Prompt window, and then change to the following folder:

      %Program Files%\Microsoft Office Servers\14.0\Synchronization Service\Bin\
   d. Type the following command, and then press Enter:

miiskmu.exe /i <*Path*> **{0E19E162-827E-4077-82D4-E6ABD531636E}**

Where <*Path*> is the full path of the file to which you exported the key.

e. (Optional) To check that the encryption key was imported correctly, at the command prompt, type the following command, and then press Enter:

miiskmu.exe /c {0E19E162-827E-4077-82D4-E6ABD531636E}

7. Restore the content databases to the new database server.
8. Create service applications on the new farm for each existing service application in the old farm.

   You must duplicate all the settings from your existing farm.
9. Use database attach to create the databases on the new farm. For more information, see Perform a database attach upgrade to SharePoint Server 2010 (*http://technet.microsoft.com/library/b6580f87-40b4-4768-b589-6ba54013f7e4(Office.14).aspx*) and Attach and restore a read-only content database (SharePoint Server 2010) (*http://technet.microsoft.com/library/5417b04a-c7d9-4e9a-86fb-ee1d1c63508b(Office.14).aspx*).
10. Verify that there are no issues with the new farm.
11. Enable the new farm as the production farm by configuring DNS to point to the new farm or by making sure that the new farm is load balanced. Verify that users can access the new farm.
12. Allow time for users to switch from cached DNS, and then decommission the old farm.
13. Verify update completion and success. For more information, see Verify update completion and success.

Verify update completion and success

Regardless of the update strategy that you use and the monitoring that you do during the software update, you must verify update completion and success. For more information, see Verify upgrade and review upgraded sites (SharePoint Server 2010) (*http://technet.microsoft.com/library/f3819495-4b67-4a71-ba46-23e43b175620(Office.14).aspx*).

Deploy Office Web Apps (Installed on SharePoint 2010 Products)

This article discusses how to deploy Microsoft Office Web Apps. Office Web Apps is the online companion to Office Word, Excel, PowerPoint and OneNote applications that enables users to access documents from anywhere. Users can view, share, and work on documents with other users online across personal computers, mobile telephones, and the Web. Office Web Apps is available to users through Windows Live and to business customers with Microsoft Office 2010 volume licensing and document management solutions that are based on Microsoft SharePoint 2010 Products.

Note:

An appropriate device, Internet connection, and supported Internet Explorer, Firefox, or Safari browser are required. Some mobile functionality requires Office Mobile 2010, which is not included in Office 2010 applications, suites, or Office Web Apps. There are some differences between the features of Office Web Apps, Office Mobile 2010 and the Office 2010 applications.

Information provided in this article is intended for IT Pros planning to deploy Office Web Apps on SharePoint 2010 Products on-premises in their organizations. SharePoint 2010 Products in this article refers to Microsoft SharePoint Server 2010 and Microsoft SharePoint Foundation 2010 unless otherwise noted. This article does not provide guidance or instructions about how to deploy SharePoint 2010 Products. New and updated content will be published on a regular basis.

It is recommended you read Understanding Office Web Apps (Installed on SharePoint 2010 Products) (*http://technet.microsoft.com/library/8480064e-14a4-4b46-ad6b-0c836b192af2(Office.14).aspx*) and Planning Office Web Apps (Installed on SharePoint 2010 Products) (*http://technet.microsoft.com/library/3bd0a617-5f12-4a7e-bb75-b15c86c7e504(Office.14).aspx*) available on Microsoft TechNet (*http://go.microsoft.com/fwlink/?LinkID=78003&clcid=0x409*) before deploying Office Web Apps in your organization. ffice Web Apps can be downloaded from the Microsoft Volume Licensing Service Center (*http://go.microsoft.com/fwlink/?LinkID=191841&clcid=0x409*) on the Office Professional Plus 2010 32Bit or Office Standard 2010 32Bit site.

