

[MS-SPBEPO]: SharePoint Back-End Protocols Overview

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

This document provides an overview of the SharePoint Back-End Protocols Overview Protocol Family. It is intended for use in conjunction with the Microsoft Protocol Technical Documents, publicly

available standard specifications, network programming art, and Microsoft Windows distributed systems concepts. It assumes that the reader is either familiar with the aforementioned material or has immediate access to it.

A Protocol Family System Document does not require the use of Microsoft programming tools or programming environments in order to implement the Protocols in the System. Developers who have access to Microsoft programming tools and environments are free to take advantage of them.

Abstract

This document describes the intended functionality of the SharePoint Back-End Protocols System and how the protocols in this system interact. It provides examples of some of the common user scenarios. It does not restate the processing rules and other details that are specific for each protocol. These details are described in the protocol specifications for each of the protocols and data structures that make up this system.

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.01	Major	Initial Availability
06/27/2008	1.0	Editorial	Revised and edited technical content
10/06/2008	1.01	Editorial	Revised and edited technical content
12/12/2008	1.02	Editorial	Revised and edited technical content
07/13/2009	1.03	Major	Revised and edited the technical content
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Major	Significantly changed the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.05	Minor	Clarified the meaning of the technical content.
11/15/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
03/18/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
04/11/2012	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.05	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	7
1.1 Glossary	8
1.2 References.....	9
2 Functional Architecture	16
2.1 Overview	16
2.1.1 Deployment Topology	16
2.1.1.1 SharePoint Products and Technologies Farm	17
2.1.1.2 Data Storage	18
2.1.1.3 Front-End Protocols	18
2.1.1.4 Back-End Protocols	19
2.1.2 Core Platform	20
2.1.2.1 Configuration Protocols.....	20
2.1.2.1.1 Configuration Database.....	20
2.1.2.1.2 Service Application Database.....	21
2.1.2.1.3 Usage Database.....	21
2.1.2.2 Content Storage	21
2.1.3 Extended Capabilities.....	21
2.1.3.1 Excel Services	22
2.1.3.2 Access Services	23
2.1.3.3 InfoPath Forms Services.....	24
2.1.3.4 User Profile Service.....	25
2.1.3.4.1 Core Service.....	26
2.1.3.4.1.1 Properties.....	27
2.1.3.4.1.2 Organizations.....	27
2.1.3.4.1.3 Profiles.....	27
2.1.3.4.1.4 Multiple Values.....	27
2.1.3.4.1.5 Vocabularies	28
2.1.3.4.1.6 Colleagues.....	28
2.1.3.4.1.7 Memberships	28
2.1.3.4.1.8 Profile Change Log.....	28
2.1.3.4.1.9 Social Data	29
2.1.3.4.2 Data Import and Export	29
2.1.3.4.2.1 Active Directory.....	29
2.1.3.4.2.2 LDAP v3 Directory	30
2.1.3.4.2.3 Business Data Connectivity (BDC) Catalog	30
2.1.3.4.3 SharePoint Products and Technologies Content Database	30
2.1.3.4.4 User Profile Service Database Indexing	30
2.1.3.4.5 Personalization and Targeting	31
2.1.3.4.6 Social Metadata	31
2.1.3.5 Search Services.....	32
2.1.3.5.1 Concepts.....	32
2.1.3.5.2 Inverted Indexes.....	32
2.1.3.5.3 Property Store	33
2.1.3.5.4 Crawling	33
2.1.3.5.5 Search Topology	34
2.1.3.6 Business Connectivity Services.....	36
2.1.3.7 Content Management Service.....	36
2.1.3.8 PerformancePoint Services.....	37
2.1.3.9 Word Automation Services.....	37

2.1.3.10	Session State	38
2.1.3.10.1	ASP.NET Session State.....	39
2.1.3.10.2	State Service	39
2.1.3.11	User Code Execution Service	40
2.1.3.12	Analytics Services	40
2.1.3.13	Visio Graphics Service	40
2.2	Protocol Summary	41
2.2.1	Core Platform Protocols.....	41
2.2.1.1	Configuration	41
2.2.1.2	Content Storage	42
2.2.2	Extended Capabilities Protocols	44
2.2.2.1	Excel Services	45
2.2.2.2	Access Services	45
2.2.2.3	InfoPath Forms Services	45
2.2.2.4	User Profile Service.....	46
2.2.2.5	Search Services.....	49
2.2.2.6	Business Connectivity Services.....	52
2.2.2.7	Content Management Service.....	53
2.2.2.8	PerformancePoint Services.....	53
2.2.2.9	Word Automation Services.....	53
2.2.2.10	Session State	54
2.2.2.11	User Code Execution	54
2.2.2.12	Analytics Services	54
2.2.2.13	Visio Graphics Service	55
2.3	Environment	55
2.3.1	Dependencies on This System.....	55
2.3.2	Dependencies on Other Systems/Components	55
2.4	Assumptions and Preconditions.....	56
2.5	Use Cases.....	56
2.5.1	Core Platform Use Cases	57
2.5.1.1	Create a File from the Client in a SharePoint Library	57
2.5.1.2	Use Claims-Based Identity to Connect Line-of-Business Applications to SharePoint Service Applications.....	59
2.5.2	Extended Capabilities Use Cases	60
2.5.2.1	Excel Services	60
2.5.2.1.1	Manage Requests from a Front-End Web Server.....	60
2.5.2.2	Access Services	62
2.5.2.2.1	Manage the Data Operation Request.....	62
2.5.2.3	InfoPath Forms Services	64
2.5.2.3.1	Use the State Service	64
2.5.2.4	User Profile Service.....	65
2.5.2.4.1	Retrieve a User's Profile	65
2.5.2.4.2	Tag a Document.....	66
2.5.2.5	Search Services.....	67
2.5.2.5.1	Create and Crawl a Content Source	67
2.5.2.6	Session State	68
2.5.2.6.1	Communicate with a Back-End Database Server	68
2.6	Versioning, Capability Negotiation, and Extensibility.....	69
2.7	Error Handling	69
2.8	Coherency Requirements	69
2.9	Security.....	69
2.10	Additional Considerations.....	69

3 Examples	70
3.1 Example 1: Create a File from the Client.....	70
3.2 Example 2: Configure a Crawl, Run a Crawl, and Query for Content	73
4 Microsoft Implementations	75
4.1 Product Behavior	75
5 Change Tracking	76
6 Index	77

1 Introduction

This document provides an informative overview of the back-end protocols that are implemented by Microsoft® Office SharePoint® Server 2007, Windows® SharePoint® Services 3.0, Microsoft® SharePoint® Server 2010, and Microsoft® SharePoint® Foundation 2010, referred to collectively in this document as SharePoint Products and Technologies, for the purposes of communication between SharePoint Products and Technologies components.

SharePoint Products and Technologies provides an extensible platform on which collaboration and other scenarios and features are built. Some examples include team-oriented collaboration Web sites, document and content publishing features, and the ability to create portals that publish data from systems external to the SharePoint Products and Technologies deployment, to name just a few. The SharePoint Products and Technologies installation usually includes multiple computers, called a **farm**.

Client applications, such as browsers, communicate with a **front-end Web server** within the farm to use the features provided by SharePoint Products and Technologies. Also, it is possible for external server applications, including those in other separate SharePoint Products and Technologies farms, to interact with front-end Web servers or a **service application** within a SharePoint Products and Technologies farm to use features from the farm or retrieve data from the farm. A SharePoint Products and Technologies front-end Web server can also communicate with service applications hosted on other servers within the farm to respond to user requests for data or to deliver a feature to a user or external server. In some cases, those service applications can be hosted on one or multiple **application servers** dedicated for that purpose within the farm.

Data created by users, as well as data used to configure and manage features and service applications provided by SharePoint Products and Technologies, is stored in databases hosted on back-end database servers. The main components of a SharePoint Products and Technologies farm are the front-end Web servers, application servers, and back-end database servers.

The diagram in section [2.1.1](#) provides a high-level overview of the system.

This document covers the protocols used between front-end Web servers and back-end database servers; application servers and back-end database servers; and front-end Web servers and application servers within SharePoint Products and Technologies. Where appropriate, the document also describes the relationship between the protocols and offers example scenarios in which they are used. This document is meant to facilitate an understanding of how to implement the protocols to support interoperability scenarios that involve SharePoint Products and Technologies. Many concepts that are described in this document at a high level are described in detail in [\[MS-WSSTS\]](#).

Because the back-end protocols are designed to rely on the internal specifics of SharePoint Products and Technologies, they are also subject to the SharePoint Products and Technologies updates and hotfixes. As a result, the implementation details of these protocols can vary between versions of SharePoint Products and Technologies. Microsoft will document the updates and hotfixes; however, backward compatibility might not be maintained.

Note that the T-SQL-based protocols, as well as some **Web services** and file-format protocols, have changed significantly between Office SharePoint Server 2007 and SharePoint Server 2010. They are considered to be completely different protocols, and cross-compatibility between versions is not supported. New versions of the protocol documents for SharePoint Server 2010 feature the same titles, with the addition of "Version 2." SharePoint Server 2010 does not support any of the T-SQL protocols not marked with "Version 2". This overview document covers both the original protocol version and the "Version 2" protocol documents, and, where appropriate, it calls out the appropriate protocol name or document short name.

In addition to the back-end protocols described in this overview document, SharePoint Products and Technologies implements several application (or front-end) protocols that are described in the SharePoint Front-End Protocols Overview [\[MS-SPFEPO\]](#). The SharePoint Products and Technologies front-end protocols are engineered to enforce correctness, coherence, and consistency on the front-end Web server. Accordingly, Microsoft® recommends that implementers use the protocols that are exposed by the front-end Web server for external extensibility and interoperability.

While this overview documentation of the protocols is complete and accurate, it is more challenging from an engineering perspective to create a custom implementation of the protocols that communicate directly with the SharePoint Products and Technologies databases and internal components than it is to use the front-end protocols designed specifically for this purpose. When using the back-end protocols directly, the implementer is responsible for employing business logic to enforce consistency and coherence in the SharePoint Products and Technologies databases. To ignore these considerations can easily result in data loss and corruption.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

- Active Directory authentication credential**
- directory service (DS)**
- GUID**
- LDAP**
- Lightweight Directory Access Protocol (LDAP)**
- property set**
- remote procedure call (RPC)**
- security identifier (SID)**
- Server Message Block (SMB)**

The following terms are defined in [\[MS-OFCGLOS\]](#):

- application server**
- ASP.NET**
- audience**
- back-end database server**
- Business Connectivity Services (BCS)**
- Business Data Connectivity (BDC)**
- content database**
- crawl rule**
- data connection**
- data macro**
- distribution list**
- document**
- document identifier**
- e-mail address**
- event**
- farm**
- font**
- front-end Web server**
- hyperlink**
- Hypertext Markup Language (HTML)**
- line-of-business (LOB) system**
- list**

list item
list of postings
member group
metadata index
partition
query
rights
search service application
Secure Store Service (SSS)
Security Assertion Markup Language (SAML)
security principal
security token service (STS)
service application
session state
shape
Simple Object Access Protocol (SOAP)
site
site collection
stored procedure
Structured Query Language (SQL)
term frequency
theme
token
Transact-Structured Query Language (T-SQL)
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
user profile
User Profile Service
user profile store
user session
Web Part
Web service
workflow

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

We conduct frequent surveys of the informative references to assure their continued availability. If you have any issue with finding an informative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-ADMWS] Microsoft Corporation, "[Search Service Administration Web Service Protocol Specification](#)".

[MS-ADS] Microsoft Corporation, "[Access Services Data Server Protocol Specification](#)".

[MS-ADTS] Microsoft Corporation, "[Active Directory Technical Specification](#)".

[MS-ASPSS] Microsoft Corporation, "[ASP.NET State Service Database Repository Communications Protocol Specification](#)".

[MS-BDCDPS] Microsoft Corporation, "[Business Data Connectivity Database Protocol Specification](#)".

[MS-BDCMFFS] Microsoft Corporation, "[Business Data Connectivity Model File Format Specification](#)".

[MS-BDCSP] Microsoft Corporation, "[Business Data Catalog Database Protocol Specification](#)".

[MS-BDCWPS] Microsoft Corporation, "[Business Data Connectivity Web Service Protocol Specification](#)".

[MS-CDEPLOY] Microsoft Corporation, "[Content Deployment Remote Import Web Service Protocol Specification](#)".

[MS-CIFO] Microsoft Corporation, "[Content Index Format Structure Specification](#)".

[MS-CIPROP] Microsoft Corporation, "[Index Propagation Protocol Specification](#)".

[MS-CIPROP2] Microsoft Corporation, "[Index Propagation Version 2 Protocol Specification](#)".

[MS-DADMWS] Microsoft Corporation, "[Search Admin Web Service Protocol Specification](#)".

[MSDN-ESO] Microsoft Corporation, "Excel Services Overview", <http://msdn.microsoft.com/en-us/library/ms546696.aspx>

[MSDN-IReadOnlySessionState] Microsoft Corporation, "IReadOnlySessionState Interface", [http://msdn.microsoft.com/en-us/library/system.web.sessionstate.ireadonlysessionstate\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.ireadonlysessionstate(VS.80).aspx)

[MSDN-IRequiresSessionState] Microsoft Corporation, "IRequiresSessionState Interface", [http://msdn.microsoft.com/en-us/library/system.web.sessionstate.irequiresessionstate\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.irequiresessionstate(VS.80).aspx)

[MSDN-ODBC] Microsoft Corporation, "Microsoft Open Database Connectivity (ODBC)", <http://msdn.microsoft.com/en-us/library/ms710252.aspx>

[MSDN-OLEDB] Microsoft Corporation, "Microsoft OLE DB", <http://msdn.microsoft.com/en-us/library/ms722784.aspx>

[MSDN-Page] Microsoft Corporation, "@ Page", [http://msdn.microsoft.com/en-us/library/ydy4x04a\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/ydy4x04a(VS.80).aspx)

[MSDN-SharePointSDK] Microsoft Corporation, "SharePoint Products and Technologies SDK: 2010 API Reference (Technical Preview)", July 2009, [http://msdn.microsoft.com/en-us/library/ee557253\(office.14\).aspx](http://msdn.microsoft.com/en-us/library/ee557253(office.14).aspx)

[MS-EMMSTORE] Microsoft Corporation, "[Enterprise Metadata Service Database Schema Specification](#)".

[MS-ESP] Microsoft Corporation, "[Excel Web Services Protocol Specification](#)".

[MS-ESREST] Microsoft Corporation, "[Excel Services REST Protocol Specification](#)".

[MS-ESURL] Microsoft Corporation, "[Excel Services Publishing Protocol Specification](#)".

[MS-EXSPWS] Microsoft Corporation, "[Excel Calculation Web Service Protocol Specification](#)".

[MS-EXSPWS2] Microsoft Corporation, "[Excel Calculation Web Service Version 2 Protocol Specification](#)".

[MS-EXSPXML] Microsoft Corporation, "[Excel Calculation Web Service XML Schema Specification](#)".

[MS-FSO] Microsoft Corporation, "[FAST Search System Overview](#)".

[MSFT-ADC] Microsoft Corporation, "Active Directory Collection", March 2003, <http://technet2.microsoft.com/WindowsServer/en/library/6f8a7c80-45fc-4916-80d9-16e6d46241f91033.mspx>

[MS-GLOADWS] Microsoft Corporation, "[Global Admin Web Service Protocol Specification](#)".

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[MS-MAVA] Microsoft Corporation, "[Microsoft Office SharePoint Server \(MOSS\) Analytics View Access Protocol Specification](#)".

[MS-MOSSUAN] Microsoft Corporation, "[SQL Usage Analytics Protocol Specification](#)".

[MS-NOTESWS] Microsoft Corporation, "[MS Search Lotus Notes Web Service Protocol Specification](#)".

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".

[MS-PPSAPP] Microsoft Corporation, "[PerformancePoint Services Application Server Protocol Specification](#)".

[MS-PPSSPROC] Microsoft Corporation, "[PerformancePoint Services Stored Procedures Protocol Specification](#)".

[MS-PRIMEPF] Microsoft Corporation, "[Deployment Package Format Specification](#)".

[MS-QSSWS] Microsoft Corporation, "[Search Query Shared Services Protocol Specification](#)".

[MS-SADMWS] Microsoft Corporation, "[Search Site Administration Web Service Protocol Specification](#)".

[MS-SEARCH] Microsoft Corporation, "[Search Protocol Specification](#)".

[MS-SITED3S] Microsoft Corporation, "[Site Data 2003 Web Service Protocol Specification](#)".

[MS-SMB] Microsoft Corporation, "[Server Message Block \(SMB\) Protocol Specification](#)".

[MS-SPFEPO] Microsoft Corporation, "[SharePoint Front-End Protocols Overview](#)".

[MS-SPPTC] Microsoft Corporation, "[User Code Execution Protocol Specification](#)".

[MS-SPSCLSP] Microsoft Corporation, "[SPSCrawl Stored Procedures Protocol Specification](#)".

[MS-SPSCLSP2] Microsoft Corporation, "[SPSCrawl Stored Procedures Version 2 Protocol Specification](#)".

[MS-SPSCRWL] Microsoft Corporation, "[SPSCrawl Web Service Protocol Specification](#)".

[MS-SPSETSP] Microsoft Corporation, "[SharePoint Subscription Settings Stored Procedures Protocol Specification](#)".

[MS-SPSETWS] Microsoft Corporation, "[SharePoint Subscription Settings Web Service Protocol Specification](#)".

[MS-SPSTATE] Microsoft Corporation, "[Temporary State Service Protocol Specification](#)".

[MS-SPTWS] Microsoft Corporation, "[Service Platform Topology Web Service Protocol Specification](#)".

[MS-SQLPADM] Microsoft Corporation, "[SQL Administration Protocol Specification](#)".

[MS-SQLPADM2] Microsoft Corporation, "[SQL Administration Version 2 Protocol Specification](#)".

[MS-SQLPGAT] Microsoft Corporation, "[SQL Gatherer Protocol Specification](#)".

[MS-SQLPGAT2] Microsoft Corporation, "[SQL Gatherer Version 2 Protocol Specification](#)".

[MS-SQLPQ] Microsoft Corporation, "[Search Service Database Query Protocol Specification](#)".

[MS-SQLPQ2] Microsoft Corporation, "[Search Service Database Query Version 2 Protocol Specification](#)".

[MS-SQP] Microsoft Corporation, "[MSSearch Query Protocol Specification](#)".

[MS-SQP2] Microsoft Corporation, "[MSSearch Query Version 2 Protocol Specification](#)".

[MS-SRCHTP] Microsoft Corporation, "[Search Topology Protocol Specification](#)".

[MS-SSDPS] Microsoft Corporation, "[Secure Store Database Protocol Specification](#)".

[MS-SSOSP] Microsoft Corporation, "[Single Sign-On Database Protocol Specification](#)".

[MS-SSP] Microsoft Corporation, "[Single Sign-On Protocol Specification](#)".

[MS-SSPSJ] Microsoft Corporation, "[SSP Scheduled Jobs Stored Procedures Protocol Specification](#)".

[MS-SSPSOS] Microsoft Corporation, "[SQL Configuration Object Stored Procedures Protocol Specification](#)".

[MS-SSWPS] Microsoft Corporation, "[Secure Store Web Service Protocol Specification](#)".

[MS-SYS] Microsoft Corporation, "[Windows System Overview](#)".

[MS-TDS] Microsoft Corporation, "[Tabular Data Stream Protocol Specification](#)".

[MS-UPASP] Microsoft Corporation, "[User Profile Admin Stored Procedures Protocol Specification](#)".

[MS-UIESP] Microsoft Corporation, "[User Profile Import and Export Stored Procedures Protocol Specification](#)".

[MS-UIEWS] Microsoft Corporation, "[User Profile Import and Export Web Service Protocol Specification](#)".

[MS-UPSAUD] Microsoft Corporation, "[User Profile Service Audiences Protocol Specification](#)".

[MS-UPSAUD2] Microsoft Corporation, "[User Profile Service Audiences Version 2 Protocol Specification](#)".

[MS-UPSCDAP] Microsoft Corporation, "[User Profile Synchronization \(UPS\): Configuration Database Data Access Protocol Specification](#)".

[MS-UPSCDS] Microsoft Corporation, "[User Profile Synchronization \(UPS\): Configuration Data Structure](#)".

[MS-UPSCHNG] Microsoft Corporation, "[User Profile Change Log Stored Procedure Protocol Specification](#)".

[MS-UPSCHNG2] Microsoft Corporation, "[User Profile Change Log Stored Procedure Version 2 Protocol Specification](#)".

[MS-UPSCP] Microsoft Corporation, "[User Profile Synchronization \(UPS\): Configuration Protocol Extensions](#)".

[MS-UPSCSP] Microsoft Corporation, "[User Profile Service Application Caching Stored Procedures Protocol Specification](#)".

[MS-UPSCWS] Microsoft Corporation, "[User Profile Service Application Caching Web Service Protocol Specification](#)".

[MS-UPSDBDAP] Microsoft Corporation, "[User Profile Synchronization \(UPS\): Database Data Access Protocol Specification](#)".

[MS-UPSDWS] Microsoft Corporation, "[User Profile Social Data Web Service Protocol Specification](#)".

[MS-UPSGRAD] Microsoft Corporation, "[User Profile Service Push Protocol Specification](#)".

[MS-UPSIMP] Microsoft Corporation, "[User Profile Import Protocol Specification](#)".

[MS-UPSLDAP] Microsoft Corporation, "[User Profile Synchronization \(UPS\): Lightweight Directory Access Protocol Version 3 Extensions](#)".

[MS-UPSMODS] Microsoft Corporation, "[User Profile Synchronization \(UPS\): Management Objects Data Structure](#)".

[MS-UPSPROF] Microsoft Corporation, "[User Profile Stored Procedures Protocol Specification](#)".

[MS-UPSPROF2] Microsoft Corporation, "[User Profile Stored Procedures Version 2 Protocol Specification](#)".

[MS-UPSRCHSP] Microsoft Corporation, "[User Profile Search Stored Procedures Protocol Specification](#)".

[MS-UPSSCXPP] Microsoft Corporation, "[User Profile Synchronization \(UPS\): Schema Exchange Protocol Profile](#)".

