

[MS-SPBEPO2]: SharePoint Back-End Protocols Overview Version 2

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.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

This document provides an overview of the SharePoint Back-End Protocols Overview Version 2 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.

The SharePoint Back-End Protocols system is designed for communication within the servers and to enable communication between the SharePoint Products and Technologies components. The system consists of two main groups: core platform and extended capabilities. The extended capabilities protocols build on top of the core platform protocols, which provide basic storage and administrative services. The core platform protocols, such as [\[MS-WSSCFGD3\]](#), are designed to enable the retrieval, sharing, and storage of configuration data; while the core platform protocols, such as [\[MS-WSSCADM3\]](#), are designed to enable communications for content storage. The extended capabilities protocols are used to interoperate with the extended capabilities components, such as [\[MS-EXSPXML3\]](#) for Excel Services, [\[MS-ADS\]](#) for Access Services, InfoPath Forms Services, [\[MS-UPSAUD3\]](#) for User Profile Service, [\[MS-FQL2\]](#) for Search Services, [\[MS-BDCSP\]](#) for Business Connectivity Services, [\[MS-CDEPLOY\]](#) for Content Management Service, [\[MS-PPSAPP\]](#) for PerformancePoint Services, [\[MS-WORDSWCF\]](#) for Word Automation Services, [\[MS-ASPSS\]](#) for Session State, [\[MS-SPPTC\]](#) for User Code Execution Service, [\[MS-VGSP\]](#) for Visio Graphics Service, [\[MS-WMS\]](#) for Work Management, [\[MS-PASCWS\]](#) for PowerPoint Automation Services, and [\[MS-TSWS\]](#) for Translation Services.

Revision Summary

Date	Revision History	Revision Class	Comments
01/20/2012	0.1	New	Released new document.
04/11/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
07/16/2012	0.2	Minor	Clarified the meaning of the technical content.

Preliminary

Table of Contents

1 Introduction	7
1.1 Glossary	8
1.2 References	9
2 Functional Architecture	15
2.1 Overview	15
2.1.1 Deployment Topology	15
2.1.1.1 SharePoint Products and Technologies Farm	16
2.1.1.2 Data Storage	17
2.1.1.3 Front-End Protocols	17
2.1.1.4 Back-End Protocols	17
2.1.2 Core Platform	19
2.1.2.1 Configuration Protocols	19
2.1.2.1.1 Configuration Database	19
2.1.2.1.1.1 Site Map	19
2.1.2.1.1.1.1 Site Collection Lookup	20
2.1.2.1.2 Service Application Database	22
2.1.2.1.3 Usage Database	22
2.1.2.2 Content Storage	22
2.1.3 Extended Capabilities	22
2.1.3.1 Excel Services	23
2.1.3.2 Access Services	24
2.1.3.3 InfoPath Forms Services	27
2.1.3.4 User Profile Service	27
2.1.3.4.1 Core Service	28
2.1.3.4.1.1 Properties	29
2.1.3.4.1.2 Organizations	29
2.1.3.4.1.3 Profiles	29
2.1.3.4.1.4 Multiple Values	29
2.1.3.4.1.5 Term Sets (Vocabularies)	29
2.1.3.4.1.6 Colleagues	30
2.1.3.4.1.7 Memberships	30
2.1.3.4.1.8 Profile Change Log	30
2.1.3.4.1.9 Social Data	30
2.1.3.4.2 Data Import and Export	30
2.1.3.4.2.1 Active Directory	31
2.1.3.4.2.2 LDAP v3 Directory	31
2.1.3.4.2.3 Business Data Connectivity (BDC) Catalog	31
2.1.3.4.3 SharePoint Products and Technologies Content Database	32
2.1.3.4.4 User Profile Service Database Indexing	32
2.1.3.4.5 Personalization and Targeting	32
2.1.3.4.6 Social Metadata	32
2.1.3.5 Search Services	33
2.1.3.5.1 Concepts	33
2.1.3.5.2 Crawling	33
2.1.3.5.3 Querying	34
2.1.3.6 Business Connectivity Services	34
2.1.3.7 Content Management Service	34
2.1.3.8 PerformancePoint Services	34
2.1.3.9 Word Automation Services	35

2.1.3.10	Session State	36
2.1.3.10.1	ASP.NET Session State.....	36
2.1.3.10.2	State Service	37
2.1.3.11	User Code Execution Service	38
2.1.3.12	Visio Graphics Service	38
2.1.3.13	Work Management.....	39
2.1.3.14	PowerPoint Automation Services	39
2.1.3.15	Translation Services	39
2.2	Protocol Summary	39
2.2.1	Core Platform Protocols.....	39
2.2.1.1	Configuration	39
2.2.1.2	Content Storage	41
2.2.2	Extended Capabilities Protocols	42
2.2.2.1	Excel Services	42
2.2.2.2	Access Services	42
2.2.2.3	InfoPath Forms Services	43
2.2.2.4	User Profile Service.....	43
2.2.2.5	Search Services.....	45
2.2.2.6	Business Connectivity Services.....	46
2.2.2.7	Content Management Service.....	47
2.2.2.8	PerformancePoint Services.....	47
2.2.2.9	Word Automation Services	48
2.2.2.10	Session State	48
2.2.2.11	User Code Execution Service	49
2.2.2.12	Visio Graphics Service	49
2.2.2.13	Work Management.....	49
2.2.2.14	PowerPoint Automation Services	50
2.2.2.15	Translation Services	50
2.3	Environment	50
2.3.1	Dependencies on This System.....	50
2.3.2	Dependencies on Other Systems/Components	50
2.4	Assumptions and Preconditions.....	51
2.5	Use Cases	51
2.5.1	Core Platform Use Cases	52
2.5.1.1	Create a File from the Client in a SharePoint Library	52
2.5.1.2	Use Claims-Based Identity to Connect Line-of-Business Applications to SharePoint Service Applications	53
2.5.2	Extended Capabilities Use Cases	55
2.5.2.1	Excel Services	55
2.5.2.1.1	Manage Requests from a Front-End Web Server.....	55
2.5.2.2	Access Services	56
2.5.2.2.1	Manage the Data Operation Request.....	57
2.5.2.3	InfoPath Forms Services	58
2.5.2.3.1	Use the State Service	58
2.5.2.4	User Profile Service.....	59
2.5.2.4.1	Retrieve a User's Profile	59
2.5.2.4.2	Tag a Document.....	60
2.5.2.5	Search Services.....	61
2.5.2.5.1	Crawl and Query	61
2.5.2.6	Session State	62
2.5.2.6.1	Communicate with a Back-End Database Server	62
2.6	Versioning, Capability Negotiation, and Extensibility.....	63
2.7	Error Handling.....	63

2.8	Coherency Requirements	63
2.9	Security.....	63
2.10	Additional Considerations.....	63
3	Examples.....	64
3.1	Example 1: Create a File from the Client.....	64
3.2	Example 2: Crawl and Query for Content	67
4	Microsoft Implementations	69
4.1	Product Behavior	69
5	Change Tracking.....	70
6	Index	73

Preliminary

1 Introduction

This document provides an informative overview of the back-end protocols that are implemented by Microsoft® SharePoint® Server 2013 Preview and Microsoft® SharePoint® Foundation 2013 Preview only, referred to collectively in this document as SharePoint Products and Technologies, for the purposes of communication between SharePoint Products and Technologies components.