In this article:

- Understanding Office Web Apps deployment
- Additional configuration (optional)
- Troubleshooting

In addition to the previous sections, one of the following sections will apply to your specific Office Web Apps deployment:

- Install and configure Office Web Apps on an existing stand-alone SharePoint server
- Install and configure Office Web Apps on a new stand-alone SharePoint server

- Install and configure Office Web Apps on an existing SharePoint server farm
- Install and configure Office Web Apps on a new SharePoint server farm

Understanding Office Web Apps deployment

For both stand-alone SharePoint servers and SharePoint server farms, deploying Office Web Apps involves three primary phases:

Running setup and PSConfig Tasks include running Setup.exe and SharePoint Products and Technologies Post Setup and Configuration Wizard (PSConfig) on a stand-alone SharePoint server or each server in a SharePoint server farm. Running Setup.exe installs Office Web Apps files and components on a server. Running PSConfig is required as part of Office Web Apps setup in order to register the Office Web Apps services and, depending on the SharePoint installation type, start the service instances, create the service applications and service application proxies, and activate the Office Web Apps Feature.

Activating the Office Web Apps services Includes starting the service instances, and creating the service applications and service application proxies. Whether you must activate the services will depend on the state of SharePoint and whether PSconfig and the SharePoint Farm Configuration Wizard have previously been run.

Activating the Office Web Apps Feature Includes activating the Office Web Apps Feature on all existing SharePoint site collections where the Office Web Apps should be available. If PSconfig or the SharePoint Farm Configuration Wizard have been run before installing Office Web Apps, at least one site collection will exist. The feature will be activated automatically for new site collections created after Office Web Apps is installed.

Note:
Some procedures in this article require using Windows PowerShell or SharePoint 2010 Management Shell to run cmdlets. To run SharePoint 2010 cmdlets in Windows PowerShell, you must add the Microsoft.SharePoint.PowerShell snap-in by using the **Add-PSSnapin** cmdlet, or you can run the SharePoint 2010 cmdlets by using SharePoint 2010 Management Shell. By default, the Microsoft.SharePoint.PowerShell snap-in is already registered, and the snap-in is added to the SharePoint 2010 Management Shell. To run SharePoint 2010 cmdlets, you must be a member of the SharePoint_Shell_Access role on the configuration database and a member of the WSS_ADMIN_WPG local group on the computer where SharePoint 2010 Products is installed. To run scripts in Windows PowerShell or SharePoint 2010 Management Shell, you must set the execution policy by using the **set-executionpolicy** cmdlet, using the **unrestricted** parameter. For more information about the **Add-PSSnapin** cmdlet, see Add-PSSnapin (*http://go.microsoft.com/fwlink/?LinkId=188450&clcid=0x409*). For more information about how to use Windows PowerShell with SharePoint 2010 Products, see SharePoint 2010 Products administration by using Windows PowerShell (*http://technet.microsoft.com/library/ae4901b4-505a-42a9-b8d4-fca778abc12e(Office.14).aspx*).

Install and configure Office Web Apps on an existing stand-alone SharePoint server

This section applies only if you are installing Office Web Apps on an existing SharePoint server and PSConfig was previously run as part of SharePoint setup.

Caution:
When you run Setup.exe, Office Web Apps setup configures the default open behavior for browser-enabled documents in SharePoint to open documents in the browser. If Office Web Apps setup was run, but the Office Web Apps Services and Feature has not yet been activated, a user may get a broken link when opening a document in the browser. When deploying Office Web Apps on a live production server farm, to prevent broken links to documents while completing additional deployment tasks after running setup, we recommend you enable the OpenInClient feature on existing site collections before running setup. For more information, see Additional configuration (optional).

Run Office Web Apps setup

Complete this task to install Office Web Apps components and files on a server.

To run Office Web Apps setup

1. From the root folder, run **Setup.exe**.
2. On the **Enter your Product Key** page, enter your product key, and then click **Continue**.
3. On the **Choose a file location** page, click **Install Now** to install to the default location. To install to a different location, specify the location that you want to install to and then click **Install Now**.
4. When Setup finishes, a dialog box prompts you to complete the configuration of your server. Be sure that the **Run the SharePoint Products and Technologies Configuration Wizard now** check box is selected and then click **Close** to start PSConfig.