[MS-UPSSDSP] Microsoft Corporation, "[User Profile Social Data Stored Procedures Protocol Specification](#)".

[MS-UPSSYNC] Microsoft Corporation, "[User Profile Synchronization Stored Procedures Protocol Specification](#)".

[MS-UPSSYNC2] Microsoft Corporation, "[User Profile Synchronization Stored Procedures Version 2 Protocol Specification](#)".

[MS-UPWCFWS] Microsoft Corporation, "[User Profile Property Service Application Web Service Protocol Specification](#)".

[MS-USPSWS] Microsoft Corporation, "[User Profile Service Web Service Protocol Specification](#)".

[MS-USRPCH] Microsoft Corporation, "[User Profile Change Log Web Service Protocol Specification](#)".

[MS-UTSP] Microsoft Corporation, "[SharePoint Usage Tracking Stored Procedures Protocol Specification](#)".

[MS-VGSFF] Microsoft Corporation, "[Visio Graphics Service File Format \(.vdw\) Specification](#)".

[MS-VGSP] Microsoft Corporation, "[Visio Graphics Service Protocol Specification](#)".

[MS-WORDSSP] Microsoft Corporation, "[Word Automation Services Stored Procedures Protocol Specification](#)".

[MS-WORDSWCF] Microsoft Corporation, "[Word Automation Services WCF Service Protocol Specification](#)".

[MS-WSSCADM] Microsoft Corporation, "[Windows SharePoint Services Content Database Administrative Communications Protocol Specification](#)".

[MS-WSSCADM2] Microsoft Corporation, "[Windows SharePoint Services Content Database Administrative Communications Version 2 Protocol Specification](#)".

[MS-WSSCCSP] Microsoft Corporation, "[Windows SharePoint Services: Content Database Core List Schema and Site Provisioning Communications Protocol Specification](#)".

[MS-WSSCCSP2] Microsoft Corporation, "[Windows SharePoint Services Content Database Core List Schema and Site Provisioning Communications Version 2 Protocol Specification](#)".

[MS-WSSCFGD] Microsoft Corporation, "[Windows SharePoint Services: Configuration Database Communications Protocol Specification](#)".

[MS-WSSCFGD2] Microsoft Corporation, "[Windows SharePoint Services: Configuration Database Communications Version 2 Protocol Specification](#)".

[MS-WSSDLIM] Microsoft Corporation, "[Windows SharePoint Services: Content Database Document and List Item Management Communications Protocol Specification](#)".

[MS-WSSDLIM2] Microsoft Corporation, "[Windows SharePoint Services: Content Database Document and List Item Management Communications Version 2 Protocol Specification](#)".

[MS-WSSDM] Microsoft Corporation, "[Windows SharePoint Services: Content Database Data Migration Communications Protocol Specification](#)".

[MS-WSSDM2] Microsoft Corporation, "[Windows SharePoint Services: Content Database Data Migration Communications Version 2 Protocol Specification](#)".

[MS-WSSSEUX] Microsoft Corporation, "[Windows SharePoint Services: Content Database End-User Experience Communications Protocol Specification](#)".

[MS-WSSSEUX2] Microsoft Corporation, "[Windows SharePoint Services: Content Database End-User Experience Communications Version 2 Protocol Specification](#)".

[MS-WSSFO] Microsoft Corporation, "[Windows SharePoint Services: File Operations Database Communications Protocol Specification](#)".

[MS-WSSFO2] Microsoft Corporation, "[Windows SharePoint Services: File Operations Database Communications Version 2 Protocol Specification](#)".

[MS-WSSO] Microsoft Corporation, "[Windows SharePoint Services Overview](#)".

[MS-WSSPROG] Microsoft Corporation, "[Windows SharePoint Services: Content Database Programmability Extensions Communications Protocol Specification](#)".

[MS-WSSPROG2] Microsoft Corporation, "[Windows SharePoint Services: Content Database Programmability Extensions Communications Version 2 Protocol Specification](#)".

- [MS-WSSTS] Microsoft Corporation, "[Windows SharePoint Services Technical Specification](#)".
- [RFC2518] Goland, Y., Whitehead, E., Faizi, A., Carter, S., and Jensen, D., "HTTP Extensions for Distributed Authoring WebDAV", RFC 2518, February 1999, <http://www.ietf.org/rfc/rfc2518.txt>
- [RFC2616] Fielding, R., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>
- [RFC4510] Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", RFC 4510, June 2006, <http://www.rfc-editor.org/rfc/rfc4510.txt>
- [SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>
- [SOAP1.2/2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part2-20030624>
- [XMLSCHEMA] World Wide Web Consortium, "XML Schema", September 2005, <http://www.w3.org/2001/XMLSchema>

2 Functional Architecture

The following sections describe the functional architecture of the SharePoint Back-End Protocols system.

2.1 Overview

SharePoint Products and Technologies is a Web-based platform that provides infrastructure for building and deploying applications as well as a set of built-in capabilities and features. The following sections describe the high-level topology, architecture, and deployment of SharePoint Products and Technologies as related to back-end protocols.

For information about the front-end protocol capabilities of SharePoint Products and Technologies, see [\[MS-SPFEPO\]](#). For a technical description of SharePoint Products and Technologies concepts, see [\[MS-WSSTS\]](#). For an overview of basic SharePoint Products and Technologies file, print, user/group administration, see [\[MS-WSSO\]](#).

2.1.1 Deployment Topology

SharePoint Products and Technologies can be deployed in a wide range of topologies to provide various degrees of scale-out and high availability, depending on the requirements of particular deployment scenarios. Particular deployments of one or more individual servers are farms. The following figure shows both the back-end and front-end SharePoint Products and Technologies protocol topology. Although these components can be deployed on individual server computers as shown in this figure, the components can also be run compatibly on a single server.

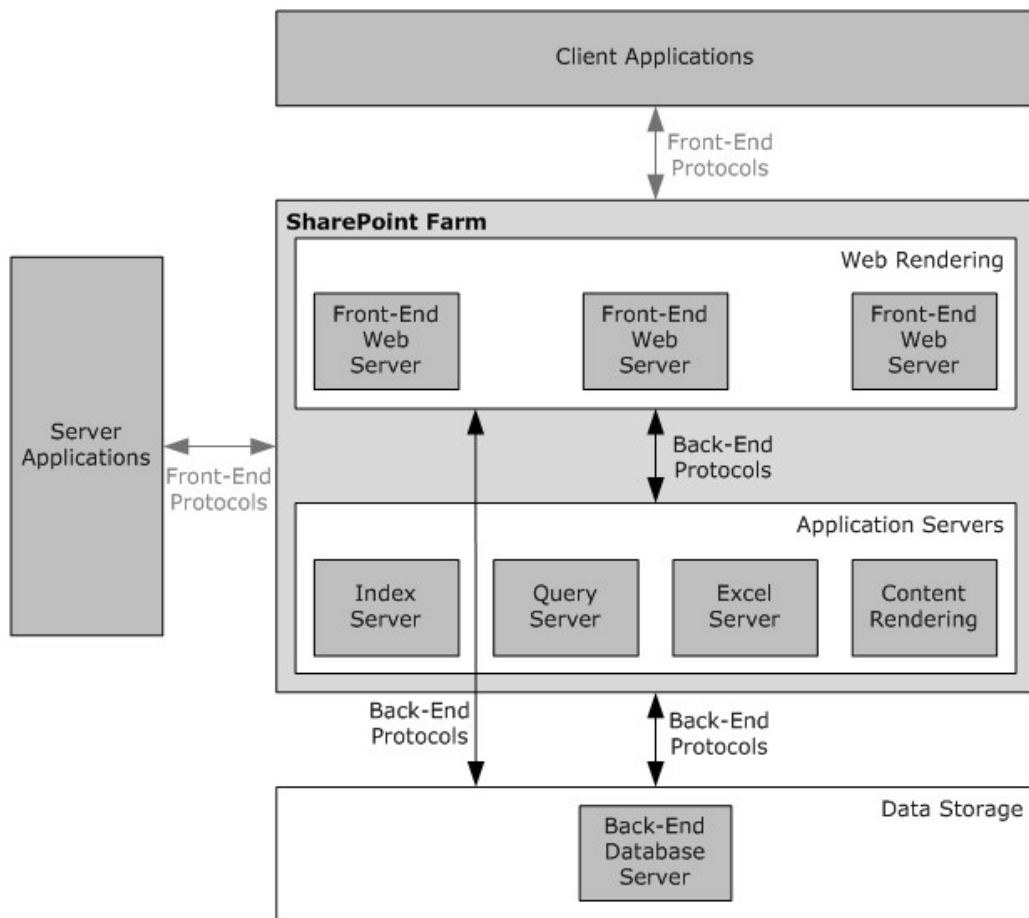


Figure 1: SharePoint Products and Technologies deployment topology

Deployments of SharePoint Products and Technologies can also add additional servers of each type, as required by the specific usage scenario. By providing additional computing resources, these servers (for example, front-end Web servers), can provide load-balancing and scalability.

The following sections briefly describe the SharePoint Products and Technologies deployment topology.

2.1.1.1 SharePoint Products and Technologies Farm

As shown in the figure in section [2.1.1](#) a SharePoint Products and Technologies deployment can be scaled-out across multiple servers in a farm deployment to increase throughput and availability.

SharePoint Products and Technologies supports network load-balancing technologies that distribute user requests across multiple servers in a farm. These individual front-end Web servers are stateless; that is, any front-end Web server in the farm is prepared to handle any user request in the same way as any other front-end Web server in the farm.

To provide additional resource management, SharePoint Products and Technologies also supports the ability to deploy servers as application servers to provide dedicated infrastructure and services for specific service applications or features that are provided by a farm. For example, a specific server could be created as the Excel® Services application server within a farm and be the resource

to do all Excel Services calculations for user requests to that farm. This allows any front-end Web servers to offload that possibly resource-heavy work and avoid affecting other user requests.

From a back-end storage perspective, as the SharePoint Products and Technologies deployment grows and the capacity of an individual back-end database server is fully consumed, additional back-end database server resources can be deployed by adding additional servers that host completely separate content databases. Different **site collections** can be deployed into those separate **content databases**, and when a user request comes to a particular front-end Web server, that front-end Web server will retrieve the site content strictly from the appropriate back-end database. This provides the ability to load-balance across multiple back-end resources, but does require manual placement of high-load sites into separate content databases.

2.1.1.2 Data Storage

SharePoint Products and Technologies stores user content and application configurations across multiple relational databases on the back-end database servers. At a high level, the following are the relational databases:

- **Content database:** Stores all user and site content. For information about content databases and protocols, see section [2.1.2.2](#).
- **Configuration database:** Stores topology and configuration information for the complete deployment. Every farm has exactly one configuration database. For information about the configuration database protocols, see section [2.1.2.1.1](#). For information about the user of the configuration database, see [\[MS-WSSO\]](#) section 2.1.2.6.
- **Central Administration content database:** Stores content associated with the Central Administration site.
- **Shared Services Administration content database:** Microsoft® Office SharePoint® Server 2007 stores content for the Shared Services Administration Web site in this database. This database is not used in Microsoft® SharePoint® Server 2010.
- **Service applications databases:** Stores content and configuration information for service applications, such as the **User Profile Service**. The capabilities of service applications are described in [\[MS-SPFEPO\]](#) section 2.1.2.2.

Service applications are a new model in SharePoint Server 2010. Office SharePoint Server 2007 used the concept of **Shared Services Providers (SSPs)**. For the purposes of this overview, SSPs and service applications can be treated as interchangeable, but it is important to note that they do use different protocols. For more information about service applications see [\[MSDN-SharePointSDK\]](#).

In SharePoint Server 2010, service applications use claims-based identity when communicating with each other, and they can use it when communicating with external systems. For more information about claims-based identity, and how claims-based identity is used with user **authentication** (2), see [\[MS-SPFEPO\]](#) section 2.1.2.4.1.

2.1.1.3 Front-End Protocols

The SharePoint Products and Technologies front-end protocols are used in integrations coming from outside the SharePoint Products and Technologies farm, for example requests by protocol client applications, such as browsers, to communicate with a front-end Web server within the farm to use specific features. It is also possible for external server applications, including those in other separate SharePoint Products and Technologies farms, to interact with front-end Web servers or a service

application within a SharePoint Products and Technologies farm to use features of the farm or retrieve data from the farm. For an informative overview of front-end protocols see [\[MS-SPFEPO\]](#).

The components that implement and respond to front-end protocols in turn use back-end protocols to do the following:

- Communicate with back-end database servers to store and retrieve user and configuration data.
- Communicate with application servers or other front-end Web servers within the farm to deliver specific features or have them in turn do the data retrieval.

2.1.1.4 Back-End Protocols

The back-end protocols described in this document are implemented by SharePoint Products and Technologies servers for communication within the farm and to enable communication between the SharePoint Products and Technologies components. Communication between the SharePoint Products and Technologies components can include the following:

- Front-end Web server to application server
- Front-end Web server to back-end database server
- Application server to application server
- Application server to back-end database server

The SharePoint Products and Technologies back-end protocols are implemented by using a variety of communication transports within the SharePoint Products and Technologies farm. These transports include the following:

- **Simple Object Access Protocol (SOAP)** Web services, as described in [\[SOAP1.1\]](#), [\[SOAP1.2/1\]](#), [\[SOAP1.2/2\]](#), and [\[RFC2616\]](#)
- **Server Message Block (SMB)**, as described in [\[MS-SMB\]](#)
- Tabular data stream (TDS), as described in [\[MS-TDS\]](#)

The back-end protocols are divided into two main groups: core platform and extended capabilities. The core platform protocols provide basic storage and administrative services, and the extended capabilities protocols build on top of those core services.

The core platform includes the following two groups of back-end protocols:

- Configuration
- Content storage

The extended capabilities include the following groups of back-end protocols:

- Excel® Services
- Microsoft® Access Services
- InfoPath Forms Services
- User Profile Service
- Search services

- Business Connectivity Services
- Content Management Service
- PerformancePoint Services
- Word Automation Services
- Session state
- User Code Execution Service
- Analytics Services
- Visio Graphics Service

The following sections summarize the core platform and extended capabilities groups of back-end protocols.

2.1.2 Core Platform

SharePoint Products and Technologies implements core data storage and system configuration related to the configuration, content, and service application databases described in section [2.1.1.2](#). In Microsoft® Office SharePoint® Server 2007, the core platform controlled the schema of the Shared Services Provider (SSP) database; in Microsoft® SharePoint® Server 2010, the service applications (which replaced SSPs) can specify the database schema for their application databases when calling the core system configuration protocols.

The core platform protocols described in this section implement the communication with these databases. At a high level, the core platform protocols are implemented as **Transact-Structured Query Language (T-SQL)** queries and **stored procedures** that are transported by using tabular data stream (TDS).

2.1.2.1 Configuration Protocols

This section covers the back-end configuration protocols that are used by all front-end and application servers to communicate with the SharePoint Products and Technologies configuration database. As described in section [2.1.2.1.1](#), the configuration database is used to manage all computers in the farm. This section also covers database creation for service applications (SSPs in Microsoft® Office SharePoint® Server 2007). The protocols used by the extended capability services applications to interact with their service application databases are covered in their respective sections within section [2.1.3](#). The final database covered here is the usage database, which is used to aggregate usage and diagnostic data used to help monitor usage, performance, and reliability of the SharePoint Products and Technologies farm.

2.1.2.1.1 Configuration Database

The configuration database describes the core SharePoint Products and Technologies farm configuration and structure. As discussed in section [2.1.1.2](#), each farm has exactly one configuration database.

The configuration database also defines how the farm uses the content databases. As described in section [2.1.2.2](#), content databases contain individual site collections. However, this partitioning is abstracted from users of SharePoint Products and Technologies front-end protocols; the **URL** namespace exposed by SharePoint Products and Technologies front-end Web servers does not directly indicate which content database stores a specific site collection. This mapping from URL to

content database is performed by using information stored in the configuration database. The information about how this mapping works is described in [\[MS-WSSO\]](#) section 2.1.2.6.1.

The configuration database also stores configuration objects that describe application settings across SharePoint Products and Technologies farms. This allows application settings across multiple servers in the farm to be maintained uniformly as configuration objects.

2.1.2.1.2 Service Application Database

A service application implemented in Microsoft® SharePoint® Server 2010 (SSP in Microsoft® Office SharePoint® Server 2007) includes capabilities to span multiple site collections in a deployment and can even work across multiple SharePoint Products and Technologies farms. Settings and data used by service applications are stored in the service application database. For more information about service applications, see [\[MS-SPFEPO\]](#) section 2.1.2.2 and [\[MSDN-SharePointSDK\]](#).

2.1.2.1.3 Usage Database

Microsoft® SharePoint® Server 2010 has a usage database that supports the storage, retrieval, and reporting of usage and diagnostic data. It is used by SharePoint Products and Technologies (and can be extended by third-party features built on SharePoint Products and Technologies) to store usage data of various kinds, including user-request information, performance counters, data on slow or expensive queries, and other relevant performance data. For more information see [\[MS-UTSP\]](#).

2.1.2.2 Content Storage

Content databases provide the core persistence for SharePoint site, list, and document library content. Information such as customizations to Web pages in a site and content additions to lists and libraries are stored in a content database. Individual site collections are contained entirely within individual content databases. They cannot span multiple content databases; however, SharePoint Products and Technologies farms do support vertical partitioning of site collections across content databases as a mechanism for scale-out. In this way, after the usage load exceeds capacity for an individual back-end database server, additional servers can be added to hold new site collections. For example, a deployment of 100,000 site collections might be divided between two separate back-end database servers, with each server holding the content for 50,000 site collections. Note that a single back-end database server will likely use more than one content database to store multiple site collections.

SharePoint Products and Technologies implements the protocols described in the table in section [2.2.1.2](#) to enable communication between the front-end Web server and the content database. These individual protocols can also be used to store configuration information in the configuration database. All content database protocols use the Tabular Data Stream (TDS) Protocol [\[MS-TDS\]](#) as a protocol transport.

2.1.3 Extended Capabilities

These sections provide an overview of the following extended capabilities, which build on the core platform protocols covered in previous sections of this document:

- Excel® Services
- Microsoft® Access Services
- InfoPath Forms Services
- User Profile Service

- Search services
- Business Connectivity Services
- Content Management Service
- PerformancePoint Services
- Word Automation Services
- Session state
- User Code Execution Service
- Analytics Services
- Visio Graphics Service

2.1.3.1 Excel Services

Excel® Services is composed of three main components: Microsoft® Excel® Calculation Services (ECS), Excel Web Access (EWA), and Excel Web Services.

The ECS component loads and calculates workbooks, applies user changes to workbooks, accesses external data that the workbooks might reference, and saves workbooks back to the SharePoint Products and Technologies content database. ECS can be deployed on a separate application server in the farm.

EWA interacts with ECS and renders the **HTML** representation of workbooks. Using EWA, a user can view and interact with a workbook. Excel Web Services provides programmatic access to Excel Services capabilities. Using Excel Web Services, a user can load, modify values in, calculate, and retrieve the calculated results of a workbook. ECS also allows more than one user to edit the same workbook at the same time. The following figure shows the high-level architecture of Excel Services and how it can be deployed in the farm topology. Though Excel Calculation Services can be deployed on a separate server as shown in this diagram, it can also be configured to run on the same server with EWA and Excel Web Services.

For technical information about the front-end protocols implemented by Excel Services, see [\[MS-ESP\]](#), [\[MS-ESURL\]](#), and [\[MS-ESREST\]](#). For technical information about Excel Calculation Web Services XML Specification see [\[MS-EXSPXML\]](#). For details about the protocol implemented by Excel Calculation Web Services in Microsoft® Office SharePoint® Server 2007 see [\[MS-EXSPWS\]](#). For details about the protocol implemented by Excel Calculation Web Services in Microsoft® SharePoint® Server 2010 see [\[MS-EXSPWS2\]](#).

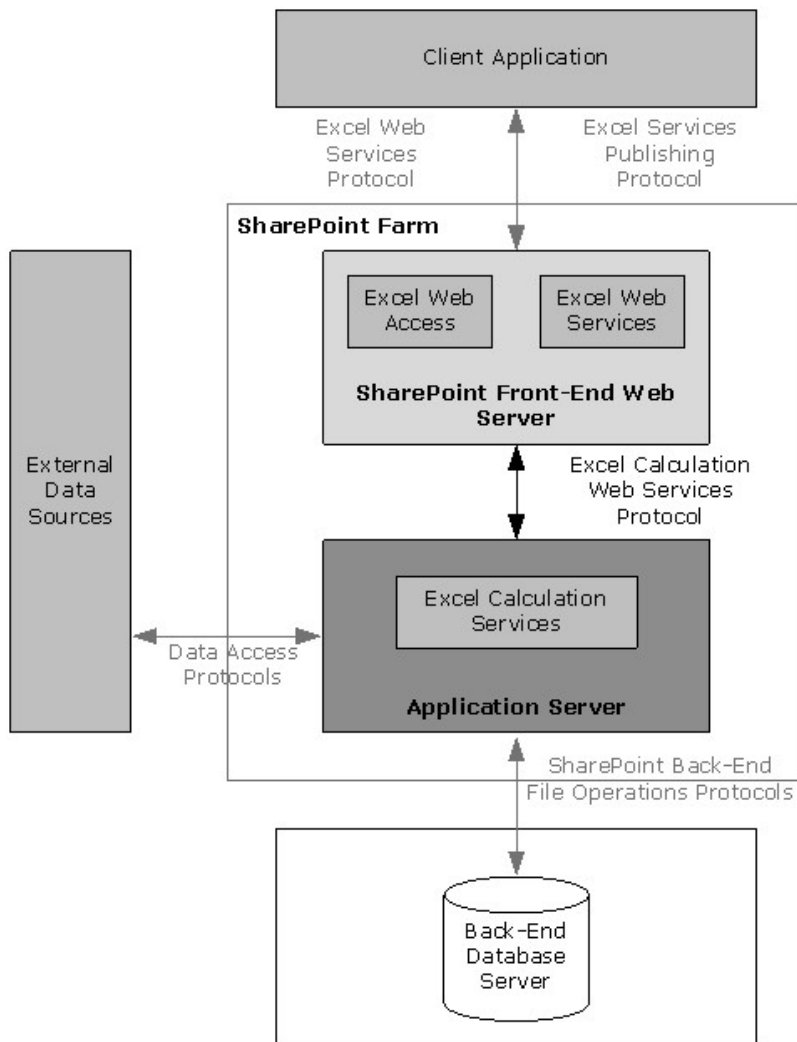


Figure 2: High-level architecture of Excel Services components

2.1.3.2 Access Services

Microsoft® SharePoint® Server 2010 implements Microsoft® Access Services, which is used to create database applications and manage data. For information about the Microsoft® Access® 2010 database applications, see [\[MS-SPFEPO\]](#) section 2.1.3.2.