For Microsoft® SharePoint 2007 Products and Technologies, Microsoft® SharePoint® 2010 Products and Technologies, Windows® SharePoint® Services 3.0 and Microsoft® SharePoint® Foundation 2010 back-end protocols please see [\[MS-SPBEPO\]](#) section 1.

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 do change significantly between SharePoint Products and Technologies versions. They are considered to be completely different protocols, and cross-compatibility between versions is not supported. In such cases new versions of the protocol documents feature the same titles; with the addition of "Version 2" or "Version 3" depending on how many previous versions of the protocol exist.

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 Version 2 [\[MS-SPFEPO2\]](#) section 2.1. 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 credential directory service (DS)**
- GUID**
- LDAP**
- Lightweight Directory Access Protocol (LDAP)**
- property set**
- security identifier (SID)**
- Server Message Block (SMB)**

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

- app**
- app instance**
- app principal**
- app product identifier**
- application server**
- ASP.NET**
- audience**
- back-end database server**
- Business Connectivity Services (BCS)**
- Business Data Connectivity (BDC)**
- configuration database**
- content database**
- data connection**
- data macro**
- data range**
- database application**
- distribution list**
- document**
- e-mail address**
- event**
- farm**
- font**
- front-end Web server**

hyperlink
Hypertext Markup Language (HTML)
line-of-business (LOB) system
list
list item
member group
partition
presentation
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
Shared Services Provider (SSP)
Simple Object Access Protocol (SOAP)
site
site collection
site subscription
stored procedure
Structured Query Language (SQL)
task
theme
Transact-Structured Query Language (T-SQL)
translation group
translation item
translation job
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
user profile
User Profile Service
user profile store
user session
Web Part
Web service
workflow

The following terms are specific to this document:

data sub-range: A subset of consecutive scale-out partition keys inside a data range. A data-sub range either starts with the same scale-out partition key or ends with the same scale-out partition key as the data range it belongs to does.

scale-out partition: A collection of data about a logical entity.

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-ADR] Microsoft Corporation, "[Access Services Data Run Time Protocol Specification](#)".

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

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

[MS-APPMDP] Microsoft Corporation, "[SharePoint App Management Database Protocol Specification](#)".

[MS-APPMWSP] Microsoft Corporation, "[SharePoint App Management Web Service Protocol Specification](#)".

[MS-ASDSP] Microsoft Corporation, "[Access Services Database Stored Procedures Protocol Specification](#)".

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

[MS-BDCDPS2] Microsoft Corporation, "[Business Data Connectivity Database Version 2 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](#)".

[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-EDSA] Microsoft Corporation, "[ediscovey Service Application Protocol Specification](#)".

[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-EXSPWS3] Microsoft Corporation, "[Excel Calculation Web Service Version 3 Protocol Specification](#)".

[MS-EXSPXML3] Microsoft Corporation, "[Excel Calculation Version 2 Web Service XML Schema](#)".

[MS-FQL2] Microsoft Corporation, "[Fast Query Language Version 2 Protocol Specification](#)".

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

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

[MS-KQL] Microsoft Corporation, "[Keyword Query Language Structure Protocol Specification](#)".

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

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

[MS-PASCWS] Microsoft Corporation, "[PowerPoint Automation Services Conversion Web Service Protocol Specification](#)".

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

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

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

[MS-SPASA] Microsoft Corporation, "[SharePoint Analytics Service Application Protocol Specification](#)".

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

[MS-SPFEPO2] Microsoft Corporation, "[SharePoint Front-End Protocols Overview Version 2](#)".

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

[MS-SPSCLSP3] Microsoft Corporation, "[SPSCrawl Stored Procedures Version 3 Protocol Specification](#)".

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

[MS-SPSETSP2] Microsoft Corporation, "[SharePoint Subscription Settings Stored Procedures Version 2 Protocol Specification](#)".

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

[MS-SPSSDBSOGP] Microsoft Corporation, "[SharePoint Shared Service Database Scale Out Generic Protocol Specification](#)".

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

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

[MS-SRCHCSOM] Microsoft Corporation, "[Search Client Query Protocol Specification](#)".

[MS-SSDPS2] Microsoft Corporation, "[Secure Store Database Version 2 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-TSSPROC] Microsoft Corporation, "[Translation Services Stored Procedures Protocol Specification](#)".

[MS-TSWS] Microsoft Corporation, "[Translation Services Web Service Protocol Specification](#)".

[MS-UPASP2] Microsoft Corporation, "[User Profile Admin Stored Procedures Version 2 Protocol Specification](#)".

[MS-UPFCWS] Microsoft Corporation, "[User Activity Feed Cache Web Service Protocol Specification](#)".

[MS-UPIESP2] Microsoft Corporation, "[User Profile Import and Export Stored Procedures Version 2 Protocol Specification](#)".

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

[MS-UPSAUD3] Microsoft Corporation, "[User Profile Service Audiences Version 3 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-UPSCHNG3] Microsoft Corporation, "[User Profile Change Log Stored Procedure Version 3 Protocol Specification](#)".

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

[MS-UPSCSP2] Microsoft Corporation, "[User Profile Service Application Caching Stored Procedures Version 2 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-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-UPSPROF3] Microsoft Corporation, "[User Profile Stored Procedures Version 3 Protocol Specification](#)".

[MS-UPSRCHSP2] Microsoft Corporation, "[User Profile Search Stored Procedures Version 2 Protocol Specification](#)".

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

[MS-UPSSDSP2] Microsoft Corporation, "[User Profile Social Data Stored Procedures Version 2 Protocol Specification](#)".

[MS-UPSSYNC3] Microsoft Corporation, "[User Profile Synchronization Stored Procedures Version 3 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-UTSP2] Microsoft Corporation, "[SharePoint Usage Tracking Stored Procedures Version 2 Protocol Specification](#)".

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

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

[MS-VSDX] Microsoft Corporation, "[Visio Graphics Service VSDX File Format](#)".

[MS-WMS] Microsoft Corporation, "[Work Management Service Web Service Protocol Specification](#)".

[MS-WORDSSP2] Microsoft Corporation, "[Word Automation Services Stored Procedures Version 2 Protocol Specification](#)".

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

[MS-WSSCADM3] Microsoft Corporation, "[Windows SharePoint Services Content Database Administrative Communications Version 3 Protocol Specification](#)".

[MS-WSSCCSP3] Microsoft Corporation, "[Windows SharePoint Services: Content Database Core List Schema and Site Provisioning Communications Version 3 Protocol Specification](#)".