Run PSConfig to register the services

Complete this task to register the Office Web Apps services on the SharePoint server.

To run PSConfig to register the services

1. If you left the **Run the SharePoint Products and Technologies Configuration Wizard now** check box selected in the previous step, on the PSconfig **Welcome to SharePoint Products** page, click **Next**.
2. In the dialog box that notifies you that some services might have to be restarted or reset during configuration, click **Yes**.
3. On the **Configuration Successful** page, click **Finish**. Your new SharePoint site opens.

Start the service instances

A service instance provides the physical location for a service application. You must start the service instances before you create the service applications and the service application proxies. You can start the service instances by using SharePoint Central Administration or by using Windows PowerShell.

Procedures in this task start the service instances on the server specified.

To start the service instances by using Central Administration

1. Click **Start**, point to **All Programs**, **Microsoft SharePoint 2010 Products**, and then **SharePoint 2010 Central Administration**.
2. On the SharePoint Central Administration home page, in **System Settings**, click **Manage services on this server**.
3. On the Services on server:<servername>page, start **Excel Calculation Services**, **Word Viewing Service**, and **PowerPoint Service**. The OneNote Web App does not use a SharePoint service.

To start the service instances by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

```
$machinesToActivate = @("<servername>")
$serviceInstanceNames = @("Word Viewing Service", "PowerPoint Service", "Excel
Calculation Services")
foreach ($machine in $machinesToActivate)
{
  foreach ($serviceInstance in $serviceInstanceNames)
  {
     $serviceID = $(Get-SPServiceInstance | where {$_.TypeName -match
$serviceInstance} | where {$_.Server -match "SPServer Name="+$machine}).ID
     Start-SPServiceInstance -Identity $serviceID
  }
}
```

2. Specify the following parameters:

Parameter	Value
$machinesToActivate	Server name

3. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).
4. From the Windows PowerShell command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

Create the service applications and the service application proxies

After the service instances have been started, the service applications and the service application proxies which connect the SharePoint Web front-ends to the service applications must be created. You can create the service applications and the service application proxies for the Word, PowerPoint, and Excel Web apps by using Central Administration or by using Windows PowerShell. The OneNote Web App does not require a service application. After created, the service applications will run on started service instances automatically.

Procedures in this task will create the Office Web Apps service applications and service application proxies for the Word, PowerPoint, and Excel Web apps. This task can only be completed after you have started the service instances.

To create the service applications and the service application proxies by using Central Administration

1. Click **Start**, point to **All Programs**, **Microsoft SharePoint 2010 Products**, and then **SharePoint 2010 Central Administration**.
2. On the SharePoint Central Administration home page, in **Application Management**, click **Manage service applications**.
3. On the Service Applications page, click **New**, and then click **Word Viewing Service**.
4. In the Word Viewing Service Application dialog box, in **Name**, type **Word Viewing Service Application**. In **Application Pool**, select **Use existing application pool**, and then in the listbox, select **SharePoint Web Services Default**. In **Add to default proxy list**, verify **Add this service application's proxy to the farm's default proxy list** is selected (default), and then click **OK**.
5. On the Service Applications page, click **New**, and then click **PowerPoint Service Application**.
6. In the PowerPoint Service Application dialog box, in **Name**, type **PowerPoint Service Application**. In **Application Pool**, select **Use existing application pool**, and then in the listbox, select **SharePoint Web Services Default**. In **Add to default proxy list**, verify **Add this service application's proxy to the farm's default proxy list** is selected (default), and then click **OK**.
7. On the Service Applications page, click **New**, and then click **Excel Services Application**.
8. In the Excel Services Application dialog box, in **Name**, type **Excel Services Application**. In **Application Pool**, select **Use existing application pool**, and then in the listbox, select **SharePoint Web Services Default**. In **Add to default proxy list**, verify **Add this service**

application's proxy to the farm's default proxy list is selected (default), and then click **OK**.