Access Services is composed of two main components: Access Services Data Server (described in [\[MS-ADS\]](#)), which retrieves and manages data from SharePoint Products and Technologies storage for the database applications, and Access Services Web Services ([\[MS-SPFEPO\]](#) section 2.1.3.2), which provides the Web-based rendering for Access 2010 database applications.

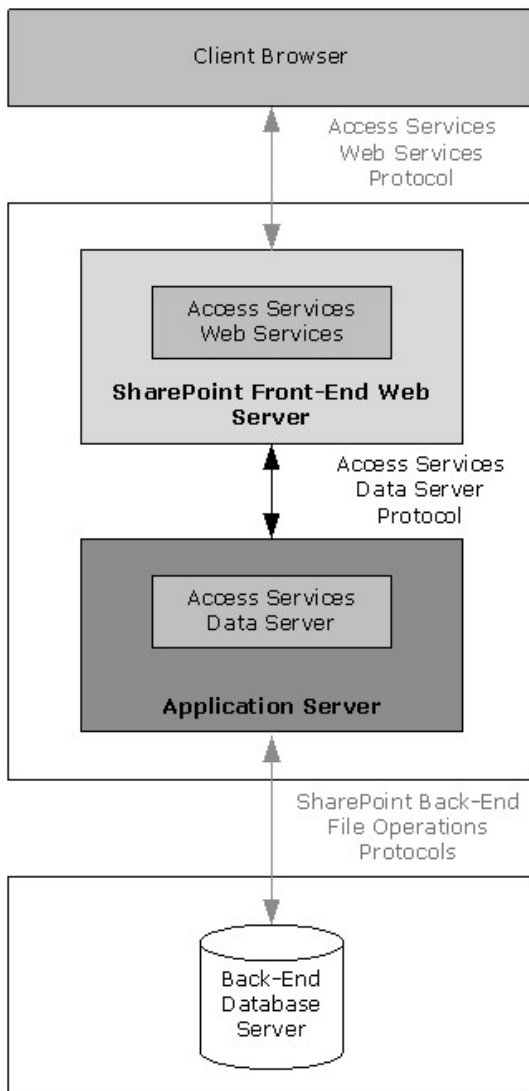


Figure 3: High-level architecture of Access Services components

2.1.3.3 InfoPath Forms Services

As described in [\[MS-SPFEPO\]](#) section 2.1.3.3, Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® Server 2010 with InfoPath® Forms Services enabled allow forms editing in the Web browser. After the form templates are published, Office SharePoint Server 2007 and SharePoint Server 2010 with InfoPath Forms Services can render them for display and form submission by users.

In Office SharePoint Server 2007, InfoPath Forms Services used **ASP.NET session state**, summarized in section [2.1.3.10.1](#).

In SharePoint Server 2010, InfoPath Forms Services uses the State Service, described in section [2.1.3.10.2](#).

2.1.3.4 User Profile Service

A User Profile Service stores information about people (users), groups, organizations, and related social activity. The information that is stored can be as basic as name, **e-mail address**, and telephone number, or it can be more specific such as employee start date, cost center, or skills and expertise. The activity information can be related to events such as a user's birthday, service anniversaries, or related to social activity such as applying tags to a document or a site. Groups and organization information includes data such as memberships, owners, and delegates. The User Profile Service in SharePoint Products and Technologies aggregates the information from multiple data sources (as described in [\[MS-UPSIMP\]](#) for Microsoft® Office SharePoint® Server 2007 and in [\[MS-UPIESP\]](#) and [\[MS-UPIEWS\]](#) for Microsoft® SharePoint® Server 2010) into a single repository, and provides a uniform user experience for this data. In addition, the User Profile Service provides programmatic access to this data as described in [\[MS-UPSPROF\]](#) for Office SharePoint Server 2007 and in [\[MS-UPSCSP\]](#), [\[MS-UPSCWS\]](#), and [\[MS-UPSPROF2\]](#) for SharePoint Server 2010.

In SharePoint Server 2010, the User Profile Service also implements support for retrieving social activity data, maintaining organization information, such as owners, members, and, delegates and adding social data to any **URI**. A URI can point to a resource or content inside or outside of SharePoint Products and Technologies. Social data can be in the form of numerical ratings, a note represented by lines of text, or keywords or terms, managed by the Content Management Service as described in section [2.1.3.7](#). Common usage scenarios for this metadata include rating a document, tagging content, or writing a descriptive line of text as a reminder note for some content.

The following is a list of actions that can be performed on the user profile data:

- Grouping users into audiences (as described in [\[MS-UPSAUD\]](#) for Office SharePoint Server 2007 and [\[MS-UPSAUD2\]](#) for SharePoint Server 2010).
- Indexing of the data for search (as described in [\[MS-SPSCRWL\]](#) and [\[MS-SPSCLSP\]](#) for Office SharePoint Server 2007, and in [\[MS-SPSCRWL\]](#) and [\[MS-SPSCLSP2\]](#) for SharePoint Server 2010).
- Redistributing the data to other SharePoint sites (as described in [\[MS-UPSSYNC\]](#) for Office SharePoint Server 2007 and [\[MS-UPSSYNC2\]](#) for SharePoint Server 2010).
- Importing and exporting data from external directory sources, such as Active Directory® Domain Services (AD DS) (as described in [\[MS-UPSIMP\]](#) for Office SharePoint Server 2007 and in [\[MS-UPIESP\]](#) and [\[MS-UPIEWS\]](#) for SharePoint Server 2010).
- Adding social metadata to a URI or collecting information regarding metadata on a URI for SharePoint Server 2010 (as described in [\[MS-UPSSDSP\]](#) for SharePoint Server 2010).
- The User Profile Service also allows a consistent way to access the change log for all data changes, regardless of the source of the change. This functionality is described in [\[MS-USRPCH\]](#) and [\[MS-UPSCHNG\]](#) for Office SharePoint Server 2007 and in [\[MS-UPSCHNG2\]](#) for SharePoint Server 2010.

The following figure shows a high-level view of the User Profile Service Web service architecture.

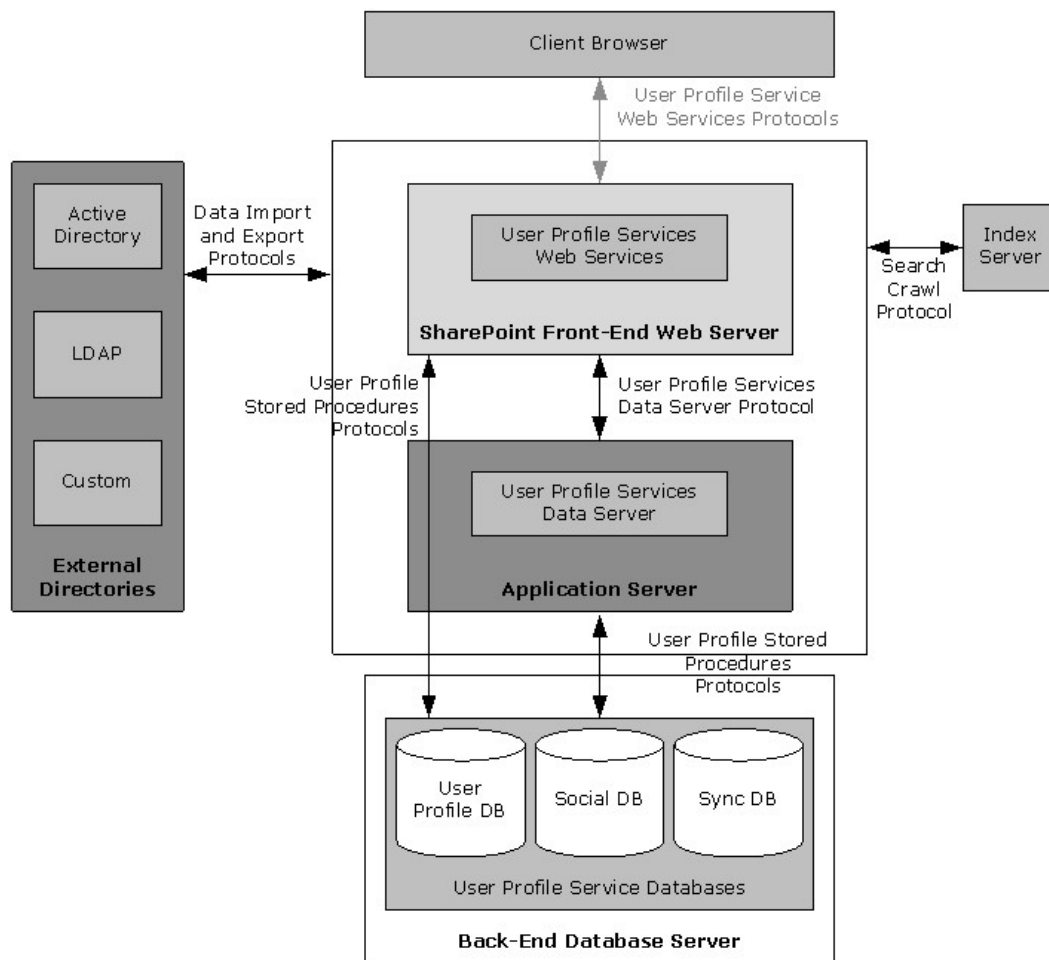


Figure 4: High-level view of the User Profile Service protocol architecture

2.1.3.4.1 Core Service

The core User Profile Service describes the set of protocols that enable an implementer to access and modify the data and metadata stored in the service. The User Profile Service Web service (described in [\[MS-USPSWS\]](#)) allows data operations such as creating, reading, updating, or deleting user profiles. In addition, user profiles can be searched by identifier such as name, e-mail address, or alias.

The User Profile Service databases store and manage user information.

The back-end protocols, as described in [\[MS-UPSPROF\]](#) for Microsoft® Office SharePoint® Server 2007 and [\[MS-UPSCHNG2\]](#) for Microsoft® SharePoint® Server 2010, provide the data for the front-end protocol, as described in [\[MS-USPSWS\]](#).

When accessing data from the user profile database, the use of the User Profile Service Web Service Protocol [\[MS-USPSWS\]](#) is recommended over the User Profile Stored Procedures Protocol [\[MS-UPSPROF\]](#) for Office SharePoint Server 2007 or the User Profile Change Log Stored Procedure Version 2 Protocol [\[MS-UPSCHNG2\]](#) for SharePoint Server 2010.

The key pieces of data stored in the User Profile Service database are the following:

- Properties
- Organizations
- Profiles
- Multiple values
- Vocabularies
- Colleagues
- Memberships
- Profile change log
- Social data
- Social activity

2.1.3.4.1.1 Properties

The User Profile Service database schema defines what the profile object contains. The database schema is a list of properties, their data types, and optionally, valid property values. System properties are used by SharePoint Products and Technologies and cannot be deleted. Non-system (normal or "regular") properties can be deleted. Administrators and implementers can create only non-system properties; however, there are no fixed limits on the number of properties that can be created for a profile.

2.1.3.4.1.2 Organizations

In Microsoft® SharePoint® Server 2010, organizations can be created to represent an entity, such as a business division or a working group containing members, delegates, contact information, and additional properties defined in the schema. Organizations can have a hierarchical relation with other organizations. This hierarchical relationship can then be used for actions such as searching for specific users related to the organization or its child organizations.

2.1.3.4.1.3 Profiles

A profile is a record for one user, and in some situations represents one identity (for example, a user account). The profile consists of a set of identifiers (such as user name, e-mail address, and alias) and a list of values for additional properties that are defined in the schema. In Microsoft® SharePoint® Server 2010, organizations can also have profiles, with identifiers such as the organization name, owner, delegates, and a list of values for additional properties that are defined in the schema.

2.1.3.4.1.4 Multiple Values

Many scenarios require more than one value for a given property, for example, the list of a user's skills. The User Profile Service Web service enables the storage of multiple, individually addressable property values. This functionality is typically used to **query** all profiles that contain a specific value for a property (for example, find all users who have a "Skill" called "programming").

2.1.3.4.1.5 Vocabularies

In Microsoft® Office SharePoint® Server 2007, the vocabulary feature of the profile store allows the administrator to designate a property to contain values that can be selected from a list of allowed values (a vocabulary). Vocabularies drive the user to use or select property values for a property that are consistent with another property's use of the same value. There are two types of vocabularies in the profile store: open and closed.

Open vocabularies allow the user to select an existing value, or if it does not exist, to enter a value specific to the user. This value is then made available for all future selections. For example, an **Interests** profile property might be an open vocabulary property that allows the user to select from interests that others have used to describe themselves or to enter a new interest, for example, "skiing". When subsequent users update their profile, the vocabulary contains the word "skiing".

For Microsoft® SharePoint® Server 2010, similar functionality is maintained, but the information is driven by term sets that are stored and managed by the term store defined in the Content Management Service (section [2.1.3.7](#)). This allows a term set administrator to define centrally controlled terms with the full benefit of other functionality implemented by the Content Management Service. Such centrally controlled term sets are closed vocabularies for users to use or select.

2.1.3.4.1.6 Colleagues

The profile service stores relationships between users, as well as metadata about users. The list of other users that the user knows is named "colleagues". The User Profile Stored Procedures Protocol [\[MS-UPSPROF2\]](#) not only returns the user list, but can also return a list of users that are common between any two users.

2.1.3.4.1.7 Memberships

The profile service also stores and models a user's affiliations with groups or organizations. The service models two types of groups: SharePoint **sites (2)** and **distribution lists**. By using the User Profile Synchronization Stored Procedures Protocol [\[MS-UPSSYNC\]](#) for Microsoft® Office SharePoint® Server 2007 or the User Profile Synchronization Stored Procedures Version 2 Protocol [\[MS-UPSSYNC2\]](#) for Microsoft® SharePoint® Server 2010, the profile store determines the list of sites that a user is a member of. For Office SharePoint Server 2007, the User Profile Import Protocol [\[MS-UPSIMP\]](#) can be used to create distribution lists and security group memberships in the profile store. The creation of distribution lists and security group memberships is accomplished by using the User Profile Import and Export Stored Procedures Protocol [\[MS-UPIESP\]](#) and User Profile Import/Export Web Service Protocol [\[MS-UPIEWS\]](#) in SharePoint Server 2010.

2.1.3.4.1.8 Profile Change Log

As with any database store, it is important to identify and access the information that has changed. The uses for this information are many. For example, when synchronizing two profile stores, the changes can be applied from one database to another.

The change log is a chronological list of changes, and each change has a time stamp. The change log service (as described in [\[MS-UPSCHNG\]](#) for Microsoft® Office SharePoint® Server 2007, and [\[MS-UPSCHNG2\]](#) for Microsoft® SharePoint® Server 2010) returns the changes that occurred in the database after a given time stamp or social activity. The User Profile Change Log Stored Procedure Protocol [\[MS-UPSCHNG\]](#) for Office SharePoint Server 2007 and the User Profile Change Log Stored Procedure Version 2 Protocol [\[MS-UPSCHNG2\]](#) for SharePoint Server 2010 are the back-end protocols that correspond to the capabilities provided by the front-end protocol User Profile Change Log Web Service Protocol [\[MS-USRPCH\]](#).

Any property or membership change triggers the creation of an entry in the change log.

2.1.3.4.1.9 Social Data

In Microsoft® SharePoint® Server 2010, the User Profile Service facilitates a user applying social data on a URI. A user can select, for example, a document or an arbitrary URL and attach additional social metadata to the URL. Some common scenarios for using metadata include the ability to rate, comment, bookmark, or add keywords to content.

2.1.3.4.2 Data Import and Export

The profile service described in section [2.1.3.4.1](#) is useful only when there is a large body of data on which applications and features are built. Enterprise systems already contain a significant amount of information about users, whether in enterprise **line-of-business (LOB) systems**, directories, or other databases. The profile service imports data from most available sources by using the protocols defined in [\[MS-UPSIMP\]](#) for Microsoft® Office SharePoint® Server 2007 and in [\[MS-UPIESP\]](#) and [\[MS-UPIEWS\]](#) for Microsoft® SharePoint® Server 2010. In Office SharePoint Server 2007, data from available sources was imported directly into the profile store. In SharePoint Server 2010, data is imported into a synchronization database and then into the profile store, by using the protocols described in [\[MS-UPSCDS\]](#), [\[MS-UPSCDAP\]](#), [\[MS-UPSDBDAP\]](#), [\[MS-UPSLDAP\]](#), [\[MS-UPSSCXPP\]](#), [\[MS-UPSMODS\]](#), and [\[MS-UPSCP\]](#).

After the data is imported and aggregated in the profile service, it can be exported to other areas where user data is required. The profile service uses the User Profile Synchronization Stored Procedures Protocol ([\[MS-UPSSYNC\]](#) for Office SharePoint Server 2007) and User Profile Synchronization Stored Procedures Version 2 Protocol ([\[MS-UPSSYNC2\]](#) for SharePoint Server 2010) to synchronize data from the profile store into various SharePoint Products and Technologies content databases.

The profile service can import data from three sources:

- An **Active Directory** directory
- A **Lightweight Directory Access Protocol (LDAP)** v3 directory
- The SharePoint Products and Technologies **Business Data Connectivity (BDC)** catalog (Business Data Catalog in Office SharePoint Server 2007)

Once the data is available in the user profile store, the profile service can export the data to the SharePoint Products and Technologies content database. In SharePoint Server 2010, the User Profile Service can also export data back to Active Directory and Lightweight Directory Access Protocol (LDAP) directories. The protocols used for import and export in SharePoint Server 2010 are defined in [\[MS-UPIESP\]](#) and [\[MS-UPIEWS\]](#). The SharePoint Products and Technologies BDC catalog remains import only.

2.1.3.4.2.1 Active Directory

Active Directory® Domain Services (AD DS) can provide a rich store of user information, as described in [\[MSFT-ADC\]](#). It contains identifying information, descriptive metadata (such as office number, title, and phone number), and relationship information (such as manager and groups a user belongs to). The profile service can import that information into the database. For each directory, the administrator creates a directory connection and describes which directory to connect to, what **credentials** to use, and what subset of the data to import.

In Microsoft® SharePoint® Server 2010, a User Profile Service administrator can define properties that will be exported back to AD DS. A popular scenario for the export functionality is to send a user's picture back to be stored in AD DS.

2.1.3.4.2.2 LDAP v3 Directory

SharePoint Products and Technologies also supports directories that implement the LDAP v3 standard described in [\[RFC4510\]](#). Importing from LDAP v3 works in much the same way as importing from Active Directory® Domain Services (AD DS). However, groups are not imported from an LDAP v3 directory.

In Microsoft® SharePoint® Server 2010, a User Profile Service administrator can define properties that will be exported back to LDAP. A popular scenario for the export functionality is to send a user's picture back to be stored in LDAP.

2.1.3.4.2.3 Business Data Connectivity (BDC) Catalog

SharePoint Products and Technologies supports importing user data from external systems that are configured by using the Business Data Catalog feature in Microsoft® Office SharePoint® Server 2007, called Business Data Connectivity (BDC) in Microsoft® SharePoint® Server 2010. The Business Data Connectivity service is part of **Business Connectivity Services (BCS)**. For information about Business Connectivity Services, see [\[MS-SPFEPO\]](#) section 2.1.3.6.

This action does not specify any particular wire protocol, because BDC connects to whatever protocol is provided by the back-end system. The BDC information provides a consistent interface to Web services and databases through a declarative XML syntax. The User Profile Service uses BDC to import data from the data sources that BDC supports. For example, the human resources (HR) database contains the employee's hire date, which is not published to the company directory. The profile service can create a profile for the employee from the directory entry and complete the employee's **hiredate** property by importing that value from the HR database.

One major limitation to importing from BDC is that profiles cannot be created or removed by an import. The BDC import only annotates existing profile records.

2.1.3.4.3 SharePoint Products and Technologies Content Database

SharePoint Products and Technologies stores user information in every site collection for various uses such as security, name display, and presence. The user record in each site collection is created by copying information from the directory; however, the record is not updated automatically. The User Profile Synchronization Stored Procedures Protocol [\[MS-UPSSYNC\]](#) for Microsoft® Office SharePoint® Server 2007 and the User Profile Synchronization Stored Procedures Version 2 Protocol [\[MS-UPSSYNC2\]](#) for Microsoft® SharePoint® Server 2010 update the records on a schedule that the administrator sets.

The protocol server tracks user information that has changed in the User Profile Service database since the last time it synchronized data, as described in [\[MS-UPSCHNG\]](#) for Office SharePoint Server 2007 and [\[MS-UPSCHNG2\]](#) for SharePoint Server 2010. The protocol client scans every content database for new user records and updates them with the latest information from the User Profile Service database. It also locates the records for users whose information has changed in the User Profile Service database and updates those as well.

2.1.3.4.4 User Profile Service Database Indexing

SharePoint Products and Technologies implements full-text search of information stored in the user profile database. For example, users can find the profiles of other users by searching on interesting

keywords related to expertise, name, or any other information stored in the User Profile Service. This capability is implemented by using the search infrastructure that is described in the various protocol specifications for the **search service application**, which requires that the User Profile Service database is first indexed. This indexing is done using the protocols defined in [\[MS-SPSCRWL\]](#) and [\[MS-SPSCLSP\]](#) for Microsoft® Office SharePoint® Server 2007, and [\[MS-SPSCRWL\]](#) and [\[MS-SPSCLSP2\]](#) for Microsoft® SharePoint® Server 2010.