[MS-WSSCFGD3] Microsoft Corporation, "[Windows SharePoint Services Configuration Database Communications Version 3 Protocol Specification](#)".

[MS-WSSDLIM3] Microsoft Corporation, "[Windows SharePoint Services: Content Database Document and List Item Management Communications Version 3 Protocol Specification](#)".

[MS-WSSDM3] Microsoft Corporation, "[Windows SharePoint Services: Content Database Data Migration Communications Version 3 Protocol Specification](#)".

[MS-WSSEUX3] Microsoft Corporation, "[Windows SharePoint Services: Content Database End-User Experience Communications Version 3 Protocol Specification](#)".

[MS-WSSFO3] Microsoft Corporation, "[Windows SharePoint Services \(WSS\): File Operations Database Communications Version 3 Protocol Specification](#)".

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

[MS-WSSPROG3] Microsoft Corporation, "[Windows SharePoint Services Content Database Programmability Extensions Communications Version 3 Protocol Specification](#)".

[MS-WSSTS] Microsoft Corporation, "[Windows SharePoint Services Technical Specification](#)".

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, <http://www.ietf.org/rfc/rfc2396.txt>

[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 [\[MS-SPFEPO2\]](#) section 1. For a technical description of SharePoint Products and Technologies concepts, see [\[MS-WSSTS\]](#).

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.

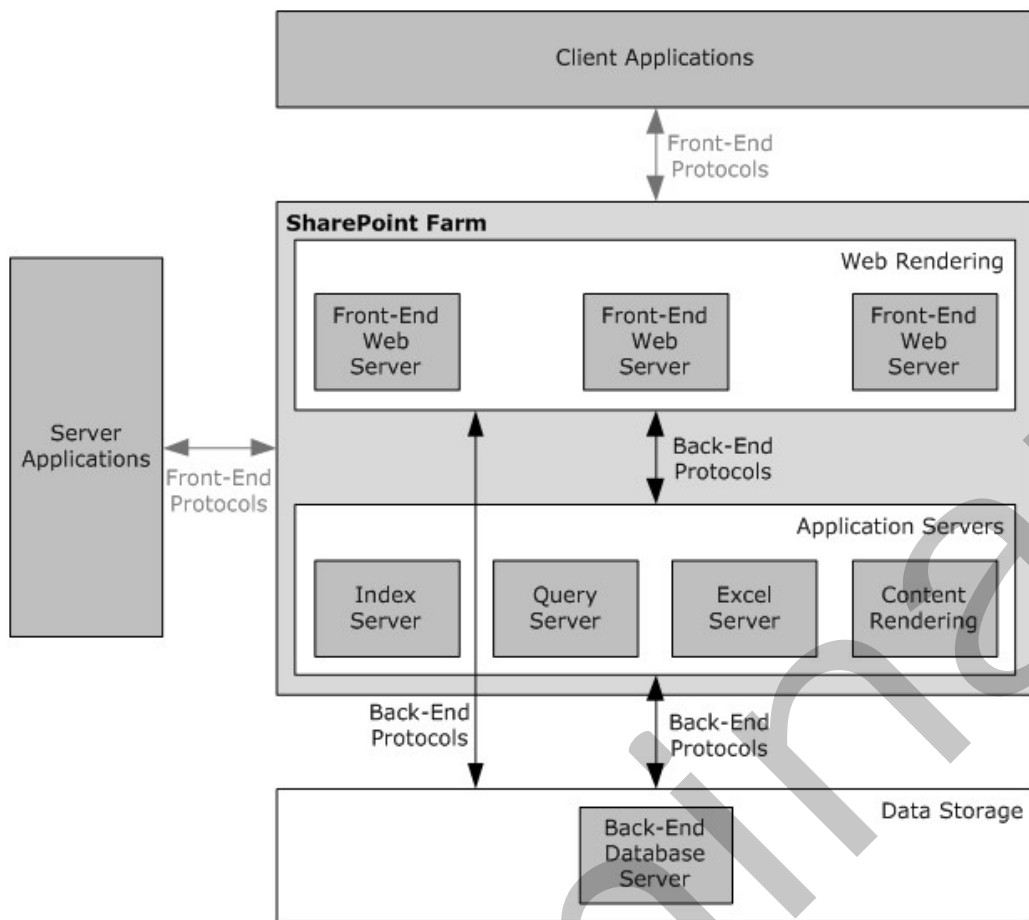


Figure 1: SharePoint Products and Technologies deployment topology

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](#).
- **Central Administration content database:** Stores content associated with the Central Administration site.
- **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 [\[MS-SPFEPO2\]](#) section 2.1.1.2.2.

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 [\[MS-SPFEPO2\]](#) section 1.

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

- Visio Graphics Service
- Work Management

- PowerPoint Automation Services
- Translation Services

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](#). Service applications 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. 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 section [2.1.2.1.1.1](#).

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.

2.1.2.1.1.1 Site Map

In addition to a description of the farm topology, the **configuration database** also stores a Site Map, which is a mapping of all site collections to the individual content databases that contain the end-user content for the site collections. The following diagram shows how those URLs can be mapped to individual content databases.

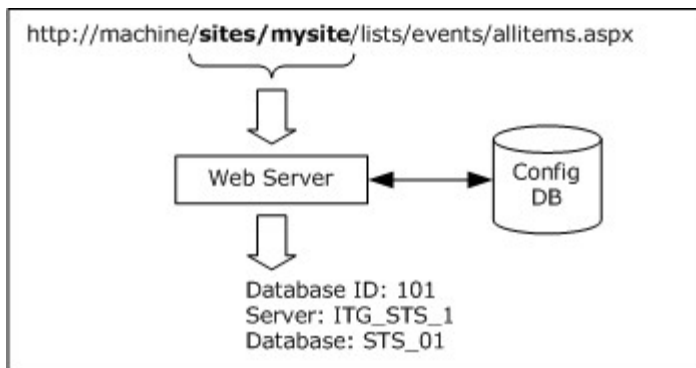


Figure 2: Determining Site Collection URL

This mapping can use a section of the URL (in this case "sites/mysite") to map to an individual back-end database server and content database. The URL is stored in a Server-Relative format, so this mapping is effective across multiple, different names for the server. The server addresses `http://machine`, `http://localhost`, or `http://157.55.234.184` are all equivalent.

The Site Collection Lookup (section [2.1.2.1.1.1.1](#)) describes the process of looking up a site collection from the site map, and determining the connection string to the content database that holds the site collection's end-user generated content.