Note:
You can choose to create a new application pool to be used with a service application. When creating a new application pool, you can specify the security account used by the application pool to be a predefined Network Service account, or you can specify a managed account. The account must have read\write privileges for the SPContent database and SPConfig database. For more information about services account permissions in SharePoint, see Account permissions and security settings (SharePoint Server 2010) (*http://technet.microsoft.com/library/55b99d80-3fa7-49f0-bdf4-adb5aa959019(Office.14).aspx*).

To create the service applications and the service application proxies by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

```
$appPool = Get-SPServiceApplicationPool -Name "SharePoint Web Services Default"

New-SPWordViewingServiceApplication -Name "WdView" -ApplicationPool $appPool |

New-SPWordViewingServiceApplicationProxy -Name "WdProxy"

New-SPPowerPointServiceApplication -Name "PPT" -ApplicationPool $appPool | New-

SPPowerPointServiceApplicationProxy -Name "PPTProxy"

New-SPExcelServiceApplication -Name "Excel" -ApplicationPool $appPool
```

2. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).
3. From the Windows PowerShell command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

Activate the Office Web Apps Feature

After the service instances have been started, and the service applications and service application proxies have been created, to use Office Web Apps require the Office Web Apps Feature be activated on existing site collections. You can activate the feature on a single site collection in a browser on the Site collection features page or by using Windows PowerShell. If you have a large number of site collections, you can activate the feature on all site collections at the same time by using Windows PowerShell.

Procedures in this task will activate the Office Web Apps Feature on one or more existing site collections. This task must be completed only once after the service applications and the service application proxies have been created. If you have more than one existing site collection, and you are

activating the feature on one site collection at a time, you may have to perform the procedure for each site collection.

Note:
This task does not need to be completed for new site collections created after Office Web Apps has been installed.

To activate the Office Web Apps Feature on a site collection on the Site collection features page

1. In a browser, in the SharePoint site, click **Site Actions**, and then click **Site Settings**.
2. On the Site Settings page, in **Site Collection Administration**, click **Site Collection Features**.
3. On the Features page, for **Office Web Apps**, click **Activate**.

To activate the Office Web Apps Feature on a site collection by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

```
$webAppsFeatureId = $(Get-SPFeature -limit all | where {$_.displayname -eq
"OfficeWebApps"}).Id
$singleSiteCollection = Get-SPSite -Identity http://<site_name> Enable-SPFeature
$webAppsFeatureId -Url $singleSiteCollection.URL
```

2. Specify the following parameters:

Parameter	Value
–Identity	URL

3. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).
4. In the Windows PowerShell console, at the command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

To activate the Office Web Apps Feature on all site collections by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

```
$webAppsFeatureId = $(Get-SPFeature -limit all | where {$_.displayname -eq
"OfficeWebApps"}).Id
Get-SPSite -limit ALL |foreach{Enable-SPFeature $webAppsFeatureId -url $_.URL }
```

2. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).
3. From the Windows PowerShell command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

Install and configure Office Web Apps on a new stand-alone SharePoint server

This section applies only if you are installing Office Web Apps on a new SharePoint installation where PSConfig has not previously been run as part of SharePoint setup.

Run Office Web Apps setup

Complete this task to install Office Web Apps components and files on a server.

To run Office Web Apps setup

1. From the root folder, run **Setup.exe**.
2. On the **Enter your Product Key** page, enter your product key, and then click **Continue**.
3. On the **Choose a file location** page, click **Install Now** to install to the default location. To install to a different location, specify the location that you want to install to, and then click **Install Now**.
4. When Setup finishes, a dialog box prompts you to complete the configuration of your server. Be sure that the **Run the SharePoint Products and Technologies Configuration Wizard now** check box is selected, and then click **Close** to start PSConfig.

Run PSConfig to register the services, start the service instances, create the service applications and proxies, and activate the Office Web Apps Feature

Complete this task to register the services, start the service instances, create the service applications and service application proxies, and activate the Office Web Apps Feature.