2.1.3.4.5 Personalization and Targeting

It is possible to use the information in the User Profile Service database for personalization and content targeting. For example, a company's employee portal site can be customized to show something specific to the job role of the employee to people in different job areas of the company. For example, rather than showing the same content to everyone, a person who works in "Sales" will see something different about a given employee than what someone who works in "Development" might see.

SharePoint Products and Technologies addresses the challenge of partitioning people into specific groups through the use of **audiences**, as described in [\[MS-UPSAUD\]](#) for Microsoft® Office SharePoint® Server 2007 and [\[MS-UPSAUD2\]](#) for Microsoft® SharePoint® Server 2010. Audiences define groups of people by their characteristics. An audience can be referred to as a dynamic group that is implemented through the use of rules, which are periodically evaluated in batch through a process called audience compilation.

2.1.3.4.6 Social Metadata

In Microsoft® SharePoint® Server 2010, the User Profile Service facilitates a user applying social data on a URI.

A user can select, for example, a document or an arbitrary URL and attach additional social metadata to the URL.

Some common scenarios for using metadata include the ability to:

- "Rate" a document, content, or person.
- Add a note or comment to content.
- Bookmark content with some predefined keywords, such as "a favorite".
- Add tags to the content to categorize the content or highlight it with colleagues.

The Web services described in [\[MS-UPSDWS\]](#) use the User Profile Social Data Stored Procedures Protocol [\[MS-UPSSDSP\]](#) to attach social metadata to the corresponding URI. In the case of bookmarks and tags, the User Profile Service works with the Content Management Service (section [2.1.3.7](#)) to store the value of the tag; the User Profile Service database only stores the corresponding **GUID**.

For all forms of metadata, four key pieces of information are stored in the User Profile Service database:

1. Who added the metadata: the record identifier of the user in the profile store.
2. When was it added: a time stamp.
3. Where it was applied: the URI.

4. Which social metadata it is: for rating, it is a number ranging from 1-5; for note, it is the text of the note; and for tags and bookmarks, it is the GUID returned from the Content Management Service.

2.1.3.5 Search Services

The following is a description of the architecture and components of the subsystems in the search functionality in the SharePoint Products and Technologies search services, along with the protocols involved in interoperability between the components.

For a system overview and listing of Microsoft® FAST™ Search Server 2010 for SharePoint® and Microsoft® FAST™ Search Server 2010 for SharePoint® Internet Sites protocols see [\[MS-FSQ\]](#).

2.1.3.5.1 Concepts

Search services use two basic data structures to resolve query requests from the protocol clients: inverted indexes and the property store. These data structures are populated during the crawl process, and are run on the index server in Microsoft® Office SharePoint® Server 2007 and Windows® SharePoint® Services 3.0 and on one or more servers that have a crawl component in Microsoft® SharePoint® Server 2010 and Microsoft® SharePoint® Foundation 2010.

2.1.3.5.2 Inverted Indexes

An inverted index is a data structure that provides mapping between **tokens** to the list of indexed documents that contain these tokens. The list of documents that are indexed is a **list of postings**. Tokens include ordinary language words as well as numbers and alphanumeric sequences.

The list of postings for a given token is composed of internal references to each indexed document that contains that token. These references are called **document identifiers (3)**. Document identifiers are assigned by the index server in Microsoft® Office SharePoint® Server 2007 and Windows® SharePoint® Services 3.0 and by one or more servers that have a crawl component in Microsoft® SharePoint® Server 2010 and Microsoft® SharePoint® Foundation 2010 during crawls. The list of postings is clustered in contiguous pages on disk or in memory to allow the most efficient retrieval of the set of document identifiers by the keyword. The list of postings is also sorted to allow efficient merges between multiple lists if the user issues a multiword query. The list of postings might also be compressed by using delta compression, where the list will actually contain the differences in document identifier values between the current document identifier and the previous document identifier. For information about the inverted index format, see [\[MS-CIFO\]](#).

For each token, the inverted index also contains a list of occurrences (the actual positions) of the corresponding token within each document that is indexed. A list of occurrences can be used to determine if a particular document contains a phrase of two or more tokens by comparing their corresponding positions. The total number of occurrences can be used for relevance ranking, known as **term frequency (TF)**. The total number of postings in the list for a particular token is known as Document Frequency (DF). Both TF and DF can be used to compute the relevance ranking score of a document during query execution.

When documents are indexed, not only is the textual content (the full set of tokens) extracted, but the search system also stores the field (or property) in which the text was found. A collection of documents can contain millions of uniquely named properties, so the search system can mark a subset of interesting or significant properties as managed properties. Examples of such managed properties include **Title**, **URL**, **Body**, and **Author**.

Each list of postings and list of occurrences is associated with a token and with a key. Keys contain two fields: the actual token and a property identifier. Each property identifier corresponds to a

unique managed property. This allows execution of queries that require locating documents that contain a specific keyword in a specific property, for example, "title", or "body", or "author".

Finally, the inverted index contains some information about the computed relevance of each document. It has a data structure that stores the static rank for each document identifier. The static rank is the query-independent relevance rank that enables the index server to return a set of document identifiers that match a query and a set sorted by relevance.

A fundamental behavior of the inverted index is that it can only retrieve document identifiers.

2.1.3.5.3 Property Store

A property store is logically a tabular data structure, plus an associated set of operations. It is stored in the back-end database. It is also known as the **metadata index** in Microsoft® SharePoint® Server 2010 and Microsoft® SharePoint® Foundation 2010.

The property store has one column that contains the document identifier, and arbitrary sets of other columns that contain useful document properties. Because the inverted index (section [2.1.3.5.2](#)) can only retrieve document identifiers, the property store is then used to find property values related to the indexed document. Typically, the property store would contain at least the document identifier, Uniform Resource Locator (URL) of the document that is indexed, author, size, and the data needed to generate contextual summaries to show with the search results. Logically, the document identifier (3) is the unique identifier of the document records in the property store.

The property store is also used to support property-related queries, such as range-queries (for example, all documents with a size of 0-100). The inverted index cannot support such queries for a random data type.

The property store provides quick mapping from document identifiers to document properties, as opposed to the inverted index, which provides quick mapping from keywords to document identifiers.

2.1.3.5.4 Crawling

The crawl operation is a background process that involves traversal of the URL space configured by the user. The crawl operation is implemented by the index server in Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® Foundation 2010 and one or more servers that have a crawl component in Microsoft® SharePoint® Server 2010 and SharePoint Foundation 2010. If the URL space is considered a graph, where each node is a URL and the edges are created from **hyperlinks** (in the case of HTML) or other types of links between URLs, the basic crawl operation procedure consists of an exhaustive graph traversal.

The crawl operation maintains a crawl queue and crawl history, both of which are stored in the back-end database. The crawl history contains all URLs visited by the index server during the crawl. The crawl queue contains the list of URLs that still need to be processed during the crawl operations.

An administrator configures the crawl operation, and specifies both the location from which the crawl starts (the start address) plus the methodology of crawling. The methodology is one of the following:

- Recursively follow hyperlinks discovered within documents.
- Follow a hierarchy within a repository by using the links provided by the repository itself.

Additionally, an administrator might specify **crawl rules** to prevent the indexing of documents within a specified location.

The crawl operation starts by adding the start addresses to the crawl queue. The index server picks URLs from the queue to process. Processing of a URL involves retrieving the document data from the document repository specified by the URL, parsing the document, and determining the set of links to follow. To avoid infinite cycles in the traversal, each retrieved hyperlink is checked against the crawl history to determine if the document was already visited during this crawl. Each link is also checked against the crawl rules. If the rule determines that the URL of the link needs to be excluded from the crawl operation, the link is discarded; otherwise, it is added to the crawl queue. The crawl operation completes when the crawl queue is empty.

Subsequent crawl operations do not have to reindex every document in the URL space. These incremental crawls can be of two types: repositories that can provide detailed information about the list of changes that need to be indexed and repositories that cannot provide this detailed information. SharePoint Products and Technologies includes the former. This type of repository monitors which documents within the repository change, and thus can provide change logs. A change log is the list of URLs of changed and deleted documents to reindex. For all other repository types, the set of URLs in the crawl history is added to the crawl queue and a crawl operation is initiated. By comparing the time stamp of each document that is crawled to its time stamp in the crawl history, the system can selectively reindex only those documents that have changed or have been deleted.

2.1.3.5.5 Search Topology

The following two figures provide a view of the topology of various computer roles that are involved in providing search functionality.

The index server in Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® Foundation 2010, and the one or more servers that have a crawl component in Microsoft® SharePoint® Server 2010 and SharePoint Foundation 2010, crawl the content of the documents in the URL space according to an administrator's configuration and update the inverted indexes and the property stores with the updated content. The index server uses the back-end database server to store the crawl state, and it stores the document properties in the property stores, which are also located on the back-end database server. When new or updated inverted indexes are created, the index server propagates them (whole or in incremental fragments) to each query server in the farm.

In Office SharePoint Server 2007, the index server also performs a user profile import from a **directory service (DS)**. The profile import is a special type of crawl that enumerates the URL space of the DS. However, rather than producing the inverted indexes, it uses the User Profile Import Protocol [\[MS-UPSIMP\]](#) to populate the user profile records in the back-end database server. The index server imports user records from either Active Directory® Domain Services (AD DS) ([\[MS-ADTS\]](#)) or from any DS supporting LDAP [\[RFC4510\]](#).

One or more query servers host the inverted index as it is propagated from the index server, following (or during) crawls. These query servers use the inverted index to return a relevance-sorted list of document identifiers in response to a query from a front-end Web server.

The back back-end database stores several components: the property stores, the crawl state, the crawl queues, the security descriptors, and the anchor tables (which contain the URL space annotated with hyperlinked text, used for the relevance-ranking computation). It also stores the search application configuration data.

In Office SharePoint Server 2007 and SharePoint Foundation 2010, the role of the front-end Web server is to receive the user's search query from the client browser or application. It parses the query and breaks it into those portions that can be resolved by information in the inverted indexes and those portions that require the property store. The query is sent to the query server and sometimes to the back-end database. The returned document identifiers, along with the property that was matched in each document, are typically combined with information from the anchor table

to sort the matching documents by relevance. Metadata is then returned for each matching document from the property store and security descriptor table on the back-end database. The documents are security trimmed, and any duplicate documents are removed. Finally, the search results are returned to the user.

In SharePoint Server 2010, the role of the front-end Web server is to receive the user's search query from the client browser or application and send it to the server that hosts the search manager Web service. This server then parses the query and breaks it into those portions that can be resolved by information in the inverted indexes and those portions that require the property store. The query is sent to the query server and sometimes to the back-end databases. The returned document identifiers, along with the property that was matched in each document, are typically combined with information from the anchor table to sort the matching documents by relevance. Metadata is then returned for each matching document from the property store and security descriptor table on the back-end database. The documents are security trimmed, and any duplicate documents are removed. Finally, the search results are returned to the user.

The following figure shows the topology of protocols implemented by search services in Office SharePoint Server 2007.

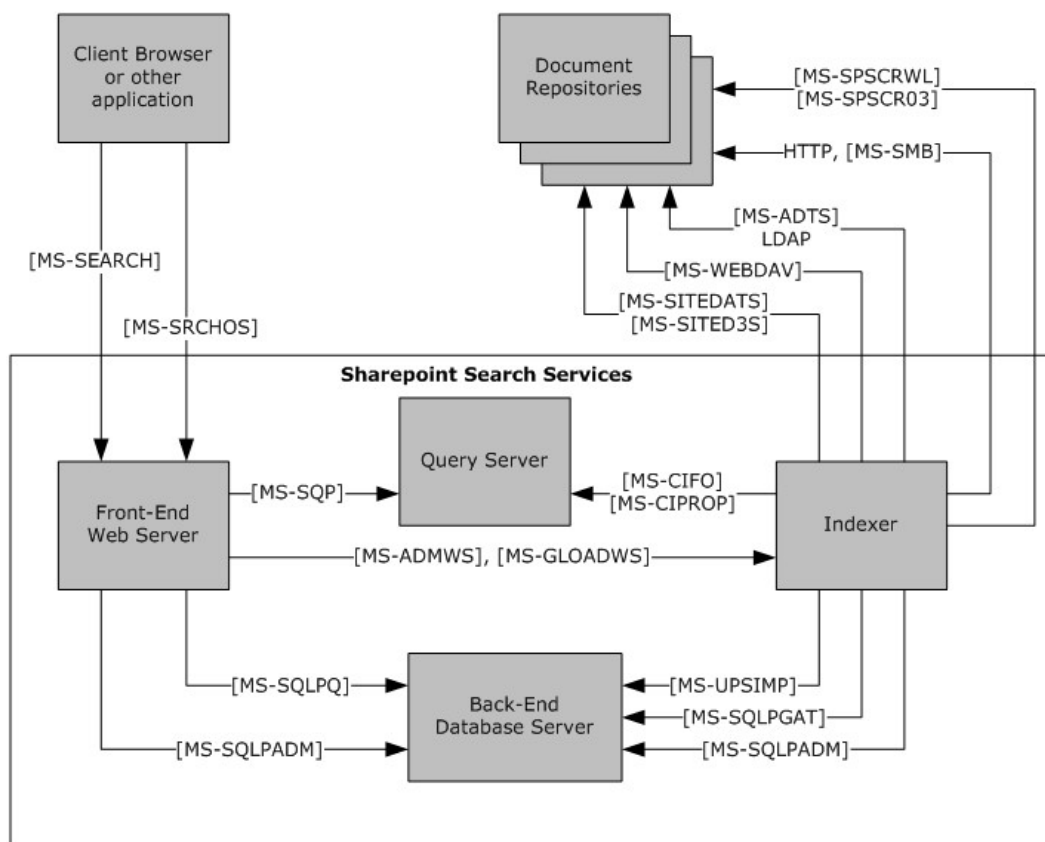


Figure 5: Search services protocol topology for Office SharePoint Server 2007

The following figure shows the topology of protocols implemented by search services in SharePoint Foundation 2010 and SharePoint Server 2010.

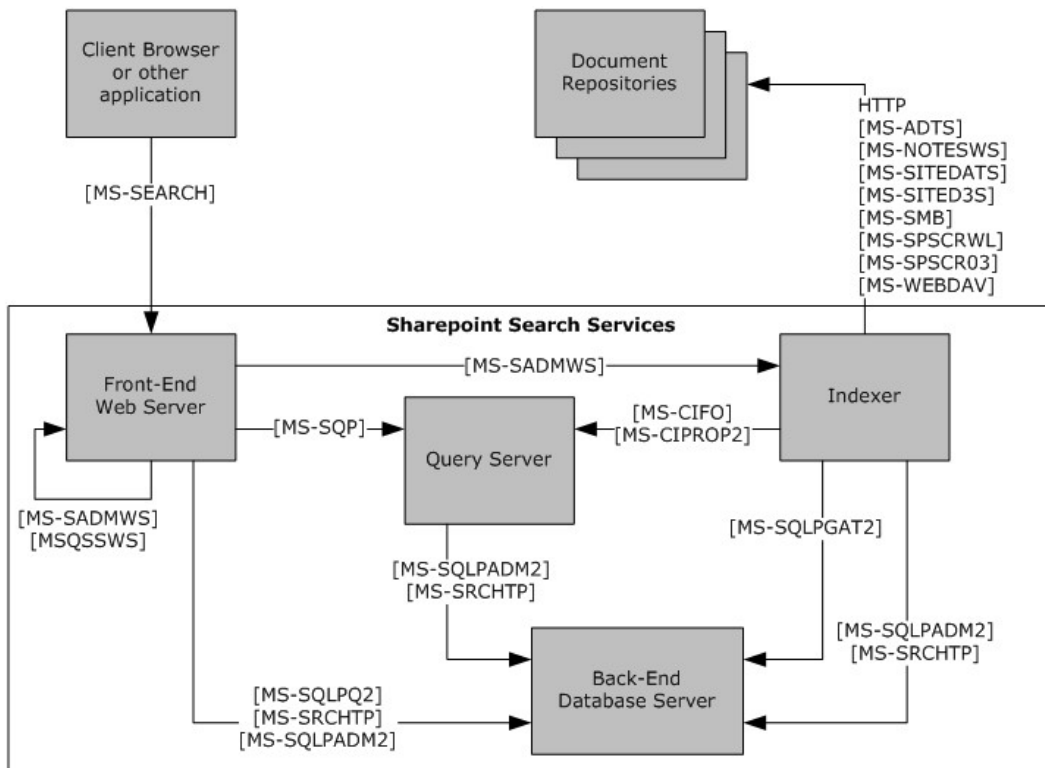


Figure 6: Search services protocol topology for SharePoint Foundation 2010 and SharePoint Server 2010

2.1.3.6 Business Connectivity Services

SharePoint Products and Technologies implements capabilities to access data from external systems, as described in [\[MS-SPFEPO\]](#) section 2.2.2.6. Business Connectivity Services is a shared service and stores its configuration information in the SharePoint Products and Technologies configuration database. In addition, SharePoint Products and Technologies implements a **Secure Store Service (SSS)** that securely stores user credentials in a separate relational database. The protocols described in section 2.2.2.6 in this document implement communication between a front-end Web server and a back-end database server for accessing data from external systems. Both protocols are implemented on TDS ([\[MS-TDS\]](#)) as a transport protocol.

2.1.3.7 Content Management Service

As described in [\[MS-SPFEPO\]](#) section 2.1.3.7, content stored in SharePoint Products and Technologies is accessible from a back-end database server by using the list and document library storage protocols. These protocols all provide communication between a front-end Web server and a back-end database server on top of TDS ([\[MS-TDS\]](#)) as a transport protocol.

In addition, SharePoint Products and Technologies implements two specific back-end protocols for deployment and transformation of content. Section 2.2.2.7 introduces the Content Deployment Remote Import Web Service Protocol ([\[MS-CDEPLOY\]](#)). This protocol uses the PRIME format (as described in [\[MS-PRIMEPF\]](#)) as the serialized format for importing content to another SharePoint site. Then, this site can be hosted on the same farm or on a remote farm.

Also, SharePoint Products and Technologies implements capabilities that enable clients to create, delete, merge, reuse, deprecate, describe, translate, and define synonyms for metadata terms to be applied to digital assets and resources, and to build hierarchical structures of terms (or terms sets). Section [2.2.2.7](#) introduces the Enterprise Metadata Service Database Schema Specification (as described in [\[MS-EMMSTORE\]](#)), which describes these capabilities.

2.1.3.8 PerformancePoint Services

PerformancePoint Services in Microsoft® SharePoint® Server 2010 is a collection of services in Microsoft® SharePoint® Server 2010. PerformancePoint Services 2010 uses two content stores to manage dashboard objects and the **data connections (1)** between those objects.

The first content store is a SharePoint Products and Technologies list or document library. Because PerformancePoint Services 2010 objects are stored in standard, content-type-specific lists, they can be accessed easily from the SharePoint Products and Technologies content database. These lists store serialized XML data and metadata describing data sources, KPIs, scorecards, grids, charts, report views, indicators, filters, and dashboards. User access to these objects is controlled by the list protocol.

In addition to the use of lists and document libraries, PerformancePoint Services 2010 uses a separate database to store scorecard annotations, user-selected filters, and temporary state data structures passed between the front-end Web server and the back-end database server. This server-to-server protocol uses the PerformancePoint Services Application Server Protocol [\[MS-PPSAPP\]](#) as its transport between the front-end Web server, the middle-tier application server, and the back-end database server.

2.1.3.9 Word Automation Services

The functionality and protocols described in this section are new in Microsoft® SharePoint® Server 2010.

Word Automation Services provides the capability to convert large numbers of input files in groups, each group using a specific set of conversion preferences (for example, embed **fonts** in the output file). See [\[MS-SPFEPO\]](#) section 2.2.2.7 for more information about the conversion service that is not related to Word Automation Services.

Word Automation Services is composed of two components: the Word Automation Services WCF Service Protocol [\[MS-WORDSWCF\]](#) and the Word Automation Services Stored Procedures Protocol [\[MS-WORDSSP\]](#) (and the underlying database).

The following diagram illustrates the relationship between these two protocols.

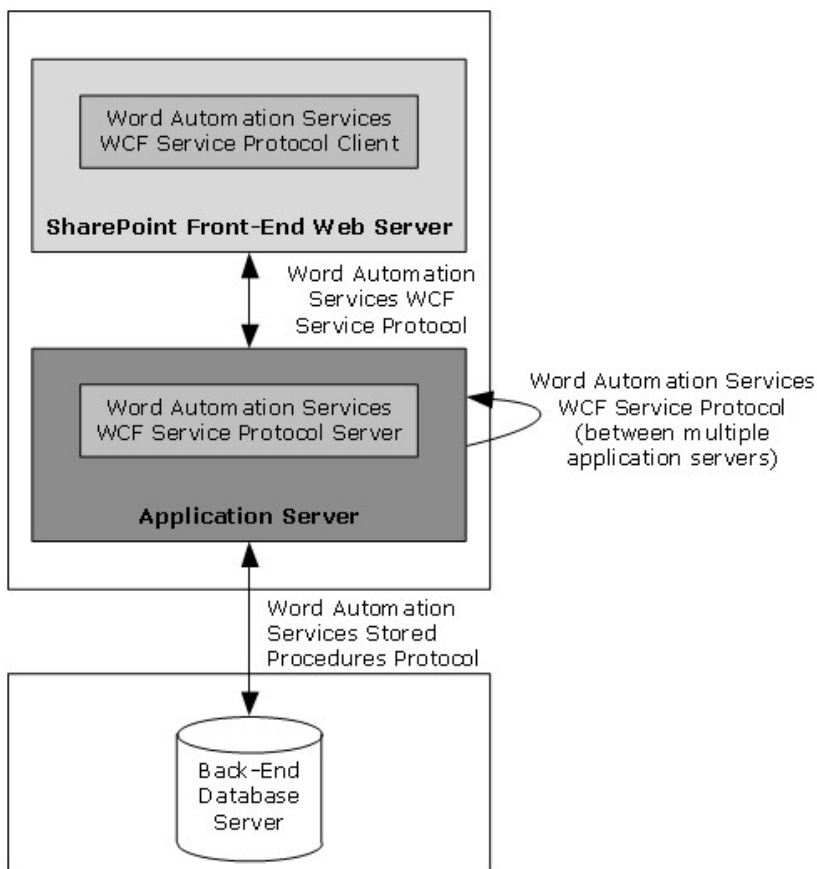


Figure 7: Relationship between the two components of Word Automation Services

The Word Automation Services WCF Service Protocol is used to convert documents from one file format to another file format, as described in [MS-WORDSWCF]. This protocol allows a protocol client to create and manage a persistent queue of document conversions. It also allows the protocol client to initiate the conversion of documents in the queue from one file format to another file format. The protocol server maintains the queue and initiates conversions based on requests from a protocol client. The protocol server reads from and writes to the queue database by using the database protocol, as described in [MS-WORDSSP].