2.1.2.1.1.1.1 Site Collection Lookup

When a request is made for content at a specific URL, the front-end Web server has to determine to which site collection the URL is referring, as well as the connection string of the content database holding the content for that site collection. This is accomplished in the following steps:

1. **Web Application Lookup:** Site collection lookup begins by examining the portion of the incoming URL beginning with the Scheme Component and ending with the Authority Component (for example, `http://example.com:80`). Scheme and Authority are defined in [RFC2396](#) sections 3.1 and 3.2. This URL is compared against a stored set of Web Application URLs. If one of the URLs in the list matches the incoming URL, the associated Web Application is used for the remainder of the operation.
2. **Prefix Matching:** Web Applications contain a set of site collection prefixes. These prefixes are URL Path Components that are used to determine which portion of the incoming URL Path Component is the Server-Relative URL of the site collection. This is done by matching all of the prefixes against the start of the Path Component of the incoming URL. If more than one prefix matches the beginning of the incoming URL Path Component, the longest matching prefix is used. A Web Application can contain any combination of the two types of prefix:

- **Explicit Prefixes:** An explicit prefix indicates that the portion of the Path Component up to and including the prefix is included in the site collection Server-Relative URL. For example, if a user requests `http://example.com/sitename/web/list/document.htm`, and if the Web Application corresponding to `http://example.com` contains an explicit prefix named "sitename", then `"/sitename"` is the Server-Relative URL of the site collection.

Incoming URL	Web Application explicit prefixes	Resulting Site Collection Server-Relative URL
<code>http://example.com/a/b/c.htm</code>	"a"	"/a"

Incoming URL	Web Application explicit prefixes	Resulting Site Collection Server-Relative URL
http://example.com/a/b/c.htm	"a", "a/b"	"/a/b"
http://example.com/a/b.htm	"a", "a/b"	"/a"
http://example.com/a/b.htm	"c"	<No Match>
http://example.com/a/b.htm	""	"/"

- Wildcard Prefixes:** A wildcard prefix indicates that the portion of the Path Component up to and including the first Path Component segment following the prefix is included in the site collection name. For example, if a user makes a request for http://example.com/sitesname/web/list/document.htm, and if the Web Application corresponding to http://example.com contains a wildcard prefix named "sites", then "/sites/sitesname" is the Server-Relative URL of the site collection.

Incoming URL	Web Application wildcard prefixes	Resulting Site Collection Server-Relative URL
http://example.com/a/b/c/d.htm	"a", "a/b"	"/a/b/c"
http://example.com/a/b.htm	"a", "a/b"	<No Match>
http://example.com/a/b.htm	""	"/a"

- Site Collection Identifier Lookup:** Once the site collection URL is determined, it is passed to the Configuration Database, along with the Web Application identifier. A site collection identifier is returned along with the identifier of the content database in which the site collection content is stored. If the specified combination of site collection URL and Web Application identifier cannot be found in the Configuration Database, the site collection does not exist. (Note that some site collections are identified not by the Path Component of the URL, but by the URL Authority Component—for example, "example.com:80". These are known as "Host Header Site Collections". If the Web Application cannot be identified from the Scheme and Authority Components of the incoming URL, site collection lookup assumes that the incoming URL refers to a Host Header site collection. In this case, the Authority Component of the incoming URL is passed to the Configuration Database which returns the corresponding site collection identifier. The site collection identifier is then passed back to the Configuration Database which returns the identifier of the content database in which the site collection content is stored. If the specified Authority Component cannot be found in the Configuration Database, the site collection does not exist.)
- Content Database Connection String Lookup:** Once the content database identifier is known, a lookup occurs to determine connection string information about the content database. The following steps occur to generate this connection string:
 - The content database identifier is passed to the configuration database, which returns the content database name and the identifier of the Database Service that is hosting the content database. If SQL Authentication is intended to be used when connecting to the content database, the connection user name and password are also returned at this time.
 - The identifier of the Database Service is then passed to the configuration database, which returns the name of the Database Service and the identifier of the Server on which the Database Service is running.

3. The identifier of the Server is passed to the configuration database which returns the address of the Server.
4. Finally, the Server address, Database Service name, content database name, and optionally, the content database user name and password are combined to build the content database Connection String.

2.1.2.1.2 Service Application Database

A service application implemented in SharePoint Products and Technologies 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 [\[MS-SPFEPO2\]](#) section 2.1.1.2.2 and [\[MSDN-SharePointSDK\]](#).

2.1.2.1.3 Usage Database

SharePoint Products and Technologies 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 [\[MS-UTSP2\]](#).

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 Protocol as described in [\[MS-TDS\]](#) 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
- 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
- Visio Graphics Service
- Work Management
- PowerPoint Automation Services
- Translation Services

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 ECS 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 [\[MS-ESPI\]](#), [\[MS-ESURL\]](#), and [\[MS-ESREST\]](#). For technical information about Excel Calculation Web Services XML Specification, see [\[MS-EXSPXML3\]](#). For technical information about the protocol implemented by Excel Calculation Web Services, see [\[MS-EXSPWS3\]](#).

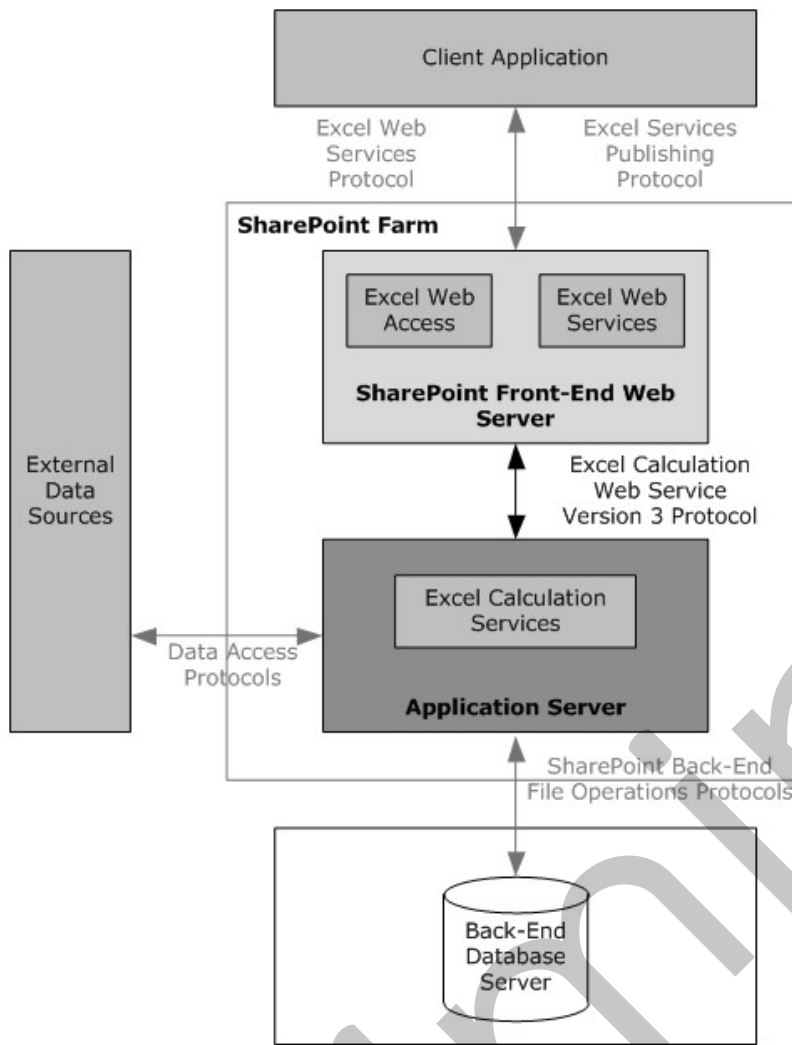


Figure 3: High-level architecture of Excel Services components

2.1.3.2 Access Services

SharePoint Products and Technologies implements Microsoft® Access Services, which is used to create database applications and manage data. For information about the Access Services database applications, see [\[MS-SPFEPO2\]](#) section 2.1.3.2.