To run PSConfig to register the services, start the service instances, create the service applications and proxies, and activate the Office Web Apps Feature

1. If you left the **Run the SharePoint Products and Technologies Configuration Wizard now** check box selected in the previous task, on the PSconfig **Welcome to SharePoint Products** page, click **Next**.
2. In the dialog box that notifies you that some services might need to be restarted or reset during configuration, click **Yes**.
3. On the **Configuration Successful** page, click **Finish**. Your new SharePoint site opens.

Install and configure Office Web Apps on an existing SharePoint server farm

Peform the tasks in this section only if you are installing Office Web Apps on an existing SharePoint server farm where the Farm Configuration Wizard has previously been run.

Caution:
When you run Setup.exe, Office Web Apps setup configures the default open behavior for browser-enabled documents in SharePoint to open documents in the browser. If Office Web Apps setup has been run, but the Office Web Apps Services and Feature has not yet been activated, a user may get a broken link when opening a document in the browser. When deploying Office Web Apps on a live production server farm, to prevent broken links to documents while completing additional deployment tasks after running setup, it is recommended you enable the OpenInClient feature on existing site collections prior to running setup. For more information, see Additional configuration (optional).

Run Office Web Apps setup

Complete this task to install Office Web Apps on a single SharePoint server. This task must be performed on each server in the server farm.

To run Office Web Apps Setup

1. From the root folder, run **Setup.exe**.
2. On the **Enter your Product Key** page, enter your product key, and then click **Continue**.
3. On the **Choose a file location** page, click **Install Now** to install to the default location. To install to a different location, specify the location that you want to install to, and then click **Install Now**.
4. When Setup finishes, a dialog box prompts you to complete the configuration of your server. Be sure that the **Run the SharePoint Products and Technologies Configuration Wizard now** check box is selected.
5. Click **Close** to start the configuration wizard.

Run PSConfig to register services

Complete this task to register the Office Web Apps services on a single SharePoint server. This task must be performed on each server in the server farm.

To run PSConfig to register the services

1. On the **Welcome to SharePoint Products** page, click **Next**.
2. In the dialog box that notifies you that some services might need to be restarted or reset during configuration, click **Yes**.

3. On the **Modify server farm settings** page, select **Do not disconnect from this server farm**, and then click **Next**.
4. On the **Configuration Successful** page, click **Finish**. Your new SharePoint site opens.

Start the service instances

A service instance provides the physical location for a service application. For each server that you want to run the Office Web Apps service applications; you must start the service instances. You can start the service instances by using SharePoint Central Administration or by using Windows PowerShell.

Procedures in this task will start the service instances on those servers specified. This task must be completed after you have run WCSetup and PSConfig on each server in the farm.

To start the service instances by using Central Administration

1. Click **Start**, point to **All Programs**, **Microsoft SharePoint 2010 Products**, and then **SharePoint 2010 Central Administration**.
2. On the SharePoint Central Administration home page, in **System Settings**, click **Manage services on this server**.
3. On the Services on server:<servername>page, in **Server**, select a server, and then start **Excel Calculation Services**, **Word Viewing Service**, and **PowerPoint Service**. Repeat this step for each server in the farm you want to run Office Web Apps services. The OneNote Web App does not use a SharePoint service.

To start the service instances by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

```
$machinesToActivate = @("<servername1>", "<servername2>")

$serviceInstanceNames = @("Word Viewing Service","PowerPoint Service","Excel
Calculation Services")

foreach ($machine in $machinesToActivate)

{

  foreach ($serviceInstance in $serviceInstanceNames)

  {

     $serviceID = $(Get-SPServiceInstance | where {$_.TypeName -match
$serviceInstance} | where {$_.Server -match "SPServer Name="+$machine}).ID

     Start-SPServiceInstance -Identity $serviceID

  }

}
```

2. Specify the following parameters:

Parameter	Value
$machinesToActivate	Server name

3. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).
4. From the Windows PowerShell command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

Create the service applications and the service application proxies

After the service instances have been started, the service applications and the service application proxies that connect the SharePoint Web front-ends to the service applications must be created. You can create the service applications and the service application proxies for the Word, PowerPoint, and Excel Web apps by using Central Administration or by using Windows PowerShell. The OneNote Web app does not require a service application. After created, the service applications will run on started service instances automatically.