2.1.3.10 Session State

SharePoint Products and Technologies does not ensure that a **user session** will reconnect to the same front-end Web server across multiple requests. It is sometimes necessary to store this user session data in a single, central location and to provide ways for front-end Web servers to easily locate, access, and change the data. For those SharePoint Products and Technologies applications that require this functionality, they rely on either ASP.NET session state available in Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® Server 2010, or the State Service in SharePoint Server 2010 to handle state data.

2.1.3.10.1 ASP.NET Session State

The SharePoint Products and Technologies front-end Web servers use the ASP.NET State Service Database Repository Communications Protocol [\[MS-ASPSS\]](#) when the server loads a page on which ASP.NET session state is enabled, and when ASP.NET session state is enabled on the farm.

SharePoint Products and Technologies stores ASP.NET session state data on a **Structured Query Language (SQL)** back-end database server. In Microsoft® Office SharePoint® Server 2007, this database is the Shared Services Provider (SSP) database. In Microsoft® SharePoint® Server 2010, this database is a stand-alone database.

Because it relies on ASP.NET session state, the SharePoint Products and Technologies front-end Web server implementation identifies ASP.NET session state as enabled on a page only if:

- The page has the **enableSessionState** page property set to **true**, as described in [\[MSDN-Page\]](#).
- The page implements either of the following interfaces: **IReadOnlySessionState**, as described in [\[MSDN-IReadOnlySessionState\]](#), or **IRequiresSessionState**, as described in [\[MSDN-IRequiresSessionState\]](#).

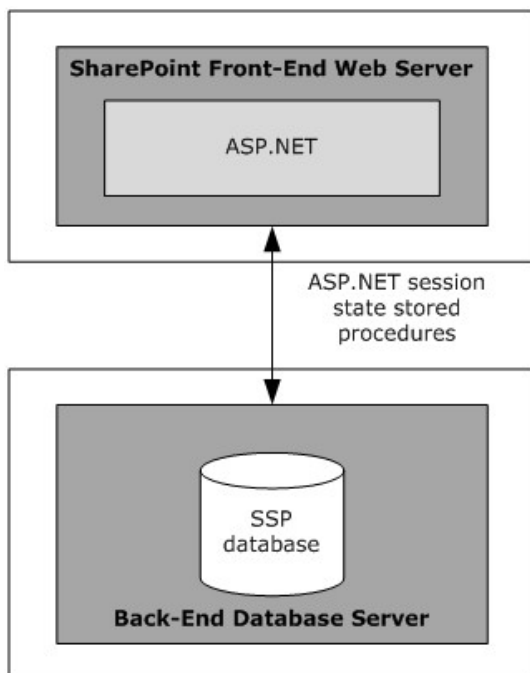


Figure 8: ASP.NET session state configuration and protocols in Office SharePoint Server 2007

2.1.3.10.2 State Service

Microsoft® SharePoint® Server 2010 front-end Web servers use the Temporary State Service Protocol [\[MS-SPSTATE\]](#) when the server loads a page that uses the State Service. SharePoint Server 2010 stores State Service data in a SQL back-end database server as a stand-alone database.

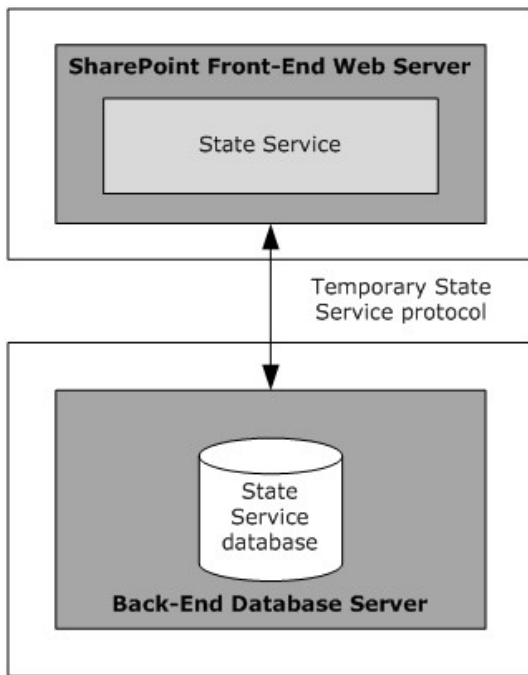


Figure 9: State Service configuration and protocols

2.1.3.11 User Code Execution Service

Microsoft® SharePoint® Server 2010 implements a capability to securely host custom solutions and executable code that comes from customers of the system. It provides better controls for the security of execution of custom code, as well as mechanisms to monitor and control the usage of resources by that code.

One requirement of these mechanisms is to move the execution of the code to separate processes and separate servers, to provide better layers of isolation and monitoring. Therefore, two roles are defined in a user-code execution process: a front-end Web server that initiates requests to run user code on behalf of a user, and a code execution back-end service that fulfills the execution request. Communication between these servers is described in the User Code Execution Protocol [\[MS-SPPTC\]](#).

2.1.3.12 Analytics Services

Analytics Services allows users to access three categories of analytical data about how the farm is being used. For information about the data that Analytics Services collects see [\[MS-SPFEPO\]](#) section 2.1.3.9. This data is stored in the Analytics Services service application database, and [\[MS-MAVA\]](#) describes how Analytics Services retrieves data from that database.

2.1.3.13 Visio Graphics Service

The Visio Graphics Service enables the server to render a drawing page in a Web drawing as a raster or vector drawing. The rendered drawing can then be viewed in a Web browser.

The typical scenario for using these services is to create a Web drawing by using a client such as Microsoft® Visio® 2010, storing the drawing on the server, and then viewing it in a Web browser. For example, a user can create a Web drawing that shows a workflow process. When the workflow is

executed on the server, a user can view the drawing of the workflow in a Web browser. Workflow status can then be overlaid on the diagram by annotating it in the Web browser (by using JavaScript/DHTML if the diagram is rendered as raster or by using XAML if it is rendered as vector).

2.2 Protocol Summary

The tables in this section provide a comprehensive list of the member protocols of the SharePoint Products and Technologies Back-End Protocols system. The member protocols are grouped according to their primary purpose.

2.2.1 Core Platform Protocols

This section lists the back-end protocols that are used to interoperate with the core platform components of SharePoint Products and Technologies.

2.2.1.1 Configuration

The protocol in the following table enables configuration of the Shared Services Provider (SSP) in Microsoft® SharePoint 2007 Products and Technologies and service applications in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
SSP Scheduled Jobs Stored Procedures Protocol Specification	Allows protocol clients to add, modify, refresh, and delete scheduled jobs from the SSP, as well as retrieve those scheduled jobs by using predefined criteria such as a unique identifier or jobs scheduled to be run within a specific time interval. In addition, the protocol specifies the actions that the protocol client takes when a particular scheduled job is retrieved and is to be run in the next specified time interval. This is used in both Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® Server 2010.	[MS-SSPS]

The protocols in the following table enable the retrieval of configuration information in SharePoint 2007 Products and Technologies only.

Protocol name	Description	Short name
SQL Configuration Object Stored Procedures Protocol Specification	Allows a protocol client to store and retrieve configuration objects on a protocol server and to maintain a cache of configuration objects that have changed on the protocol server, even if the changes were not made by the protocol client. This is used in Office SharePoint Server 2007 only.	[MS-SSPSOS]
Windows SharePoint Services: Configuration Database Communications Protocol Specification	Identifies the communications between a computer or set of computers running one or more services and a back-end database server on which the configuration data for the services is stored. The clients of this protocol are computers running services. The protocol server is a device that contains the configuration data for what will be the configuration database. This is used in Windows® SharePoint® Services 3.0.	[MS-WSSCFGD]

The protocols in the following table enable the retrieval, sharing, and storage of configuration data in SharePoint 2010 Products only.

Protocol name	Description	Short name
Windows SharePoint Services: Configuration Database Communications Version 2 Protocol Specification	Identifies the communications between a computer or set of computers running one or more services and a back-end database server on which the configuration data for the services is stored. The clients of this protocol are computers running services. The protocol server is a device that contains the configuration data for what will be the configuration database. This is used in Microsoft® SharePoint® Foundation 2010.	[MS-WSSCFGD2]
SharePoint Subscription Settings Web Service Protocol Specification	In enterprise and hosting scenarios, a number of site collections need to use a shared set of configuration data. This allows for the potential of some configuration data for this group of site collections to be delegated to additional administrators. To facilitate these scenarios, this protocol has the ability to set, remove, and read property sets for a particular set of site collections. Applies to SharePoint Foundation 2010.	[MS-SPSETWS]
Service Platform Topology Web Service Protocol Specification	Identifies the communication between the protocol client and the protocol server that is used to obtain a list of service applications and their properties. This protocol enables the protocol client to discover the list of service applications known to the protocol server. It also enables the protocol client to retrieve the list of endpoint URIs for a service application so that it can communicate with it using a protocol supported by that service application. Applies to SharePoint Foundation 2010.	[MS-SPTWS]
SharePoint Subscription Settings Stored Procedures Protocol Specification	In enterprise and hosting scenarios a number of site collections need to use a shared set of configuration data. This protocol allows for the potential of some configuration data for this group of site collections to be delegated to additional administrators. To facilitate these scenarios, this protocol has the ability to set, remove, and read property sets for a particular set of site collections. Applies to SharePoint Foundation 2010.	[MS-SPSETSP]
SharePoint Usage Tracking Stored Procedures Protocol Specification	Supports the storage, retrieval, and reporting of usage and diagnostic data. Used by the protocol client to store usage data of various kinds, including user request information, performance counters, data on slow or expensive queries, and other relevant performance data. Applies to SharePoint Foundation 2010.	[MS-UTSP]

2.2.1.2 Content Storage

The protocols in the following table enable stored procedures and communications for content storage in Microsoft® SharePoint 2007 Products and Technologies only.

Protocol name	Description	Short name
Windows SharePoint Services Content Database Administrative Communications Protocol Specification	Identifies the communication between the front-end Web server and the back-end database server. This communication is used to satisfy requests involving management and administration of content databases within SharePoint 2007 Products and Technologies.	[MS-WSSCADM]

Protocol name	Description	Short name
Windows SharePoint Services: Content Database Document and List Item Management Communications Protocol Specification	Identifies the communication sequences used by front-end Web servers and application servers to perform data query and update commands on a back-end database server as part of theme , change log, list (1) , document , and list item operations.	[MS-WSSDLIM]
Windows SharePoint Services: File Operations Database Communications Protocol Specification	Identifies the communication between the front-end Web server and the back-end database server. This communication is used to satisfy requests involving file access and administration of users and groups within SharePoint 2007 Products and Technologies.	[MS-WSSFO]
Windows SharePoint Services: Content Database Core List Schema and Site Provisioning Communications Protocol Specification	Identifies the communication sequences used by front-end Web servers and application servers to perform data query and update commands on back-end database servers as part of content type, feature, file-handling, view, provisioning, list schema, and list or Web meta-information administration operations.	[MS-WSSCCSP]
Windows SharePoint Services: Content Database End-User Experience Communications Protocol Specification	Identifies the communication between the front-end Web server and the back-end content database used in user interaction with the server. The content database stores the data associated with the lists and sites. The client sends a request for a certain operation or for data to be sent to the front-end Web server. Then, the front-end Web server communicates with the content database to perform this action.	[MS-WSSEUX]
SQL Usage Analytics Protocol Specification	Identifies stored procedures that collect and store data based on Web site activity.	[MS-MOSSUAN]
Windows SharePoint Services: Content Database Programmability Extensions Communications Protocol Specification	Identifies the communication between the front-end Web server and the back-end-database server used to satisfy requests involving events (2) , Web Parts , and workflow (2) within SharePoint 2007 Products and Technologies.	[MS-WSSPROG]
Windows SharePoint Services: Content Database Data Migration Communications Protocol Specification	Enables a front-end Web server to communicate with back-end database servers to perform data migration. The protocol allows a front-end Web server to retrieve data objects from the back-end database servers or to store data objects on back-end database servers. The protocol defines specific APIs that are each responsible for retrieving one type of data object. The application is required to use certain sequence orders to correctly retrieve the data objects. For example, after the data objects are retrieved from the back-end database servers, the application can serialize them into a file format such as the deployment package.	[MS-WSSDM]

The protocols in the following table enable communications for content storage in Microsoft® SharePoint® 2010 Products and Technologies only.

Protocol name	Description	Short name
Windows SharePoint Services	Identifies the communication between the front-end Web	[MS-

Protocol name	Description	Short name
Content Database Administrative Communications Version 2 Protocol Specification	server and the back-end database server. This communication is used to satisfy requests involving management and administration of content databases within SharePoint 2010 Products.	WSSCADM2]
Windows SharePoint Services: Content Database Document and List Item Management Communications Version 2 Protocol Specification	Identifies the communication sequences used by front-end Web servers and application servers to perform data query and update commands on a back-end database server as part of theme, change log, list (1), document, and list item operations.	[MS-WSSDLIM2]
Windows SharePoint Services: File Operations Database Communications Version 2 Protocol Specification	Identifies the communication between the front-end Web server and the back-end database server. This communication is used to satisfy requests involving file access and administration of users and groups within SharePoint 2010 Products.	[MS-WSSFO2]
Windows SharePoint Services: Content Database Core List Schema and Site Provisioning Communications Version 2 Protocol Specification	Identifies the communication sequences used by front-end Web servers and application servers to perform data query and update commands on back-end database servers as part of content type, feature, file-handling, view, provisioning, list schema, and list or Web meta-information administration operations.	[MS-WSSCCSP2]
Windows SharePoint Services: Content Database End-User Experience Communications Version 2 Protocol Specification	Identifies the communication between the front-end Web server and the back-end content database used in user interaction with the server. The content database stores the data associated with the lists and sites. The protocol client sends a request for a certain operation or for data to the front-end Web server. Then, the front-end Web server communicates with the content database to perform this action.	[MS-WSSSEUX2]
Windows SharePoint Services: Content Database Programmability Extensions Communications Version 2 Protocol Specification	Identifies the communication between the front-end Web server and the back-end-database server that is used to satisfy requests involving events (2), Web Parts, and workflow (2) within SharePoint 2010 Products.	[MS-WSSPROG2]
Windows SharePoint Services: Content Database Data Migration Communications Version 2 Protocol Specification	Enables a front-end Web server to communicate with back-end database servers to perform data migration. The protocol allows a front-end Web server to retrieve data objects from the back-end database servers or to store data objects on back-end database servers. The protocol defines specific APIs that are each responsible for retrieving one type of data object. The application is required to use certain sequence orders to correctly retrieve the data objects. For example, after the data objects are retrieved from the back-end database servers, the application can serialize them into a file format such as the deployment package.	[MS-WSSDM2]

2.2.2 Extended Capabilities Protocols

This section lists the back-end protocols that are used to interoperate with the extended capabilities components of SharePoint Products and Technologies.

2.2.2.1 Excel Services

The protocol in the following table enables access to Excel® Services workbooks in both Microsoft® SharePoint 2007 Products and Technologies and Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Excel Calculation Web Service XML Specification	Identifies the structure used to send a workbook from the back-end server to the front-end server; this is an XML format as described in [XMLSCHEMA] .	[MS-EXSPXML]

The protocol in the following table enables access to Excel Services workbooks in SharePoint 2007 Products and Technologies only.

Protocol name	Description	Short name
Excel Calculation Web Service Protocol Specification	Accesses workbooks stored on a front-end Web server or in a location that can be reached by using a UNC path. For an overview about the specific implementation of Excel Services that implements this protocol, see [MSDN-ESO] . This is implemented in Microsoft® Office SharePoint® Server 2007.	[MS-EXSPWS]

The protocol in the following table enables access to Excel Services workbooks in SharePoint 2010 Products only.

Protocol name	Description	Short name
Excel Calculation Web Service Version 2 Protocol Specification	Accesses, enables editing, and saves workbooks stored on a front-end Web server, or accesses workbooks that can be reached by using a UNC path. For an overview about the specific implementation of Excel Services which implements this protocol, see [MSDN-ESO] . This is implemented in Microsoft® SharePoint® Server 2010.	[MS-EXSPWS2]

2.2.2.2 Access Services

The protocol in the following table enables Microsoft® Access Services in Microsoft® SharePoint® 2010 Products and Technologies only.

Protocol name	Description	Short name
Access Services Data Server Protocol Specification	Identifies how the protocol server manages data on a SharePoint Products and Technologies server, caches data on the protocol server, and provides access to the protocol client. Implemented in Microsoft® SharePoint® Server 2010.	[MS-ADS]

2.2.2.3 InfoPath Forms Services

InfoPath Forms Services Protocols are covered in [\[MS-SPFEPO\]](#) section 2.2.2.3.

In Microsoft® Office SharePoint® Server 2007, InfoPath Forms Services used ASP.NET session state. In Microsoft® SharePoint® Server 2010, InfoPath Forms Services uses the State Service. Both have protocols listed in section [2.2.2.10](#).

2.2.2.4 User Profile Service

The protocols in the following table enable the User Profile Service in Microsoft® SharePoint 2007 Products and Technologies only.

Protocol name	Description	Short name
User Profile Service Audiences Protocol Specification	Allows protocol clients to create, modify, refresh, and delete audiences from a store on the protocol server, as well as retrieve the audience members of a particular audience. In addition, the protocol allows protocol clients to begin, suspend, and cancel audience compilation on the protocol server. Applies to Microsoft® Office SharePoint® Server 2007.	[MS-UPSAUD]
User Profile Change Log Stored Procedure Protocol Specification	Allows protocol clients to add or delete user profile change events from a log on the protocol server. It also allows protocol clients to retrieve those user-profile change events by using predefined criteria, such as events that have happened after a specified time or with a specific user. Applies to Office SharePoint Server 2007.	[MS-UPSCHNG]
User Profile Service Push Protocol Specification	Describes how the User Profile Service (acting as the protocol client) can communicate with an earlier version of the User Profile Service (acting as the protocol server) to continually update user profiles and audience details. Applies to Office SharePoint Server 2007.	[MS-UPSGRAD]
User Profile Import Protocol Specification	Enables user data from a directory service (DS) to be imported into a user profile store. The supported directory services are Active Directory® Domain Services (AD DS), LDAP, and custom directories. AD DS and LDAP are used to import user profiles and member groups. Applies to Office SharePoint Server 2007.	[MS-UPSIMP]
User Profile Stored Procedures Protocol Specification	Enables a protocol client to access, store, and update information in the user profile store, including information for users and groups. In the SharePoint Products and Technologies implementation, this is primarily used between a front-end Web server and the back-end database server and between the application server and the back-end database server. Applies to Office SharePoint Server 2007.	[MS-UPSPROF]
User Profile Synchronization Stored Procedures Protocol Specification	Identifies the communication between a protocol client and a user profile store (protocol server) for the purpose of bidirectional synchronization. While ultimately the bidirectional synchronization of user profiles and user data occurs between the user profile store and content databases, this protocol does not cover communication between the protocol client and a content database. Rather, for the synchronization to succeed, the protocol client is responsible for using the Windows SharePoint Services (WSS): File Operations Database Communications Protocol [MS-WSSFO] and Windows SharePoint Services Technical Specification [MS-WSSTS] to read data from and write data to the content database during the synchronization. Applies to Office SharePoint Server 2007.	[MS-UPSSYNC]

The protocols in the following table enable the User Profile Service in Microsoft® SharePoint® 2010 Products and Technologies only.