Access Services, when used with an Access web application, 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 ([\[MS-SPFEPO2\]](#) section 2.1.3.2), which provides the Web-based rendering for Access Services database applications. The following figure illustrates this architecture.

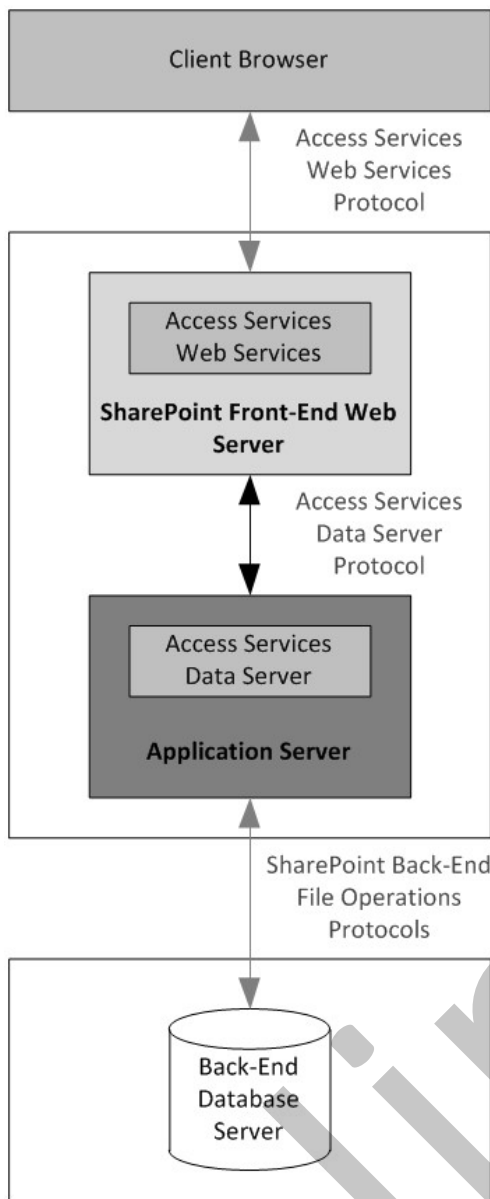


Figure 4: High-level architecture of Access Services components for an Access web app

When used with a Microsoft® Access Services Preview web app, Access Services is composed of different components. The Access Services Data Run Time (described in [\[MS-ADR\]](#)) is the back-end protocol that retrieves and manages data from a Microsoft® SQL Server® back-end database using Access Services Database Stored Procedures (described in [\[MS-ASDSP\]](#)). Access Services Web Services ([\[MS-SPFEPO2\]](#) section 2.1.3.2) provide the client-side rendering for database applications. The following figure illustrates this architecture.

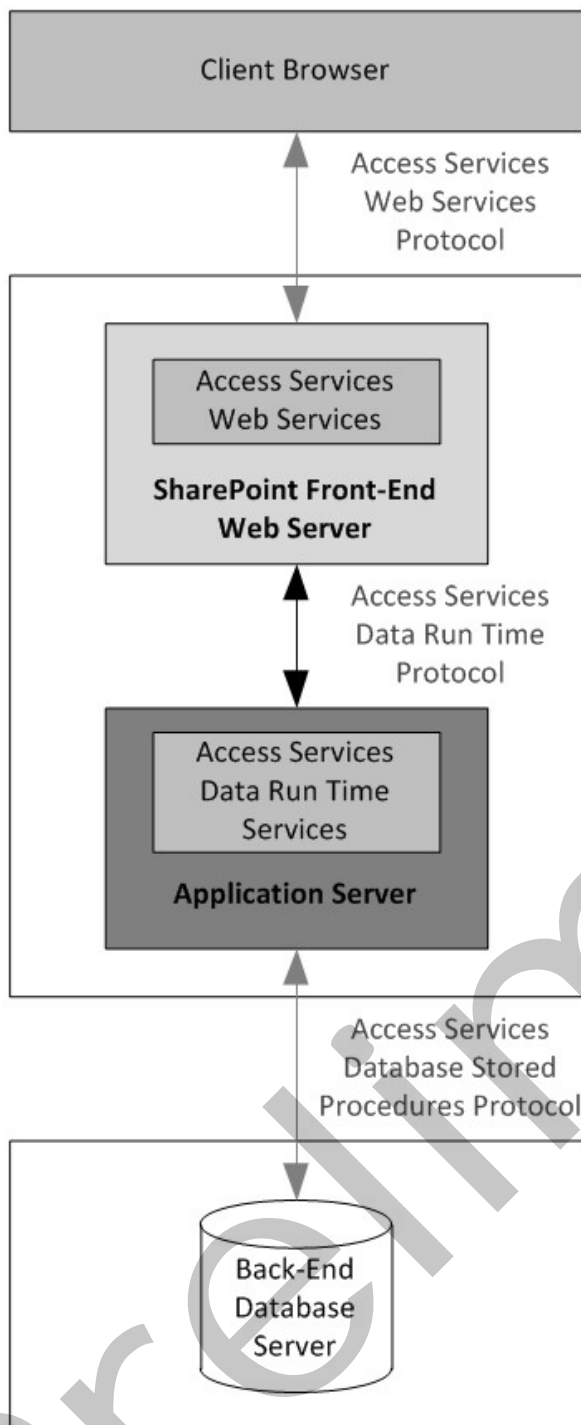


Figure 5: High-level architecture of Access Services components for a Microsoft Access Services Preview web app

2.1.3.3 InfoPath Forms Services

As described in [\[MS-SPFEPO2\]](#) section 2.1.1.3, SharePoint Products and Technologies with InfoPath® Forms Services enabled allow forms editing in the Web browser. After the form templates are published, SharePoint Products and Technologies with InfoPath Forms Services can render them for display and form submission by users.

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 or service anniversaries; or related to social activity such as applying tags to a document or a site. Groups and organizations 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 [\[MS-UPIESP2\]](#) and [\[MS-UPIEWS\]](#), 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 [\[MS-UPSCSP2\]](#), [\[MS-UPSCWS\]](#), and [\[MS-UPSPROF3\]](#).