In this task you will create the Office Web Apps service applications and service application proxies for the Word, PowerPoint, and Excel Web apps. This task can be completed only after you have started the service instances.

To create the service applications and the service application proxies by using Central Administration

1. Click **Start**, point to **All Programs**, **Microsoft SharePoint 2010 Products**, and then **SharePoint 2010 Central Administration**.
2. On the SharePoint Central Administration home page, in **Application Management**, click **Manage service applications**.
3. On the Service Applications page, click **New**, and then click **Word Viewing Service**.
4. In the Word Viewing Service Application dialog box, in **Name**, type **Word Viewing Service Application**. In **Application Pool**, select **Use existing application pool**, and then in the listbox, select **SharePoint Web Services Default**. In **Add to default proxy list**, verify **Add this service application's proxy to the farm's default proxy list** is selected (default), and then click **OK**.
5. On the Service Applications page, click **New**, and then click **PowerPoint Service Application**.

6. In the PowerPoint Service Application dialog box, in **Name**, type **PowerPoint Service Application**. In **Application Pool**, select **Use existing application pool**, and then in the listbox, select **SharePoint Web Services Default**. In **Add to default proxy list**, verify **Add this service application's proxy to the farm's default proxy list** is selected (default) and then click **OK**.
7. On the Service Applications page, click **New** and then click **Excel Services Application**.
8. In the Excel Services Application dialog box, in **Name**, type **Excel Services Application**. In **Application Pool**, select **Use existing application pool**, and then in the listbox, select **SharePoint Web Services Default**. In **Add to default proxy list**, verify **Add this service application's proxy to the farm's default proxy list** is selected (default) and then click **OK**.

Note:
You can choose to create a new application pool to be used with a service application. When creating a new application pool, you can specify the security account used by the application pool to be a predefined Network Service account, or you can specify a managed account. The account must have read\write privileges for the SPContent database and SPConfig database. For more information about services account permissions in SharePoint, see Account permissions and security settings (SharePoint Server 2010) (*http://technet.microsoft.com/library/55b99d80-3fa7-49f0-bdf4-adb5aa959019(Office.14).aspx*).

To create the service applications and the service application proxies by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

```
$appPool = Get-SPServiceApplicationPool -Name "SharePoint Web Services Default"
New-SPWordViewingServiceApplication -Name "WdView" -ApplicationPool $appPool |
New-SPWordViewingServiceApplicationProxy -Name "WdProxy"
New-SPPowerPointServiceApplication -Name "PPT" -ApplicationPool $appPool | New-
SPPowerPointServiceApplicationProxy -Name "PPTProxy"
New-SPExcelServiceApplication -Name "Excel" -ApplicationPool $appPool
```

2. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).
3. From the Windows PowerShell command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

Activate the Office Web Apps Feature

After the service instances have been started, and the service applications and service application proxies have been created, to use Office Web Apps require the Office Web Apps Feature be activated on existing site collections. You can activate the feature on a single site collection in a browser on the Site collection features page or by using Windows PowerShell. If you have a large number of site collections, you can activate the feature on all site collections at the same time by using Windows PowerShell.

Procedures in this task will activate the Office Web Apps Feature on one or more existing site collections. This task must be completed only once after the service applications and the service application proxies have been created. If you have more than one existing site collection, and you are activating the feature on one site collection at a time, you may have to perform the procedure for each site collection.

Note:
This task does not need to be completed for new site collections created after Office Web Apps has been installed.

To activate the Office Web Apps Feature on a site collection on the Site collection features page

1. In a browser, in the SharePoint site, click **Site Actions**, and then click **Site Settings**.
2. On the Site Settings page, in **Site Collection Administration**, click **Site Collection Features**.
3. On the Features page, for **Office Web Apps**, click **Activate**.