Protocol name	Description	Short name
User Profile Service Audiences Version 2 Protocol Specification	Allows protocol clients to create, modify, refresh, and delete audiences from a store on the protocol server, as well as retrieve the audience members of a particular audience. In addition, the protocol allows protocol clients to begin, suspend, and cancel audience compilation on the protocol server. Applies to Microsoft® SharePoint® Server 2010.	[MS-UPSAUD2]
User Profile Change Log Stored Procedure Version 2 Protocol Specification	Allows protocol clients to add or delete user profile change events from a log on the protocol server and retrieve social activity data to generate activity feeds. It also allows protocol clients to retrieve those user profile change events by using predefined criteria, such as events that have happened after a specified time or with a specific user. Applies to SharePoint Server 2010.	[MS-UPSCHNG2]
User Profile Stored Procedures Version 2 Protocol Specification	Enables a protocol client to access, store, and update information in the user profile store, including information for users, groups, and organizations. In the SharePoint Products and Technologies implementation, this is primarily used between a front-end Web server and the back-end database server and between the application server and the back-end database server. Applies to SharePoint Server 2010.	[MS-UPSPROF2]
User Profile Synchronization Stored Procedures Version 2 Protocol Specification	Identifies the communication between a protocol client and a user profile store (protocol server) for the purpose of bidirectional synchronization. While ultimately the bidirectional synchronization of user profile and user data occurs between the user profile store and content databases, this protocol does not cover communication between the protocol client and a content database. Rather, for the synchronization to succeed, the protocol client is responsible for using Windows SharePoint Services (WSS): File Operations Database Communications Version 2 Protocol [MS-WSSFO2] and Windows SharePoint Services Technical Specification [MS-WSSTS] to read data from and write data to the content database during the synchronization. Applies to SharePoint Server 2010.	[MS-UPSSYNC2]
User Profile Social Data Stored Procedures Protocol Specification	Provides a way for a protocol client to interact with the social metadata store. The social metadata store holds various attribute objects that represent information about documents and Web pages that can be used across site collections. This protocol provides a way for the protocol client to retrieve this information, write new information, remove information, or update existing information for each type of social metadata. Applies to SharePoint Server 2010.	[MS-UPSSDSP]
User Profile Search Stored Procedures Protocol Specification	Enables the protocol client to search for a user, an organization, an audience, or a member group stored in a user profile store on the back-end database server. In the user profile store, each user, organization, audience, and member group can have multiple profile properties. Some of these properties can be identified as searchable profile properties. This protocol facilitates searching for users, organizations, audiences, or member groups by looking up values in those searchable profile properties. Applies to	[MS-UPSRCHSP]

Protocol name	Description	Short name
	SharePoint Server 2010.	
User Profile Admin Stored Procedures Protocol Specification	Provides a way for the protocol client to interact with partitions (1) in the user profile store on the respective protocol server. It enables protocol clients to create, import, export, delete, or simply list partitions (1) in the user profile store. In addition, it also allows protocol clients to read or update property values for a partition (1) on the protocol server. Applies to SharePoint Server 2010.	[MS-UPASP]
User Profile Import and Export Stored Procedures Protocol Specification	Used to import and export user profile and member group data to and from the user profile store. A typical scenario for using this protocol is a synchronization application that runs at fixed intervals to keep the user profile store and an LDAP directory service (DS) in sync. Applies to SharePoint Server 2010.	[MS-UPIESP]
User Profile Import and Export Web Service Protocol Specification	Enables importing and exporting user profile and member group data from the user profile store . A common usage scenario involves a synchronization application that runs at fixed intervals to keep the user profile store and a Lightweight Directory Access Protocol (LDAP) DS in sync.	[MS-UIIEWS]
User Profile Service Application Caching Stored Procedures Protocol Specification	Enables a protocol client to retrieve user information stored in the user profile store. It allows the protocol client to pass in a list of e-mail addresses, user names, GUID identifiers, user profile record identifiers, or security identifiers (SIDs) and returns the user profile data. This protocol can be used in scenarios that require access to user profile data of multiple users in one call. Applies to SharePoint Server 2010.	[MS-UPSCSP]
User Profile Property Service Application Web Service Protocol Specification	Allows protocol clients to start interacting with the User Profile Service. A typical scenario is a protocol client connecting to a site and accessing user profile information relevant to the logged-on user. The first step in any such interaction is to obtain information required to establish a connection, get information about the permissions the protocol client has, and identify URLs for the sites that offer the User Profile Service. This protocol helps in this first step. Applies to SharePoint Server 2010.	[MS-UPWCFWS]
User Profile Service Application Caching Web Service Protocol Specification	Allows protocol clients to access user profile information through the middle-tier application server tier, rather than the back-end database server tier. The middle-tier application server caches commonly used user profile fields, and exposes them to protocol clients that can read these fields through this protocol. This helps distribute the load from the database tier over to the application tier, because reading these commonly used fields is the most frequent operation. Applies to SharePoint Server 2010.	[MS-UPSCWS]
User Profile Synchronization (UPS): Configuration Data Structure	Identifies the data structures used in configuring the SharePoint Server 2010 User Profile Synchronization service.	[MS-UPSCDS]
User Profile Synchronization (UPS): Configuration Database	Identifies the communication between the User Profile Synchronization Configuration Service acting as a client and the User Profile Synchronization Configuration Database	[MS-UPSCDAP]

Protocol name	Description	Short name
Data Access Protocol Specification	acting as a server.	
User Profile Synchronization (UPS): Database Data Access Protocol Specification	Identifies the communication sequences used by the synchronization engine to perform data query and update commands on back-end database servers as part of file, user, and group synchronization operations.	[MS-UPSDBDAP]
User Profile Synchronization (UPS): Lightweight Directory Access Protocol Version 3 Extensions	Identifies the extensions to the Lightweight Directory Access Protocol (LDAP) [RFC4510] that are used in the communication sequences between a Synchronization Service and directory servers.	[MS-UPSLDAP]
User Profile Synchronization (UPS): Schema Exchange Protocol Profile	Identifies the User Profile Synchronization Schema Exchange Protocol Profile [MS-UPSSCXPP] . This profile enables protocol clients to discover metadata for a server implementation of the User Profile Synchronization (UPS): Configuration Data Structure [MS-UPSCDS] .	[MS-UPSSCXPP]
User Profile Synchronization (UPS): Management Objects Data Structure	Identifies the format for the objects that are used to run and determine status for synchronization between data sources.	[MS-UPSMODS]
User Profile Synchronization (UPS): Configuration Protocol Extensions	Used to create, read, update, delete, and enumerate objects that configure synchronization between connected data sources.	[MS-UPSCP]

2.2.2.5 Search Services

The protocols in the following table enable search services in Microsoft® SharePoint 2007 Products and Technologies only.

Protocol name	Description	Short name
Index Propagation Protocol Specification	Replicates data across multiple servers, and maintains consistency among those servers when the data changes. Applies to Microsoft® Office SharePoint® Server 2007.	[MS-CIPROP]
Global Admin Web Service Protocol Specification	Enables remote administration of the Search application. Applies to Office SharePoint Server 2007.	[MS-GLOADWS]
SPSCrawl Stored Procedures Protocol Specification	Allows protocol clients to read values of user profile properties for user profiles within the context of a site. Applies to Office SharePoint Server 2007.	[MS-SPSCLSP]
SQL Administration Protocol Specification	Identifies the communication between the front-end Web server and the back-end database server that is used to satisfy requests for common search administration tasks. This server-to-server protocol uses the Tabular Data Stream Protocol ([MS-TDS]) as its transport between the front-end Web server and the back-end database server. Applies to Office SharePoint Server 2007.	[MS-SQLPADM]
SQL Gatherer	Identifies the communication between the index server and back-	[MS-

Protocol name	Description	Short name
Protocol Specification	end database server that is used to satisfy requests for search crawl tasks. This server-to-server protocol uses the Tabular Data Stream Protocol ([MS-TDS]) as its transport between the index server and the back-end database server. Applies to Office SharePoint Server 2007.	SQLPGAT]
Search Service Database Query Protocol Specification	Identifies the communication between the front-end Web server and the back-end database server used to satisfy requests for queries against the property store and for query logging operations. Applies to Office SharePoint Server 2007.	[MS-SQLPQ]
MSSearch Query Protocol Specification	Allows a protocol client to communicate with a query server to issue queries. The search service running on the query server helps efficiently organize the extracted features of a collection of documents. When processing files, an index server analyzes a set of documents, extracts useful information, and then organizes the extracted information in such a way that properties of those documents can be efficiently returned in response to queries. A collection of documents that can be queried composes a catalog. A catalog might contain an inverted index (for quick word matching) and a property store (for quick retrieval of property values). The index server makes catalogs available to query servers by means of a propagation protocol. Applies to Office SharePoint Server 2007.	[MS-SQP]
Search Service Administration Web Service Protocol Specification	Enables the protocol client to configure and control the crawling function of the search subsystem. It allows the protocol client to define the URL space of the crawl by configuring starting points and restriction rules, thereby determining the credentials used during the crawl by the index server and controlling when the crawl process occurs. Applies to Office SharePoint Server 2007.	[MS-ADMWS]

The protocols in the following table enable search services in Microsoft® SharePoint® 2010 Products and Technologies only.

Protocol name	Description	Short name
Index Propagation Version 2 Protocol Specification	Replicates data across multiple servers and maintains consistency among those servers when the data changes. Applies to Microsoft® SharePoint® Server 2010.	[MS-CIPROP2]
SPSCrawl Stored Procedures Version 2 Protocol Specification	Allows protocol clients to read values of user profile properties for user profiles within the context of a site. Applies to SharePoint Server 2010.	[MS-SPSCLSP2]
SQL Administration Version 2 Protocol Specification	Identifies the communication between the front-end Web server and the back-end database server that is used to satisfy requests for common search administration tasks. This server-to-server protocol uses the Tabular Data Stream Protocol ([MS-TDS]) as its transport between the front-end Web server and the back-end database server. Applies to SharePoint Server 2010.	[MS-SQLPADM2]
SQL Gatherer Version 2 Protocol Specification	Identifies the communication between the index server and back-end database server that is used to satisfy requests for search crawl tasks. This server-to-server protocol uses the Tabular Data Stream Protocol ([MS-TDS]) as its transport between the index	[MS-SQLPGAT2]

Protocol name	Description	Short name
	server and the back-end database server. Applies to SharePoint Server 2010.	
Search Service Database Query Version 2 Protocol Specification	Identifies the communication between the front-end Web server and the back-end database server used to satisfy requests for queries against the property store and for query logging operations. Applies to SharePoint Server 2010.	[MS-SQLPQ2]
MSSearch Query Version 2 Protocol Specification	Allows a protocol client to communicate with a query server to issue queries. The search service running on the query server helps efficiently organize the extracted features of a collection of documents. When processing files, an index server analyzes a set of documents, extracts useful information, and then organizes the extracted information in such a way that properties of those documents can be efficiently returned in response to queries. A collection of documents that can be queried composes a catalog. A catalog might contain an inverted index (for quick word-matching) and a property store (for quick retrieval of property values). The index server makes catalogs available to query servers by means of a propagation protocol. Applies to SharePoint Server 2010	[MS-SQP2]
Search Admin Web Service Protocol Specification	Enables the protocol client to configure and control the crawling function of the search subsystem. It allows the protocol client to define the URL space of the crawl by configuring starting points and restriction rules, thereby determining the credentials used during the crawl by the index server and controlling when the crawl process occurs. Applies to SharePoint Server 2010.	[MS-DADMWS]
Search Site Administration Web Service Protocol Specification	Enables the protocol client to configure and control the specific SharePoint site functions such as Best Bets and Scopes. Applies to SharePoint Server 2010.	[MS-SADMWS]
Search Lotus Notes Web Service Protocol Specification	Enables the protocol client to enumerate and retrieve content and properties from a Lotus Notes Domino Server. Applies to SharePoint Server 2010.	[MS-NOTESWS]
Search Topology Protocol Specification	Enables the protocol client to perform the data query and update operations on the protocol server that are related to the search topology operations, such as adding and removing search components from the SharePoint Products and Technologies farm. Applies to SharePoint Server 2010.	[MS-SRCHTP]
Search Query Shared Services Protocol Specification	Enables the protocol client to issue search queries against a protocol server hosting a search service application. Applies to SharePoint Server 2010.	[MS-QSSWS]

The protocol in the following table enables the index format in both SharePoint 2007 Products and Technologies and SharePoint 2010 Products.

Protocol name	Description	Short name
Content Index Format Structure Specification	Identifies the format of the internal full-text index.	[MS-CIFO]

2.2.2.6 Business Connectivity Services

The protocols in the following table enable single sign-on for Business Connectivity Services in Microsoft® SharePoint 2007 Products and Technologies only.

Protocol name	Description	Short name
Single Sign-On Database Protocol Specification	Identifies an interface for protocol clients to store and retrieve credential and related information used to authenticate to the line-of-business (LOB) system.	[MS-SSOSP]
Single Sign-On Protocol Specification	Identifies a remote procedure call (RPC) -based protocol used by protocol clients of the Single Sign-On Database Protocol [MS-SSOSP] to obtain a master secret from a master secret server.	[MS-SSP]

The protocol in the following table enables information about line-of-business (LOB) systems in both SharePoint 2007 Products and Technologies and Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Business Data Catalog Database Protocol Specification	Identifies an interface for protocol clients to store and retrieve information about interfaces of line-of-business (LOB) systems and annotations of these interfaces.	[MS-BDCSP]

The protocols in the following table enable information about interfaces and line-of-business (LOB) systems in SharePoint 2010 Products only.

Protocol name	Description	Short name
Business Data Connectivity Database Protocol Specification	Identifies an interface for protocol clients to store and retrieve information about interfaces of other software systems and annotations of these interfaces.	[MS-BDCDPS]
Business Data Connectivity Web Service Protocol Specification	Identifies an interface for protocol clients to retrieve information about interfaces of other line-of-business applications and annotations of these interfaces.	[MS-BDCWPS]
Secure Store Database Protocol Specification	Provides secure access to a database of encrypted user credentials. Enterprises have a variety of data stored in various line-of-business (LOB) systems. Typically, each of these systems has its own security model where the same user is represented by a unique system-specific security principal (2) . A unique set of credentials is required as input before a user is allowed to access the line-of-business (LOB) system, which can be stored in the SSS.	[MS-SSDPS]
Secure Store Web Service Protocol Specification	Identifies an interface for protocol clients to store and retrieve credentials and related information that is typically used to authenticate to line-of-business (LOB) systems.	[MS-SSWPS]
Business Data Connectivity Model File Format	Identifies an XML structure used to describe the API of one or more line-of-business (LOB) systems and how that structure can be mapped to a well-defined interface of canonical operations. This	[MS-BDCMFFS]

Protocol name	Description	Short name
Specification	structure is typically consumed by a run-time engine that translates canonical interface operation execution requests into the native LOB system's interface operation execution requests.	

2.2.2.7 Content Management Service

The protocol in the following table enables remote import in both Microsoft® SharePoint 2007 Products and Technologies and Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Content Deployment Remote Import Web Service Protocol Specification	Enables a protocol client to remotely manage the import of a set of content that was previously exported from a SharePoint Products and Technologies protocol server to another protocol server.	[MS-CDEPLOY]

The protocol in the following table enables management of metadata in SharePoint 2010 Products only.

Protocol name	Description	Short name
Enterprise Metadata Service Database Schema Specification	Enables a protocol client to create, delete, merge, reuse, deprecate, describe, translate, and define synonyms for metadata terms to be applied to digital assets and resources.	[MS-EMMSTORE]

2.2.2.8 PerformancePoint Services

The protocols in the following table enable communications for PerformancePoint Services in Microsoft® SharePoint® 2010 Products and Technologies only.

Protocol name	Description	Short name
PerformancePoint Services Application Server Protocol Specification	Identifies the communications used between PerformancePoint Services on a front-end Web server and a back-end server storing business intelligence metadata.	[MS-PPSAPP]
PerformancePoint Services Stored Procedure Protocol Specification	Identifies the communication requests for scorecard annotations, user-selected filters, and temporary state data structures for data source first class object (FCO) administration tasks between the front-end Web server and the back-end database server. This server-to-server protocol uses the PerformancePoint Services Application Server Protocol, as described in [MS-PPSAPP], as its transport between the front-end Web server and the back-end database server.	[MS-PPSSPROC]

2.2.2.9 Word Automation Services

The protocols in the following table enable file conversion in Microsoft® SharePoint® Server 2010 only.

Protocol name	Details	Short name
Word Automation Services WCF Service Protocol Specification	Identifies the Windows Communication Foundation (WCF) service that is used to convert documents from one file format to another file format.	[MS-WORDSWCF]
Word Automation Services Stored Procedures Protocol Specification	Identifies how protocol clients add, modify, and delete groups of conversions and individual conversions from a database on the protocol server, as well as retrieve groups of conversions and individual conversions by using predefined criteria such as unique identifiers.	[MS-WORDSSP]

2.2.2.10 Session State

The protocol in the following table enables stored procedures for the ASP.Net Session State Service in Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® Server 2010.

Protocol name	Description	Short name
ASP.NET State Service Database Repository Communications Protocol Specification	Identifies ASP.NET session state stored procedures used to initialize, query, update, and insert data into the SSP database.	[MS-ASPSS]

The protocol in the following table enables stored procedures for the State Service in SharePoint Server 2010.

Protocol name	Description	Short name
Temporary State Service Protocol Specification	Identifies stored procedures used to store temporary data.	[MS-SPSTATE]

2.2.2.11 User Code Execution

The protocol in the following table enables user code to be run remotely. This protocol is used by the Sandboxed Code Service (called User Code Execution Service in the Technical Preview documents) in Microsoft® SharePoint® Server 2010.

Protocol name	Description	Short name
User Code Execution Protocol Specification	Used to remotely run user code within a server farm on one or more front-end Web servers that are primarily dedicated to running user code rather than responding to Web requests.	[MS-SPPTC]

2.2.2.12 Analytics Services

The protocol in the following table enables the retrieval of analytical data in Microsoft® SharePoint® Server 2010.

Protocol name	Description	Short name
Microsoft Office SharePoint Server (MOSS)	Used to retrieve analytical data from the	[MS-

Protocol name	Description	Short name
Analytics View Access Protocol Specification	Analytics Services application database.	MAVA

2.2.2.13 Visio Graphics Service

The protocol in the following table enables the Visio Graphics Service in Microsoft® SharePoint® 2010 Products and Technologies only.

Protocol name	Description	Short name
Visio Graphics Service Protocol Specification	Provides methods for retrieving Web drawings as raster or vector-formatted drawing pages.	[MS-VGSP]

The file format in the following table enables the Visio Graphics Service in SharePoint 2010 Products only.

Protocol name	Description	Short name
Visio Graphics Service (.vdx) File Format Specification	Provides the file that stores information about Web drawings, which includes drawing pages, shapes , fonts, images, data connections (1), and diagram update information.	[MS-VGSFF]

2.3 Environment

The following sections identify the context in which the system exists. This includes the systems that use the interfaces provided by this system of protocols, other systems that depend on this system, and, as appropriate, how components of the system communicate.

2.3.1 Dependencies on This System

The following system depends on the SharePoint Back-End Protocols system:

- The SharePoint Front-End Protocols system: [\[MS-SPFEPO\]](#)

2.3.2 Dependencies on Other Systems/Components

The SharePoint Back-End Protocols system depends on the following systems:

- Microsoft® Windows®: [\[MS-SYS\]](#) section 1
- Windows® SharePoint® Services File, Print, User/Group Administration: [\[MS-WSSO\]](#) section 1
- Active Directory® Domain Services (AD DS): [\[MS-ADTS\]](#) section 1

Microsoft® SharePoint 2007 Products and Technologies depends on the following components to function:

- The Windows Server® 2003 operating system with Service Pack 1
 - Internet Information Services (IIS) 6.0
- The Microsoft® .NET Framework version 3.0

- ASP.NET 2.0
- Microsoft® SQL Server® 2000 with Service Pack 3a

Microsoft® SharePoint® Server 2010 depends on the following systems/components to function:

- The Windows Server® 2008 operating system with Service Pack 2 (SP2)
 - IIS 7.0
- The Microsoft® .NET Framework version 3.5
 - ASP.NET 2.0
- Windows® Identity Foundation (WIF)
- Microsoft® SQL Server® 2005 with Service Pack 1

2.4 Assumptions and Preconditions

This section summarizes the assumptions and preconditions required by the system. The scope of this information is intended to be implementation independent and is limited to the system level.

- The SharePoint Products and Technologies back-end servers are reachable by the SharePoint Products and Technologies front-end servers by means of an established IP address (or IP addresses).
- The functional components of the SharePoint Products and Technologies back-end and front-end servers are started collectively, and the SharePoint Products and Technologies front-end servers accept user requests.
- The SharePoint Products and Technologies front-end servers can reach the SharePoint Products and Technologies back-end servers, and they have appropriate permissions to access data in the content and configuration databases.
- The versions of the SharePoint Products and Technologies front-end and back-end servers match or are within an acceptable range of versions. See [\[MS-WSSO\]](#) section 2.6 for more information about versioning.
- In the case where Active Directory® Domain Services (AD DS) is used to provide end-user authentication (2), the directory service (DS) is accessible to the SharePoint Products and Technologies server. Any intermediate firewalls, routers, or connection points between components of the system need to have all required ports and gateways open for communication between them.

2.5 Use Cases

The following use cases are provided to facilitate understanding of the SharePoint Back-End Protocols system overall.

Core platform use cases:

- Create a file from the client in a SharePoint library
- Use claims-based identity to connect line-of-business applications to SharePoint Service applications

Extended capabilities use cases:

- Excel® Services: Manage Requests from a front-end Web server
- Microsoft® Access Services: Manage the data operation request
- InfoPath Forms Services: Use the State Service
- User Profile Service: Retrieve a user's profile
- User Profile Service: Tag a document
- Search Services: Create and crawl a content source
- Session State: Communicate with a back-end database server

These use cases are not intended to provide a thorough and complete model of the system for any implementation.

2.5.1 Core Platform Use Cases

2.5.1.1 Create a File from the Client in a SharePoint Library

This use case describes the simplest way to create a file using the protocols covered in this system.

The following diagram illustrates this process. In this diagram the user and client computer are creating a text file "hello.txt" that contains the text "hello" in a Microsoft® SharePoint® Foundation 2010 document library.

This use case helps provide an understanding of how user interaction with the SharePoint Products and Technologies front-end protocols, in turn, uses the SharePoint Products and Technologies back-end protocols, such as the Windows SharePoint Services: File Operations Database Communications Version 2 Protocol [\[MS-WSSF02\]](#). This is summarized in steps 2, 3, and 5 that follow the diagram, and outlined in the Core Platform Protocols in section [2.2.1](#).

The use case is referenced in [\[MS-WSSO\]](#) section 2.5.1 and in [\[MS-SPFEPO\]](#) section 2.5.1.5.

The example in section [3.1](#) also provides more details.

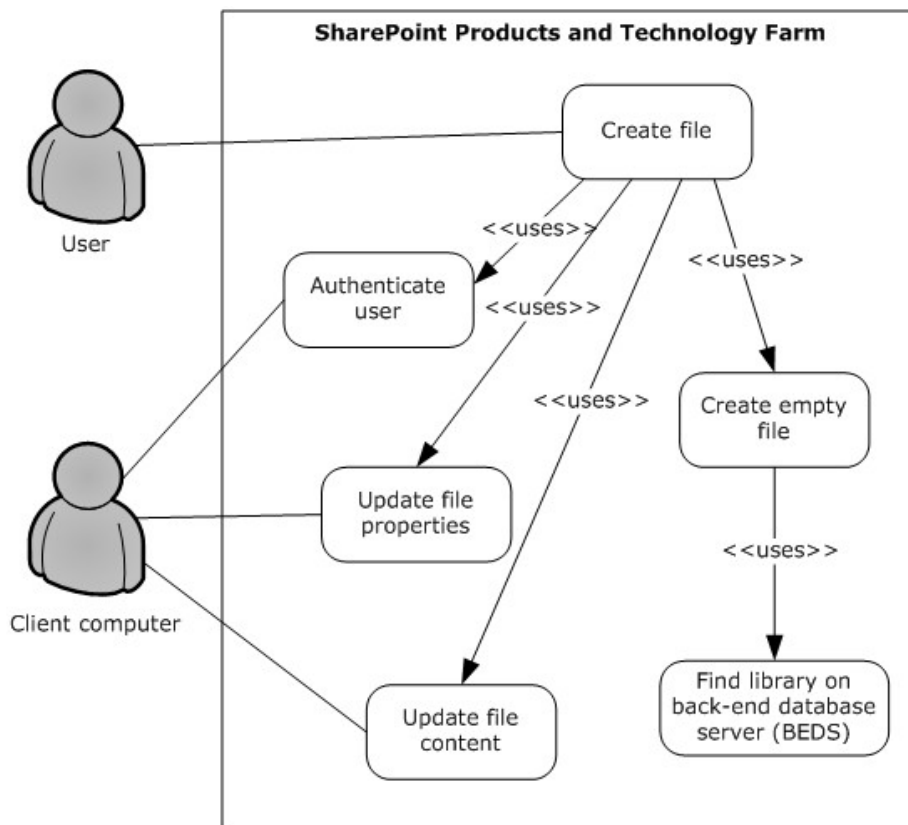


Figure 10: Process for creating a file from the client in a SharePoint library

Preconditions

- The user has permissions to an existing SharePoint library named "http://server/site/doclib".
- The user is logged on to a client computer running the Windows® 7 operating system (other client versions could have different steps), with an authenticated Microsoft® Windows® session, and can access the SharePoint site containing the library.