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 and 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 [\[MS-UPSAUD3\]](#)).
- Indexing of the data for search (as described in [\[MS-SPSCRWL\]](#) and [\[MS-SPSCLSP3\]](#)).
- Redistributing the data to other SharePoint sites (as described in [\[MS-UPSSYNC3\]](#)).
- Importing and exporting data from external directory sources, such as Active Directory® Domain Services (AD DS) (as described in [\[MS-UPIESP2\]](#) and [\[MS-UPIEWS\]](#)).
- Adding social metadata to a URI or collecting information regarding metadata on a URI (as described in [\[MS-UPSSDSP2\]](#)).
- 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 [\[MS-UPSCHNG3\]](#).

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

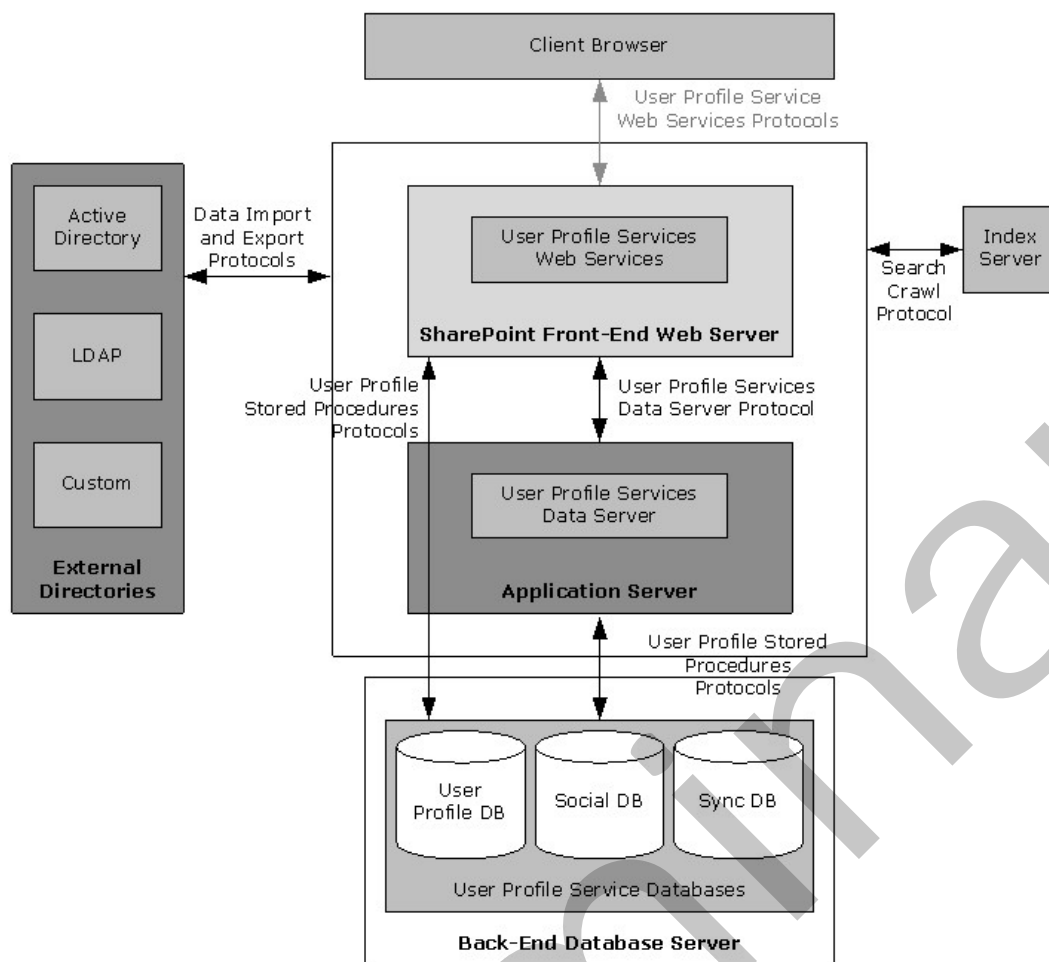


Figure 6: High-level view of the User Profile Service protocol architecture

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 [\[MS-UPSCHNG3\]](#), 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 Change Log Stored Procedure Version 3 Protocol [\[MS-UPSCHNG3\]](#).

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

- Properties

- Organizations
- Profiles
- Multiple values
- Term Sets (Vocabularies)
- Colleagues
- Memberships
- Profile change log

Social data

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

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. 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 Term Sets (Vocabularies)

Term sets 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 allow the user to select from a term set of 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".

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 Version 3 Protocol as described in [\[MS-UPSPROF3\]](#) 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 Version 3 Protocol [\[MS-UPSSYNC3\]](#), the profile store determines the list of sites that a user is a member of. The creation of distribution lists and security group memberships is accomplished by using the User Profile Import and Export Stored Procedures Protocol [\[MS-UPIESP2\]](#) and User Profile Import/Export Web Service Protocol [\[MS-UPIEWS\]](#).

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 [\[MS-UPSCHNG3\]](#)) returns the changes that occurred in the database after a given time stamp or social activity. The User Profile Change Log Stored Procedure Version 3 Protocol ([\[MS-UPSCHNG3\]](#)) is the back-end protocol that corresponds to the capabilities provided by the front-end protocol User Profile Change Log Web Service Protocol ([\[MS-USRPC\]](#)).

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

2.1.3.4.1.9 Social Data

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.

Microsoft® SharePoint® Server 2013 Preview adds support for front-end web servers saving social data posts and feed entries, and, generating consolidated feeds from an in-memory cache maintained on the Profile Service servers. The protocols used for that cache are described by [\[MS-UPFCWS\]](#) section 1.

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 [\[MS-UPIESP2\]](#) and [\[MS-UPIEWS\]](#). 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 Version 3 Protocol ([\[MS-UPSSYNC3\]](#)) 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

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. 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 are defined in [\[MS-UPIESP2\]](#) 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.

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.

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 Connectivity (BDC). The Business Data Connectivity service is part of **Business Connectivity Services (BCS)**. For information about Business Connectivity Services, see [\[MS-SPFEPO2\]](#) 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 Version 3 Protocol [\[MS-UPSSYNC3\]](#) updates 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 [\[MS-UPSCHNG3\]](#). 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 [\[MS-SPSCLSP3\]](#).

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 [\[MS-UPSAUD3\]](#). 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

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 Version 2 Protocol as described in [\[MS-UPSSDSP2\]](#) 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.

2.1.3.5.1 Concepts

Search services use two basic operating concepts: crawls and queries. Crawls involve retrieving the content of the web sites, profile repositories and other content sources by the search service, processing the content, including parsing of the documents, extracting metadata, linguistic processing (such as tokenization and stemming) and generation of the inverted index data structures that are subsequently used for fast querying. The processing steps are for illustration of a typical search engine and exact details are specific to the search service implementation. Queries involve the end user or a client application issuing queries typically containing one or more keywords, as well as other restrictions specifying the set of documents to be retrieved.