To activate Office Web Apps Feature on a site collection by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

```
$webAppsFeatureId = $(Get-SPFeature -limit all | where {$_.displayname -eq
"OfficeWebApps"}).Id
$singleSiteCollection = Get-SPSite -Identity http://<site_name> Enable-SPFeature
$webAppsFeatureId -Url $singleSiteCollection.URL
```

2. Specify the following parameters:

Parameter	Value
–Identity	URL

3. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).

4. In the Windows PowerShell console, at the command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

To activate the Office Web Apps Feature on all site collections by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

```
$webAppsFeatureId = $(Get-SPFeature -limit all | where {$_.displayname -eq
"OfficeWebApps"}).Id

Get-SPSite -limit ALL |foreach{Enable-SPFeature $webAppsFeatureId -url $_.URL }
```

2. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).
3. From the Windows PowerShell command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

Install and configure Office Web Apps on a new SharePoint server farm

Perform the tasks in this section only if you are installing Office Web Apps on a new SharePoint server farm where the Farm Configuration Wizard has not previously been run.

Run Office Web Apps setup

In this task you will install Office Web Apps files and components on a single SharePoint server in a new server farm where the Farm Configuration Wizard has not previously been run. This task must be completed on each server in the server farm.

To run Office Web Apps Setup

1. From the root folder, run **Setup.exe**.
2. On the **Enter your Product Key** page, enter your product key, and then click **Continue**.
3. On the **Choose a file location** page, click **Install Now** to install to the default location. To install to a different location, specify the location that you want to install to, and then click **Install Now**.
4. When Setup finishes, a dialog box prompts you to complete the configuration of your server. Be sure that the **Run the SharePoint Products and Technologies Configuration Wizard now** check box is selected.
5. Click **Close** to start the Farm Configuration Wizard.

Run PSConfig to register services

In this task you will register the Office Web Apps services on a single SharePoint server. This task must be completed on each server in the server farm.

To run PSConfig to register the services

1. On the **Welcome to SharePoint Products** page, click **Next**.
2. In the dialog box that notifies you that some services might need to be restarted or reset during configuration, click **Yes**.
3. On the **Modify server farm settings** page, select **Do not disconnect from this server farm**, and then click **Next**.
4. On the **Configuration Successful** page, click **Finish**. Your new SharePoint site opens.

Run the SharePoint Farm Configuration Wizard to start the service instances, create the service applications and proxies, and activate the Office Web Apps Feature

In this task you will start the service instances on all servers in the farm, create the service applications and service application proxies, and activate the Office Web Apps Feature on all existing site collections. This task must be completed after Setup.exe and PSConfig has been run on each server in the server farm.

To run the SharePoint Farm Configuration Wizard to start the service instances, create the service applications and proxies, and activate the Office Web Apps Feature

1. Click **Start**, point to **All Programs**, **Microsoft SharePoint 2010 Products**, and then **SharePoint 2010 Central Administration**.
2. On the **SharePoint Central Administration** home page, click **Configuration Wizards**.
3. On the **Configuration Wizards** page, click **Launch the Farm Configuration Wizard**.
4. In the **Farm Configuration Wizard** welcome page, choose **Walk me through the settings using this wizard**, and then click **Next**.
5. On the **Configure your SharePoint Farm** page, in **Service Account**, type a name for the Farm admin account.
6. In **Services**, select the Office Web Apps services that you want to activate, and then click **Next**.
7. Create an optional new top-level site. On the **Create Site Collection** page, follow the wizard steps to create a new top-level site.
8. On the **Configure your SharePoint Farm** page, click **Finish**.

Additional configuration (optional)

This section discusses additional configurations that are optional.

Configure the SharePoint default open behavior for browser-enabled documents

In SharePoint, you can configure whether browser-enabled documents are opened in a client application or in the browser. By default, when Office Web Apps is installed, Office documents will then open in the browser. You can override this setting using the SharePoint OpenInClient feature. The OpenInClient feature can be configured in Central Administration or by using the SPFeature cmdlet in Windows PowerShell.