Steps

1. In the **Command Prompt** window, the user types the following command and presses ENTER:**echo hello >\\server\site\doclib\hello.txt** Where *server* is the name of the server (2), *site* is the name of the site, and *doclib* is the name of the document library.
2. The front-end Web server running Windows® SharePoint® Services 3.0 or SharePoint Foundation 2010 authenticates the user.
3. Windows SharePoint Services 3.0 or SharePoint Foundation 2010 finds the location of the document library and verifies that the user has access to it.
4. Windows SharePoint Services 3.0 or SharePoint Foundation 2010 creates an empty file in the library and confirms success to the client computer.
5. The client computer updates the file properties and file contents.

Errors

- The user does not have permission to the document library; the client notifies the user of access denied.
- The client cannot connect to the Windows® SharePoint® Services front-end Web server; the client notifies the user of the error.
- The client cannot update the file properties or contents; the client notifies the user of the error.

Post-conditions

- The file is created, and the file properties and file contents are updated.
- The file is not created and the user is notified of the error.

2.5.1.2 Use Claims-Based Identity to Connect Line-of-Business Applications to SharePoint Service Applications

This use case describes how a user who is using a SharePoint Products and Technologies Web Part rendered by a SharePoint Products and Technologies front-end Web server can interact with data presented by a SharePoint Products and Technologies application server, which is, in turn, retrieving the data from an external line-of-business (LOB) application.

To securely retrieve the data on behalf of the user who has permission to the external system, SharePoint Products and Technologies passes the user's claims to the external system by using claims-based identity.

The following diagram illustrates this process.

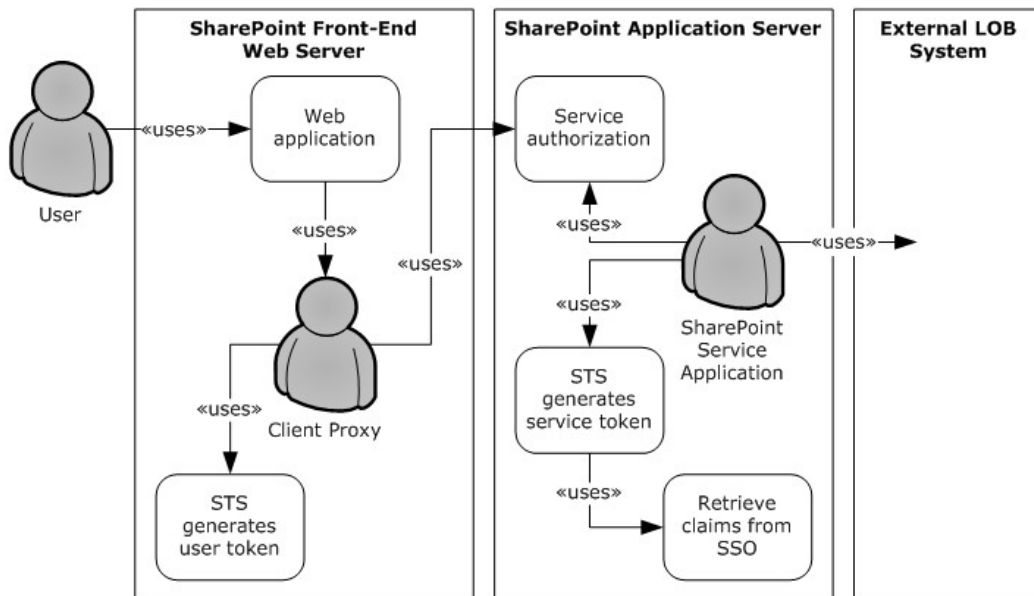


Figure 11: Process for using claims-based identity to connect line-of-business applications to SharePoint Service applications

Preconditions

- There is a trust already established by the **security token service (STS)** on the SharePoint Products and Technologies front-end Web server and the SharePoint Service application.
- There is a trust already established by the STS on the application server and the external line-of-business system STS.
- The user is properly authenticated to the server running SharePoint Products and Technologies.
- The user's credentials to access the line-of-business system have been provisioned on the SharePoint Products and Technologies application server.
- The user has access to the data in the line-of-business system.

Steps

1. The user accesses a Web application on the SharePoint Products and Technologies front-end Web server, which requires a call to a remote/external service to retrieve the data.
2. The client proxy gets a **Security Assertion Markup Language (SAML)** token containing the user's identity, and it gets the front-end application-pool identity from the STS.
3. The client proxy makes a Windows Communication Foundation (WCF) call to the application server that is passing the token to the service authorization action.
4. The SharePoint Service application in turn takes the verified token from the service authorization and directs its STS to generate a using the services identity, the user's identity, and any additional claims about the user (for example, the password for the line-of-business system stored in the single sign-on (SSO) database).
5. SharePoint Products and Technologies then calls the external line-of-business system to retrieve the data.

Post-conditions

- The SharePoint Service application is authenticated and authorized to retrieve data from the line-of-business system and returns that data to the SharePoint front-end web server.
- The SharePoint front-end web server returns the data to the user via the Web Part.

2.5.2 Extended Capabilities Use Cases

2.5.2.1 Excel Services

2.5.2.1.1 Manage Requests from a Front-End Web Server

This use case describes how Excel® Services manages requests from a SharePoint Products and Technologies front-end Web server for interaction with a workbook.

The following diagram illustrates this process.

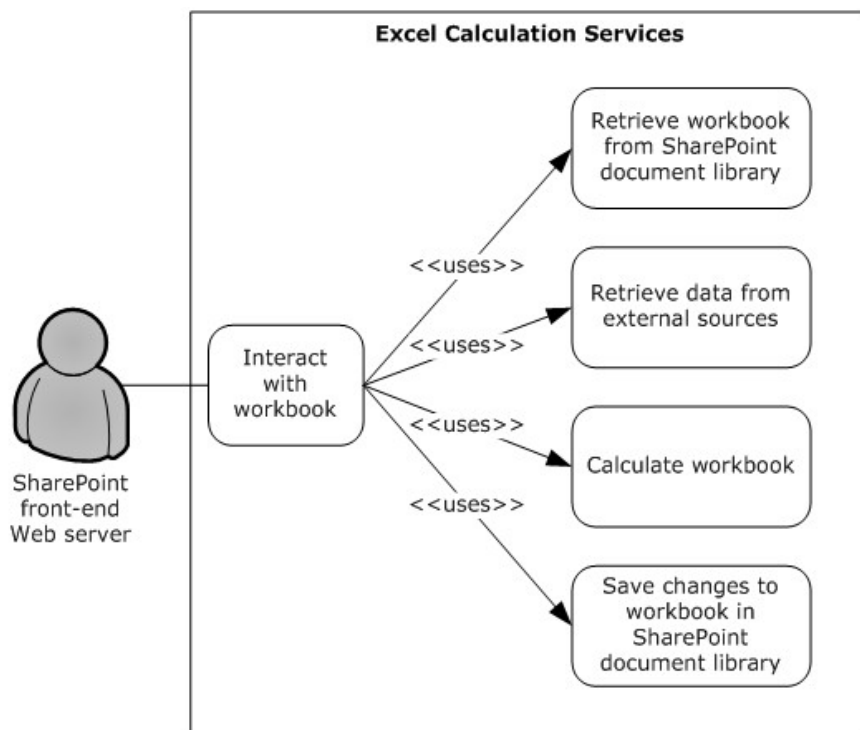


Figure 12: Process for managing requests from a front-end Web server

Preconditions

- Microsoft® Excel® Calculation Services is enabled on the SharePoint Products and Technologies farm.
- A user or application interacts with a workbook that has been published to a SharePoint site.

Steps

1. When a workbook is requested, Excel Calculation Services will retrieve the workbook using the SharePoint Products and Technologies object model application programming interfaces (APIs).
2. SharePoint Products and Technologies returns the workbook, which is then opened in a server session by Excel Calculation Services.
3. In accordance with the content in the workbook, and settings within the workbook, Excel Calculation Services will refresh external data and recalculate the workbook.
 - Refresh uses OLE DB [\[MSDN-OLEDB\]](#) or Open Database Connectivity (ODBC) [\[MSDN-ODBC\]](#) providers as required.
4. Once calculated, the requested range is returned to the SharePoint Products and Technologies front-end Web server.
5. If additional requests to set or get ranges, refresh data, or recalculate the workbook are received, Excel Calculation Services will perform the action and return the requested range to the SharePoint front-end Web server.

6. Upon request, Excel Calculation Services will save the workbook to a SharePoint site by means of the SharePoint Products and Technologies object model APIs.

Errors

- If the user does not have **rights** to the file, an error will be returned.
- If the workbook contains unsupported features that prevent it from loading, an error will be returned.
- If external data is required, and if the users does not have rights to that data, an error will be returned.
- If external data is required, and the necessary OLE DB [\[MSDN-OLEDB\]](#) or ODBC [\[MSDN-ODBC\]](#) provider is not present, an error will be returned.
- If, when a save is requested, the workbook has grown to exceed the size allowable to save in the SharePoint library, an error will be returned.

Post-conditions

- The operations requested by means of the SharePoint Products and Technologies front-end Web server are performed in Excel Calculation Services on the SharePoint Products and Technologies back-end server.
- Resulting changes to the workbook can be optionally saved or ignored, based on the SharePoint Products and Technologies front-end server request.
- The requested operations stop and the user is notified of the error.

2.5.2.2 Access Services

2.5.2.2.1 Manage the Data Operation Request

This use case describes how Microsoft® Access Services manages the data operation request from the SharePoint Products and Technologies front-end Web servers on the Microsoft® Access® 2010 application published to a SharePoint library.

The following diagram illustrates this process.

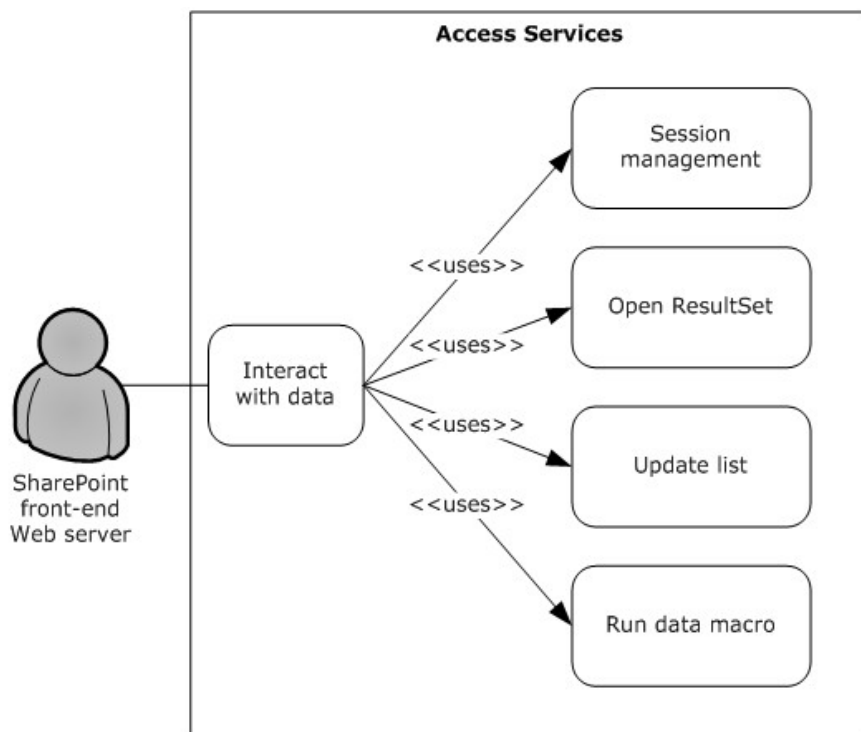


Figure 13: Process for managing the data operation request

Preconditions

- Access Services is enabled on the SharePoint Products and Technologies farm.
- A user interacts with the Access 2010 application that has been published to a SharePoint site.

Steps

1. When the user browses to a data-bound form or datasheet on the SharePoint front-end Web server, Access Services will create a session for that form or datasheet.
2. One or more ResultSets are created in that session; these ResultSets contain the list items that are used to populate the form or datasheet.
3. Data operations such as insert, update, and delete are performed on the SharePoint list, and if successful, the ResultSets are updated to be consistent with the underlying data.
4. The front-end Web server can also request that the application server run a **data macro**.

Post-condition

The data operation requested by the SharePoint Products and Technologies front-end Web server is performed on the SharePoint Products and Technologies back-end server; the result is synchronized in the SharePoint lists that publish the application.

2.5.2.3 InfoPath Forms Services

2.5.2.3.1 Use the State Service

This use case describes how InfoPath® Forms Services uses the State Service in Microsoft® SharePoint® Server 2010. This use case complements use case, Publish a Form Template and Enable Browser Rendering, in [\[MS-SPFEPO\]](#) section 2.5.2.3.1, covering a back-end protocol that is supporting that front-end use case.

In this case, the InfoPath Forms Services features running on a SharePoint Products and Technologies front-end Web server use the back-end State Service to temporarily store data that a user is adding to an InfoPath form by using a browser. In some scenarios the user could take a significant amount of time to fill out the form, possibly hours, and individual browser interactions could result in the user sending requests to different SharePoint Products and Technologies front-end Web servers. By using the State Service, InfoPath Forms Services is able to give the user a coherent browser session experience where the form does not reset or lose data.

The following diagram illustrates this process.

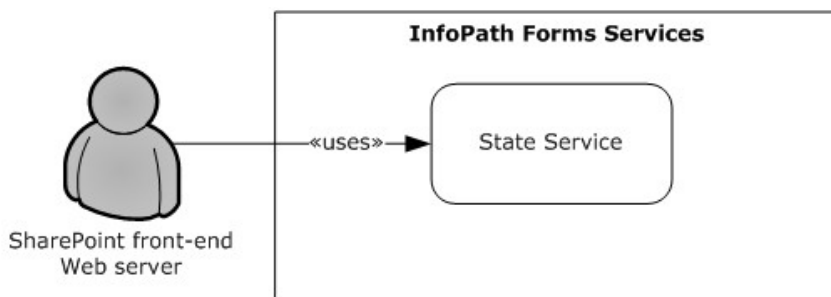


Figure 14: Process for using the State Service

Preconditions

- InfoPath Forms Services is enabled on the SharePoint Products and Technologies farm.
- The user interacts with InfoPath Forms Services in a way that state information needs to be persisted across HTTP requests, for example taking a long time to fill out a form.

Steps

1. The user opens a published InfoPath form on the SharePoint Products and Technologies front-end server.
2. The user enters some data on the form, and then leaves the session untouched for 2 hours.
3. InfoPath Forms Services uses the State Service to store data.
4. The user returns to the form and starts to enter more data, and when the front-end Web server refreshes the page, InfoPath Forms Services can retrieve the data from the State Service to keep all data the user has entered in the form session complete.

Error

- The State Service fails to respond to InfoPath Forms Services and user data is not saved; the open form will disappear from the form fields.

Post Condition

- The user saves or submits the completed form and InfoPath Forms Services removes any form data from the State Service.

2.5.2.4 User Profile Service

These use cases describe how the User Profile Service manages the data operation request from the SharePoint Products and Technologies front-end Web servers to present a user's profile.

2.5.2.4.1 Retrieve a User's Profile

This use case describes how a client application can retrieve a user's profile; this includes all information in the user's profile that is accessible to the client. Information presented to the client is based on the privacy level set on the user profile properties by the administrator, as a default, or by the user. For example, if a user wants only colleagues, and no one else, to see the user's birthday, the user can choose to do so through privacy settings. During access to the user's profile, only those users identified in the User Profile Service as colleagues will be shown the user's birthday.

The following diagram illustrates this process.

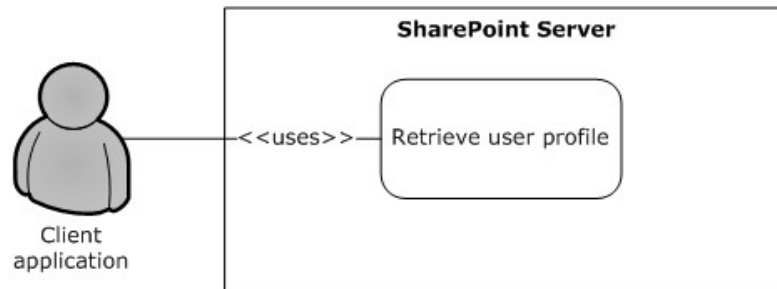


Figure 15: Process for retrieving a user's profile

Preconditions

- The client application needs to have rights to be able to access and read the user's profile.
- The requested user exists in the user profile store.

Steps

1. The client application requests access to a user's profile.
2. The User Profile Service retrieves properties for the user profile. This retrieval is filtered for appropriate privacy settings.
3. The properties are then presented to the client application, formatted in a site template and applied to the user's personal profile site.

Error

If the client application does not have access or read rights to the user's profile, an access denied error can occur.

Post-conditions

- The client sees the user's profile, which typically includes a picture, information about the user, and additional properties if shared and entered by the user.
- The client application cannot retrieve a user's profile and receives an error.

2.5.2.4.2 Tag a Document

This use case describes how a client application can apply a tag to a document, to call out a point of interest about the document, or include the document in a specific topic or grouping of interest where other items in the topic share the same tag.

The following diagram illustrates this process.

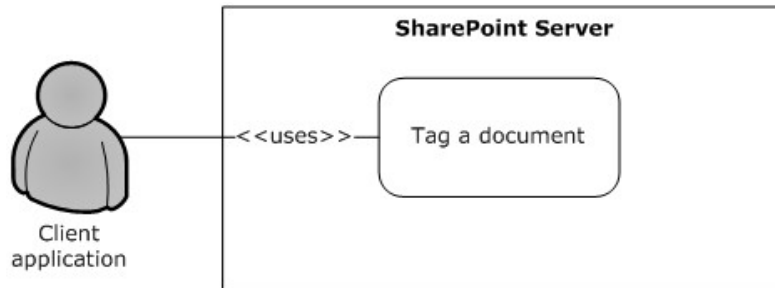


Figure 16: Process for tagging a document

Preconditions

- The client needs to have the rights to be able to access the location of the document.
- The client needs to have the rights to apply tags by using the User Profile Service.
- The Content Management Service (section [2.1.3.7](#)) is provisioned and is operational.

Steps

1. The client requests matching terms in the Content Management Service term store, by providing initial words of the tag (for example, "soc" for social computing as an intended tag).
2. A set of suggested terms is returned to the client.

Upon acceptance of a suggested term, a GUID for this term is stored in the user profile store, along with the location of the document, information about the client's identity, and time of tag creation.

Error

If the client does not have the rights to apply tags in the User Profile Service, tagging access will be disabled.

Post-conditions

- The client sees the intended tag applied to the document location.
- Tagging is disabled if the client application does not have the rights to apply tags.

2.5.2.5 Search Services

2.5.2.5.1 Create and Crawl a Content Source

This use case describes how a user can create a content source and start crawls against a particular search service application. Once the crawl is finished, the most common scenario is for the search engine to start traversing the URL space specified in the search configuration and build a full-text index and a metadata index to use for queries.

The following diagram illustrates this process.

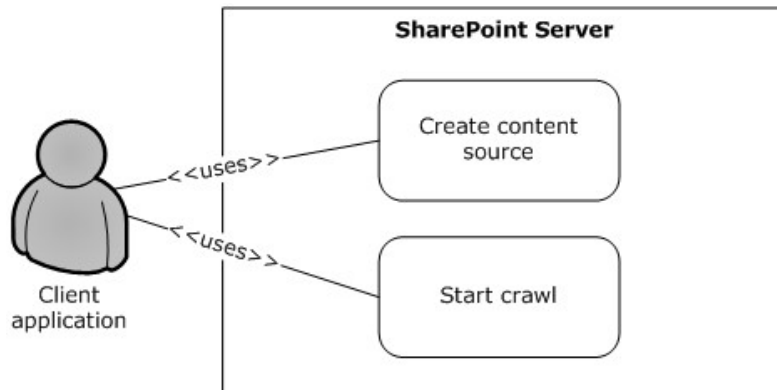


Figure 17: Process for creating and crawling a content source

Precondition

- The client application needs to have the rights to be able to access the search administration pages and update configuration information.

Steps

1. The client creates a content source by specifying the start addresses and crawl configuration for a particular repository. Crawl configurations include site hops, page depth, crawl schedules, and so on.
2. The client commits the configuration to the search application.
3. The client starts a crawl on the content source that was created.

Errors

- If the client does not have the rights to the Search Services application, an access denied error can occur.
- If a content source already exists with the same configuration, an error can occur.
- If a crawl operation is already ongoing on the particular content source, an error can occur.

Post-conditions

- The search engine begins a crawl by starting from the start address and traversing the URL space defined by the configuration of the content source.
- The user interface displays the status of the crawl to the client.

- The crawl stops and the client is notified of the error.