2.1.3.5.2 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 search service. 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.

Processing of a URL during the crawl involves retrieving the document data from the document repository specified by the URL, parsing the document, and determining the set of links to follow.

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.

2.1.3.5.3 Querying

A query against the search service can be executed using [\[MS-QSSWS\]](#), [\[MS-SRCHCSOM\]](#) or [\[MS-SEARCH\]](#) protocols. The query language can be similar to a user typing the query, as described in [\[MS-SEARCH\]](#) section 2.2.10, or advanced query syntax (FQL) as described in [\[MS-FQL2\]](#) section 2.

2.1.3.6 Business Connectivity Services

SharePoint Products and Technologies implements capabilities to access data from external systems, as described in [\[MS-SPFEPO2\]](#) section 2.1.3.6. 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 Tabular Data Stream Protocol ([\[MS-TDS\]](#)) as a transport protocol.

2.1.3.7 Content Management Service

As described in [\[MS-SPFEPO2\]](#) 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 Tabular Data Stream Protocol [\[MS-TDS\]](#) 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 is a collection of services in SharePoint Products and Technologies. PerformancePoint Services 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 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 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

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 [\[MS-SPFEPO2\]](#) 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 as described in [\[MS-WORDSWCF\]](#) and the Word Automation Services Stored Procedures Version 2 Protocol as described in [\[MS-WORDSSP2\]](#) (and the underlying database).

The following diagram illustrates the relationship between these two protocols.

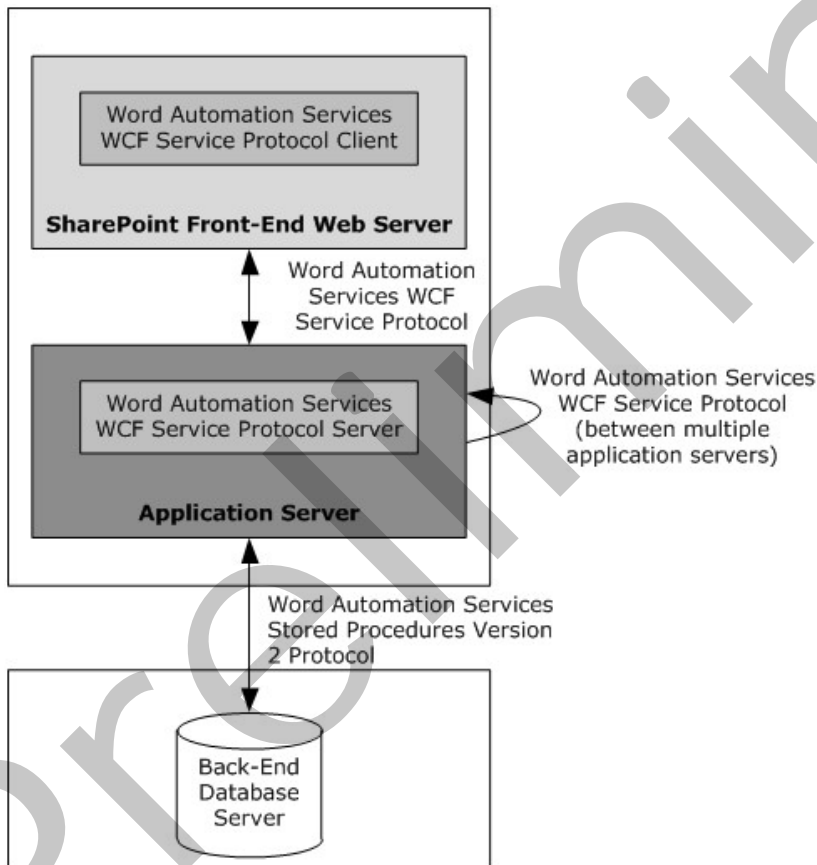


Figure 7: Relationship between the two components of Word Automation Services

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-WORDSSP2].

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**, or the State Service in SharePoint Products and Technologies 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 the **Shared Services Provider (SSP)**, which is a **Structured Query Language (SQL)** back-end database server. In SharePoint Products and Technologies, 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 [\[MSDN-IRequiresSessionState\]](#).

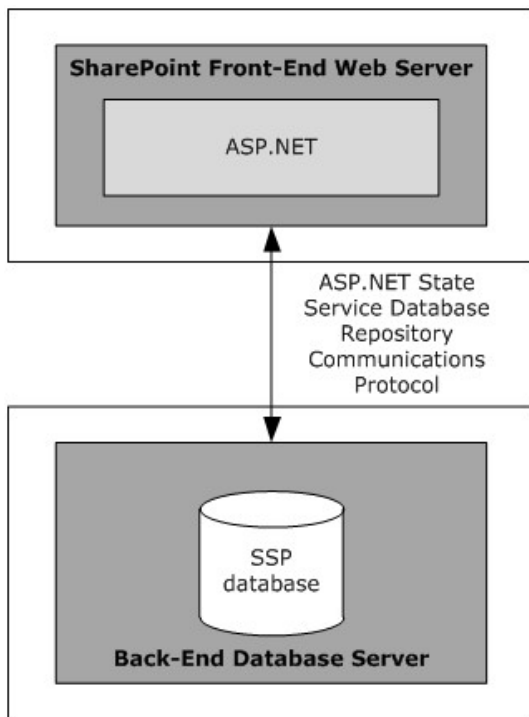


Figure 8: ASP.NET session state configuration and protocols in Office SharePoint Server

2.1.3.10.2 State Service

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

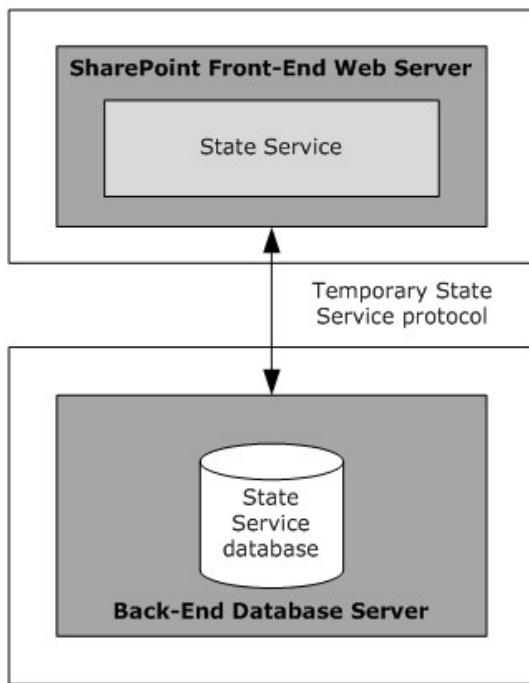


Figure 9: State Service configuration and protocols

2.1.3.11 User Code Execution Service

SharePoint Products and Technologies 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 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 Web drawing can include connections to data sources that the Visio Graphics Service can query to update information displayed in the drawing page. The Visio Graphics Service also enables comments on the drawing page to be created, viewed, and edited 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 or Microsoft® Visio® 2013 Preview, 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 business process and operational metrics for the steps in the process. The operational metrics are linked to data in a database. Users can view the Web drawing in a Web browser and see the business process overlaid with the current operational data from the database.