How documents open in SharePoint varies depending on whether or not the OpenInClient feature is present, and either enabled or disabled:

- If the OpenInClient feature is not present and Office Web Apps is not installed, documents will open in the client application (SharePoint default).
- If the OpenInClient feature is not present, Office Web Apps is installed and Office Web Apps service applications are activated, documents will open in the browser (Office Web Apps default).
- If the OpenInClient feature is present and enabled, and Office Web Apps service applications are activated, documents will open in the client application.
- If the OpenInClient feature is present and disabled, and Office Web Apps service applications are activated, documents in will open in the browser.

Caution:
When you run Setup.exe to install Office Web Apps, setup will take control of the default open behavior in SharePoint to register Word, PowerPoint, Excel, and OneNote documents to be opened in their associated Web app. If a user clicks on a document in SharePoint after Setup.exe has been run, but before the Office Web Apps Services and Feature have been activated, the user can get a broken link in the browser. When installing Office Web Apps in a live production environment, it is strongly recommended that you enable the OpenInClient Feature prior to running Office Web Apps setup.

To set the default open behavior for site collections by using Central Administration

1. In SharePoint Central Administration, click **Site Actions**, and then click **Site Settings**.
2. On the Site Settings page, under **Site Collection Administration**, click **Site Collection Features**.
3. On the Features page, for the **Open Documents in Client Applications by Default** feature, click **Activate** (OpenInClient Feature is enabled) to open documents in the client application. Click **Deactivate** (OpenInClient Feature is disabled) to open documents in the browser.

To set the SharePoint Default open behavior for browser-enabled documents to open in the browser by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

   This example disables the default open behavior in SharePoint.

   ```
   $defaultOpenBehaviorFeatureId = $(Get-SPFeature -limit all | where {$_.displayname -eq "OpenInClient"}).Id

   Get-SPSite -limit ALL |foreach{ Disable-SPFeature $defaultOpenBehaviorFeatureId -url $_.URL }
   ```

2. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).
3. In the Windows PowerShell console, at the command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

To set the SharePoint Default open behavior for browser-enabled documents to open in the client application by using Windows PowerShell

1. Using Notepad, open a new text file, and then copy and paste the following script into the file.

   This example sets the default open behavior for all documents in all sites to open in the client application (if available).

   ```
   $defaultOpenBehaviorFeatureId = $(Get-SPFeature -limit all | where {$_.displayname -eq "OpenInClient"}).Id

   Get-SPSite -limit ALL |foreach{ Enable-SPFeature $defaultOpenBehaviorFeatureId -url $_.URL }
   ```

2. Save the file with a **.ps1** file name extension to a folder where you run scripts (typically C:\scripts).
3. In the Windows PowerShell console, at the command prompt (that is, PS C:\>), type the following command and press ENTER:

   C:\<path>\<filename>.ps1

Troubleshooting

Problem Office Web Apps is installed, but documents do not open in their associated Web app in the browser.

Solution Verify the Office Web Apps Feature has been activated for the site collection in which the document resides. For more information, see Activate the Office Web Apps Feature.

Solution Verify the service instances have been started. For more information, see Start the service instances.

Solution Verify the service applications and proxies have been created. In SharePoint Central Administration, in **Application Management**, click **Manage service applications**. Verify the Word Viewing service application, PowerPoint service application, and Excel Services Application are started. If they are not started, verify the service instances have been started.

Solution Verify the SharePoint OpenInClient Feature is not enabled. For more information, see Additional configuration (optional).

Problem The Office Web Apps opens fine in view mode, but when a user clicks the Edit in Word, Edit in PowerPoint, or Edit in Excel button on the toolbar, an error is displayed.

Solution Verify that the Office Web Apps Feature is activated and the Word Viewing Service, PowerPoint Service, and Excel Calculation Services are started.

Problem When running setup, the product key will not validate.

Solution Verify you are installing an Office Web Apps version compatible with your version of SharePoint 2010 Products. Office Web Apps Trial Edition cannot be installed on a server with licensed SharePoint 2010 products.

Solution Verify you have a valid Microsoft Office 2010 volume license.