2.5.2.6 Session State

2.5.2.6.1 Communicate with a Back-End Database Server

This use case describes how ASP.NET session state or the State Service communicates with a back-end database server.

This use case is a more detailed version of the use case covered in section [2.5.2.3.1](#). In this case, the features running on a SharePoint Products and Technologies front-end Web server use the back-end State Service to temporarily store data a user generates while using the feature over the life of a browser session. In some scenarios the user could take significant time to complete the whole task, possibly hours, and individual browser interactions could result in the user sending requests to different SharePoint Products and Technologies front-end Web servers. By using the State Service feature, the user gains a coherent browser session experience where the form does not reset or lose data.

The following diagram illustrates this process.

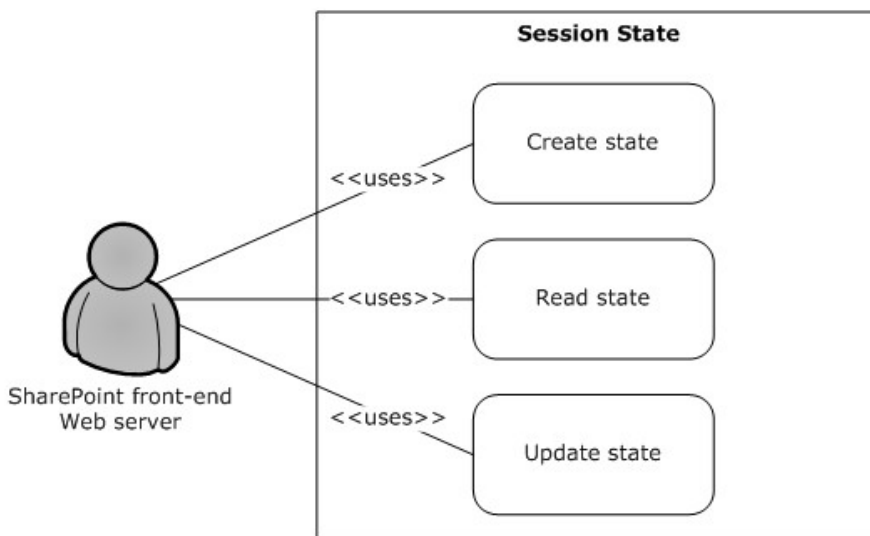


Figure 18: Process for communicating with a back-end database server

Preconditions

- ASP.NET session state or the State Service is enabled.
- A user interacts with a SharePoint Products and Technologies application on the front-end Web server that uses ASP.NET session state or the State Service.

Steps

1. The front-end Web server requests the ASP.NET session state or the State Service to store data.
2. The ASP.NET session state or the State Service persists the data to the database.
3. The application requests data; the ASP.NET session state or the State Service requests the data from the database.

4. The application changes the data; the ASP.NET session state or the State Service persists the updated data to the database.

Post-condition

- Updated data exists in a row in the state database.

2.6 Versioning, Capability Negotiation, and Extensibility

The SharePoint Products and Technologies front-end and back-end servers perform explicit version verifications.

See [\[MS-WSSO\]](#) section 2.6 for more information about SharePoint Products and Technologies front-end and back-end versioning.

2.7 Error Handling

There are no system-level error-handling behaviors. In general, for errors returned as part of a protocol in this system, the protocol documents describe what the error means. How they are handled, based on the protocol description, is an issue for the implementer.

2.8 Coherency Requirements

This system has no special coherency requirements beyond the specific details that are covered in the individual protocol documents.

2.9 Security

For a detailed overview of SharePoint Products and Technologies authentication (2) and authorization, as well as the core concepts used to manage security in SharePoint Products and Technologies see [\[MS-WSSO\]](#) section 2.9.

2.10 Additional Considerations

There are no additional considerations.

3 Examples

The examples in the following sections provide more information about the use and operation of the SharePoint Back-End Protocols system, especially interactions between system components. Protocol-level examples can be found in the individual protocol documents. The following system-level examples are provided in this document:

- Create a file from the client
- Configure a crawl, run a crawl, and query for content

These examples are subject to the following, possible common errors:

- Client unable to connect to front-end server.
- Front-end server unable to connect to Active Directory® Domain Services (AD DS) to authenticate the user.
- Front-end server unable to connect to back-end database server.

3.1 Example 1: Create a File from the Client

This example describes how to create a file from the client by using the protocols covered in this system. This example uses the "Create a File from the Client in a SharePoint Library" use case that is described in section [2.5.1.1](#).

It helps provide an understanding of how user interaction with the SharePoint Products and Technologies front-end protocols, in turn, uses the back-end protocols, such as the Windows SharePoint Services: File Operations Database Communications Version 2 Protocol [\[MS-WSSFO2\]](#), outlined in section [2.2.1.2](#).

This example assumes the following:

- The user has permissions to an existing SharePoint library named "http://server/site/doclib".
- The user is logged on to a client computer running the Windows® 7 operating system (other client versions will have different steps), with an authenticated Microsoft® Windows® session, and can access the Windows® SharePoint® Services site containing the library.
- Using a Windows command line, the user types the following command:

```
echo hello > \\server\site\doclib\hello.txt
```

The following steps illustrate the file creation process by a user using a Windows command line, assuming that no errors occur.

Some of the [\[MS-WSSFO2\]](#) examples referenced for more details in the following steps use the SharePoint Products and Technologies programming API as described in each example referenced; for more information about these APIs, see [\[MSDN-SharePointSDK\]](#).

To highlight a specific case, step 3 in this example references [\[MS-WSSFO2\]](#) section 4.2 to provide more details on what happens when a user has not visited the SharePoint Products and Technologies site before. The [\[MS-WSSFO2\]](#) section 4.2 example is created by calling the SharePoint Products and Technologies programming API: Group.Users.Add(). This example is called out because the actual steps generated between the front-end Web server and back-end database server might be different when the request is initiated by user interaction with the front-end Web server using a browser or when using the Windows command line, as in this case.

1. After the user initiates the **echo** command, the client sends a WebDAV request (as specified in [\[RFC2518\]](#)) to the front-end Web server, asking it to perform a PUT operation on the hello.txt file in the document library.
2. The front-end Web server (IIS) authenticates the user with Active Directory® Domain Services (AD DS). In practice, this might involve multiple LDAP requests with AD DS, especially if the user has not visited the site before.
3. AD DS responds with multiple LDAP results.

For more detail on authentication (2), see [\[MS-WSSO\]](#) section 2.9.2.1.

For more information about the scenario where the user has not visited the site before, see [\[MS-WSSFO2\]](#) section 4.2 for Microsoft® SharePoint® Server 2010.

4. In multiple round trips with the back-end database server, the front-end Web server locates the content database for the document library and confirms that the library exists.
5. The back-end database server returns multiple objects for the site collection, Web site, and library to the front-end Web server.

For more information about steps 4 and 5, see [\[MS-WSSFO2\]](#) section 4.6 for SharePoint Server 2010.

6. In multiple round trips with the back-end database server, the front-end Web server creates an empty file in the document library, and then, if successful, also verifies that the user has permissions to access and write to the document library.

7. The back-end database server returns multiple result sets as part of the process to create the file.

For more information about file creation, see [\[MS-WSSFO2\]](#) section 4.9 for SharePoint Server 2010.

8. The front-end Web server returns a WebDAV response, saying the file was created successfully.

9. The client sends a WebDAV HEAD request to the front-end Web server, with the URL to the hello.txt file in the document library, to verify the success of the previous call.

10. In multiple round trips with the back-end database server, the front-end Web server retrieves the file.

11. The back-end database server returns multiple result sets as part of the process to retrieve the file.

For more information about file retrieval see [\[MS-WSSFO2\]](#) section 4.1 for SharePoint Server 2010.

12. In response to the HEAD request, the front-end Web server sends a response saying the request was successful.

13. Then the client sends a WebDAV PUT request to the front-end Web server, which will have multiple parts to upload the file and update its properties.

14. The client sends a WebDAV request to the front-end Web server with an XML payload that has the file properties from the client.

15. The client sends a WebDAV request to the front-end Web server with an XML payload that has the file content; in this example, that content is simply the word "hello".
16. In multiple round trips with the back-end database server, the front-end Web server updates the file and its properties in the document library.
17. The back-end database server returns multiple result sets as part of the process to update the files.
18. On completing the update, the front-end Web server sends a WebDAV response, saying the request was successful.

The following sequence diagram illustrates the message sequence for this example. It shows the consolidation of multiple front-end Web servers to back-end database server actions, and multiple front-end Web servers to AD DS actions and into single flows.

This diagram does not document some of the initial interactions between client and server that optionally happen on some clients and that can also depend on whether the client has connected to the site previously to verify whether the server is able to support WebDAV.

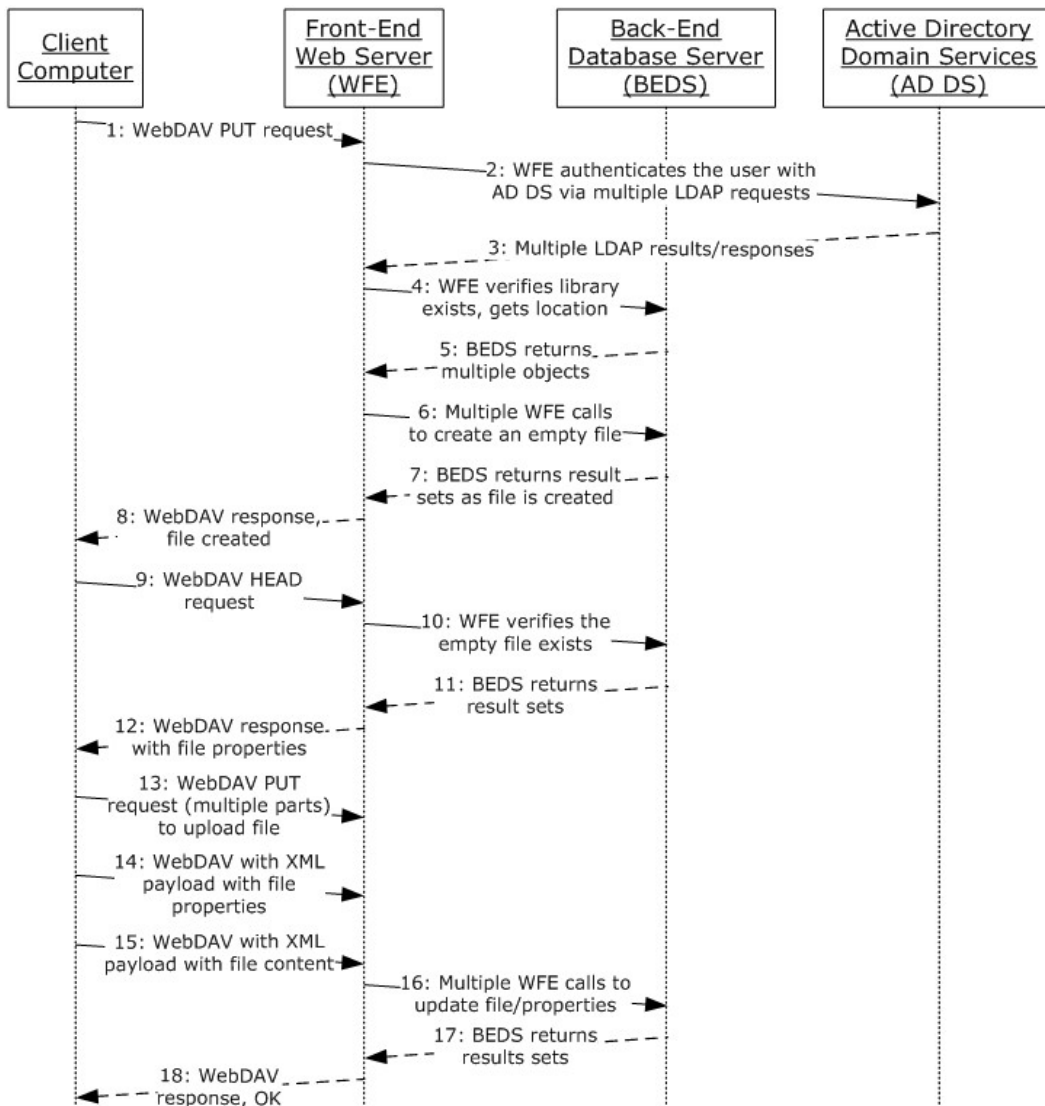


Figure 19: Sequence for creating a file from the client

3.2 Example 2: Configure a Crawl, Run a Crawl, and Query for Content

This example describes how to configure a content source for crawling, run a crawl operation, and query for results. This example uses the search services use cases in section [2.5.2.5](#).

This scenario assumes the following:

- The user has permissions to the search service application that is being configured.
- The user is logged on to a client computer running the Windows® 7 operating system (other client versions will have different steps), with an authenticated Microsoft® Windows® session, and can access the Windows® SharePoint® Services site.

The following steps illustrate the process for configuring and running a crawl and querying content, assuming that no errors occur.

Steps 1 and 2 use the SharePoint Products and Technologies programming API; the actual steps might vary when the request is generated by user interaction with the front-end Web server.

1. The user reads the current configuration of the crawler application and retrieves the current list of content sources.

2. The user adds a new content source and updates its properties.

For more information about creating and configuring a content source, see [\[MS-DADMWS\]](#) section 4.1.

Steps 3 and 4 use the SharePoint Products and Technologies programming API; the actual steps might vary when the request is generated by user interaction with the front-end Web server.

3. The user gets the current list of content sources to select the content source to crawl.

4. The user starts a crawl operation on a particular content source.

For more information about crawling a content source, see [\[MS-DADMWS\]](#) section 4.3.

5. The crawler application starts traversing the URL space as specified in the configuration.

6. For SharePoint Products and Technologies content, the crawler application communicates with the SiteData Web service and retrieves site collection information by calling GETSITE.

7. The crawler application traverses through lists and list items by calling GETLIST and GETLISTITEM against the SiteData Web service.

For more information about the SiteData Web Service see [\[MS-SITED3S\]](#) section 4.1, [\[MS-SITED3S\]](#) section 4.2, and [\[MS-SITED3S\]](#) section 4.3.

8. While retrieving content and metadata from the SiteData Web service, the crawler application uses the back-end database server to store crawl history information and creates a full-text index catalog and a metadata index for the content being crawled.

9. The crawler application propagates the full-text index catalog to a query server.

For more information about the interaction between the crawl application and the back-end database server, see [\[MS-SQLPGAT2\]](#) section 3. For more information about the propagation of the full-text index catalog to a query server, see [\[MS-CIPROP2\]](#) section 4.1. For more information about the structure of the full-text index catalog, see [\[MS-CIFO\]](#).

Step 10 uses the SharePoint Products and Technologies programming API; the actual step might vary when the request is generated by user interaction with the front-end Web server.

10. After the crawler application is finished, the user runs a query against the search query Web service using any query term and views the result set retrieved from the full-text index catalog and the metadata index.

For more information about running queries, see [\[MS-SEARCH\]](#) section 4.3.

4 Microsoft Implementations

There are no variations in the behavior of the SharePoint Back-End Protocols system in different versions of Windows® SharePoint® Services and Microsoft® SharePoint® Server beyond those described in the specifications of the protocols supported by the system, as listed in section [2.2](#).

The information in this specification is applicable to the following versions of Windows SharePoint Services and SharePoint Server:

- Windows® SharePoint® Services 3.0
- Microsoft® SharePoint® Foundation 2010
- Microsoft® Office SharePoint® Server 2007
- Microsoft® SharePoint® Server 2010

Exceptions, if any, are noted in the following section.

4.1 Product Behavior

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

5 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

6 Index

A

[Abstract](#) 1
[Access Services overview](#) 23
[Additional considerations](#) 69
[Analytics Services overview](#) 40
[Applicable protocols](#) 41
[Architecture](#) 16
[ASP.NET session state overview](#) 39
[Assumptions](#) 56

B

[Back-end protocols overview](#) 19
[Business Connectivity Services overview](#) 36

C

[Capability negotiation](#) 69
[Change tracking](#) 76
[Coherency requirements](#) 69
Communicate with a back-end database server
 [overview](#) 68
[Communications](#) 55
 [with other systems](#) 55
 [within the system](#) 55
[Component dependencies](#) 55
[Concepts](#) 16
[Configuration protocols overview](#) 20
Configure a crawl
 run a crawl
 and query for content
 [details](#) 73
Considerations
 [additional](#) 69
 [security](#) 69
[Content Management Service overview](#) 36
[Content storage overview](#) 21
Core platform protocols
 [configuration protocols](#) 20
 [content storage](#) 21
[Core platform protocols overview](#) 20
Create a file from the client
 [details](#) 70
Create a file from the client in a SharePoint library
 [overview](#) 57
Create and crawl a content source
 [overview](#) 67

D

[Data storage overview](#) 18
Dependencies
 [with other systems](#) 55
 [within the system](#) 55
Deployment topology
 [back-end protocols](#) 19
 [data storage](#) 18
 [front-end protocols](#) 18

[overview](#) 16
 [SharePoint farm](#) 17
Design intent
 [communicate with a back-end database server](#) 68
 [create a file from the client in a SharePoint library](#)
 57
 [create and crawl a content source](#) 67
 [manage requests from a front-end Web server](#) 60
 [manage the data operation request](#) 62
 [overview](#) 56
 [retrieve a user's profile](#) 65
 [tag a document](#) 66
 [use claims-based identity to connect line-of-
 business applications to SharePoint Service](#)
 [applications](#) 59
 [use the State Service](#) 64

E

[Environment](#) 55
[Error handling](#) 69
Examples
 configure a crawl
 run a crawl
 [and query for content](#) 73
 [create a file from the client](#) 70
 [overview](#) 70
 [query for content](#) 73
 [run a crawl](#) 73
[Excel Services overview](#) 22
Extended capabilities protocols
 [Access Services](#) 23
 [Analytics Services](#) 40
 [Business Connectivity Services](#) 36
 [Content Management Service](#) 36
 [Excel Services](#) 22
 [InfoPath Forms Services](#) 24
 [PerformancePoint Services](#) 37
 [search services](#) 32
 [session state](#) 38
 [User Code Execution Service](#) 40
 [User Profile Service](#) 25
 [Visio Graphics Service](#) 40
 [Word Automation Services](#) 37
[Extended capabilities protocols overview](#) 21
Extensibility
 [Microsoft implementations](#) 75
 [overview](#) 69
[External dependencies](#) 55

F

Farm
 [SharePoint overview](#) 17
[Front-end protocols overview](#) 18
[Functional architecture](#) 16
[Functional requirements - overview](#) 16

G

[Glossary](#) 8

H

[Handling requirements](#) 69

I

[Implementations - Microsoft](#) 75
[Implementer - security considerations](#) 69
[InfoPath Forms Services overview](#) 24
[Informative references](#) 9
[Initial state](#) 56
[Introduction](#) 7

M

Manage requests from a front-end Web server
[overview](#) 60
Manage the data operation request
[overview](#) 62
[Microsoft implementations](#) 75

O

Overview
[abstract](#) 1
[Access Services](#) 23
[Analytics Services](#) 40
[ASP.NET session state](#) 39
[back-end protocols](#) 19
[Business Connectivity Services](#) 36
[configuration protocols](#) 20
[Content Management Service](#) 36
[content storage](#) 21
[core platform protocols](#) 20
[data storage](#) 18
[deployment topology](#) 16
[examples](#) 70
[Excel Services](#) 22
[extended capabilities protocols](#) 21
[front-end protocols](#) 18
[InfoPath Forms Services](#) 24
[PerformancePoint Services](#) 37
[search services](#) 32
[session state](#) 38
[SharePoint farm](#) 17
[State Service](#) 39
[summary of protocols](#) 41
[synopsis](#) 16
[User Code Execution Service](#) 40
[User Profile Service](#) 25
[User Profile Services – audiences](#) 31
[User Profile Services – content database](#) 30
[User Profile Services – core service](#) 26
[User Profile Services – data import and export](#) 29
[User Profile Services – indexing the profile service data](#) 30
[User Profile Services – personalization and targeting](#) 31
[User Profile Services – social metadata](#) 31
[Visio Graphics Service](#) 40

[Word Automation Services](#) 37

P

[PerformancePoint Services overview](#) 37
[Preconditions](#) 56

Q

Query for content
[details](#) 73

R

[References](#) 9
Requirements
[coherency](#) 69
[error handling](#) 69
[overview](#) 16
[preconditions](#) 56
Retrieve a user's profile
[overview](#) 65
Run a crawl
[details](#) 73

S

[Search services overview](#) 32
[Security considerations](#) 69
[Session state overview](#) 38
[SharePoint farm overview](#) 17
[State Service overview](#) 39
[System architecture](#) 16
[System dependencies](#) 55
[with other systems](#) 55
[within the system](#) 55
[System errors](#) 69
System overview
[abstract](#) 1
[introduction](#) 7
[System protocols](#) 41
[System requirements - overview](#) 16
System use cases
[communicate with a back-end database server](#) 68
[create a file from the client in a SharePoint library](#) 57
[create and crawl a content source](#) 67
[manage requests from a front-end Web server](#) 60
[manage the data operation request](#) 62
[overview](#) 56
[retrieve a user's profile](#) 65
[tag a document](#) 66
[use claims-based identity to connect line-of-business applications to SharePoint Service applications](#) 59
[use the State Service](#) 64

T

[Table of protocols](#) 41
Tag a document
[overview](#) 66

[Tracking changes](#) 76

U

[Use cases](#) 56

[communicate with a back-end database server](#) 68

[create a file from the client in a SharePoint library](#)
57

[create and crawl a content source](#) 67

[manage requests from a front-end Web server](#) 60

[manage the data operation request](#) 62

[retrieve a user's profile](#) 65

[tag a document](#) 66

[use claims-based identity to connect line-of-
business applications to SharePoint Service
applications](#) 59

[use the State Service](#) 64

Use claims-based identity to connect line-of-
business applications to SharePoint Service
applications

[overview](#) 59

Use the State Service

[overview](#) 64

[User Code Execution Service overview](#) 40

[User Profile Service overview](#) 25

V

Versioning

[Microsoft implementations](#) 75

[overview](#) 69

[Visio Graphics Service overview](#) 40

W

[Word Automation Services overview](#) 37