2.1.3.13 Work Management

Work Management aggregates tasks for the user across SharePoint sites and other systems and synchronizes them with other task systems. Microsoft® SharePoint® Server 2013 Preview uses the protocol described in [\[MS-WMS\]](#) to do perform this function.

2.1.3.14 PowerPoint Automation Services

PowerPoint Automation Services provides the capability to convert a **presentation** from one file format to another. A typical scenario for using PowerPoint Automation Services is a file conversion application that enables users to convert presentation files to a different format for archiving purposes.

PowerPoint Automation Services is composed of the PowerPoint Automation Services Conversion Web Service Protocol [\[MS-PASCWS\]](#), which allows a protocol client to pass a presentation file to the protocol server and to receive from the protocol server a converted file in a different format.

2.1.3.15 Translation Services

Microsoft® SharePoint® Server 2013 Preview implements Translation Services, which is used to automatically machine translate SharePoint content. See [\[MS-SPFEPO2\]](#) section 2.1.3.9 for more information on using Translation services to translate documents in SharePoint

Translation Services is composed of two components: the Translation Services Web Service [\[MS-TSWS\]](#) used by the front-end object model to create translation jobs, and the Translation Services Stored Procedures [\[MS-TSSPROC\]](#) used to store and retrieve data in the underlying SharePoint database.

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 protocols in the following table enable the retrieval, sharing, and storage of configuration data in Microsoft® SharePoint® Server 2013 Preview and Microsoft® SharePoint® Foundation 2013 Preview.

Protocol name	Description	Short name
Windows SharePoint Services: Configuration Database Communications Version 3 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 SharePoint Foundation 2013 Preview.	[MS-WSSCFGD3]

Protocol name	Description	Short name
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 2013 Preview.	[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 2013 Preview.	[MS-SPTWS]
SharePoint Subscription Settings Stored Procedures Version 2 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 2013 Preview.	[MS-SPSETSP2]
SharePoint Usage Tracking Stored Procedures Version 2 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 2013 Preview.	[MS-UTSP2]
SharePoint App Management Database Protocol Specification	Identifies the communications which protocol client, communicating with a database application uses to, storing and managing basic information for apps, register app principals with associated app principal permissions, generate; read and update marketplace deployment identifiers associated with site subscriptions , and maintain the relationship between app principals, app instances and app product identifiers on a given site subscription.	[MS-APPMDP]
SharePoint Shared Service Database Scale Out Generic Protocol Specification	Identifies the communications between a database management middle-tier and the database server. The communication satisfies the requests that manage the data range and scale-out partitions , requests to create and remove data sub-ranges and can expand the data range on the protocol server. These values determine the way that the protocol server responds to queries made to the scale-out partitions. The protocol client can send requests to read or write data into scale-out partitions on the protocol server by using this protocol.	[MS-SPSSDBSOGP]
SharePoint App Management Web Service Protocol Specification	Identifies the communication between the front-end Web servers and application servers. The connection information for databases implementing the SharePoint	[MS-APPMWSP]

Protocol name	Description	Short name
	App Management Database Protocol Specification [MS-APPMDP] is transferred according to this protocol to the protocol clients.	

2.2.1.2 Content Storage

The protocols in the following table enable communications for content storage in Microsoft® SharePoint® Server 2013 Preview.

Protocol name	Description	Short name
Windows SharePoint Services Content Database Administrative Communications Version 3 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 Server 2013 Preview.	[MS-WSSCADM3]
Windows SharePoint Services: Content Database Document and List Item Management Communications Version 3 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-WSSDLIM3]
Windows SharePoint Services: File Operations Database Communications Version 3 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 Server 2013 Preview.	[MS-WSSF03]
Windows SharePoint Services: Content Database Core List Schema and Site Provisioning Communications Version 3 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-WSSCCSP3]
Windows SharePoint Services: Content Database End-User Experience Communications Version 3 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-WSSSEUX3]
Windows SharePoint Services: Content Database Programmability Extensions Communications Version 3 Protocol Specification	Identifies the communication between the front-end Web server and the back-end-database server that is used to satisfy requests involving apps , events (2) , Web Parts , and workflow (2) within SharePoint Server 2013 Preview.	[MS-WSSPROG3]
Windows SharePoint Services: Content Database Data Migration Communications Version 3 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	[MS-WSSDM3]

Protocol name	Description	Short name
	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.	

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 protocols in the following table enable access to Excel® Services workbooks in Microsoft® SharePoint® Server 2013 Preview.

Protocol name	Description	Short name
Excel Calculation Version 2 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-EXSPXML3]
Excel Calculation Web Service Version 3 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] . Applies to SharePoint Server 2013 Preview.	[MS-EXSPWS3]

2.2.2.2 Access Services

The protocols in the following table enable Microsoft® Access Services in Microsoft® SharePoint® Server 2013 Preview.

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. Used with an Access web application.	[MS-ADS]
Access Services Database Stored Procedures Protocol Specification	This protocol defines tables, table-valued functions, and stored procedures that create and maintain the metadata for Web-based database applications. Protocol clients can add and remove information about the application and about objects in the application. A typical scenario for using this protocol is the addition of new metadata to a set of well-defined tables about a database application, such as when an object is added to the application. Used with an Microsoft® Access Services Preview web app.	[MS-ASDSP]
Access Services Data Run Time	This protocol is used for managing data maintained by a protocol server, which provides access to a protocol client that renders the data	[MS-ADR]

Protocol name	Description	Short name
Protocol Specification	provided by the protocol server. The communication is always initiated by the protocol client. A typical scenario for this protocol is opening a result set and updating data. Used with an Access Services Preview web app.	

2.2.2.3 InfoPath Forms Services

InfoPath Forms Services Protocols in Microsoft® SharePoint® Server 2013 Preview are covered in [\[MS-SPFEPO2\]](#) section 2.2.2.3.

InfoPath Forms Services uses the State Service 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® Server 2013 Preview.

Protocol name	Description	Short name
User Profile Service Audiences Version 3 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 SharePoint Server 2013 Preview.	[MS-UPSAUD3]
User Profile Change Log Stored Procedure Version 3 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 2013 Preview.	[MS-UPSCHNG3]
User Profile Stored Procedures Version 3 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 2013 Preview.	[MS-UPSPROF3]
User Profile Synchronization Stored Procedures Version 3 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 3 Protocol [MS-WSSFO3] 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 2013 Preview.	[MS-UPSSYNC3]

Protocol name	Description	Short name
User Profile Social Data Stored Procedures Version 2 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 2013 Preview.	[MS-UPSSDSP2]
User Profile Search Stored Procedures Version 2 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 SharePoint Server 2013 Preview.	[MS-UPSRCHSP2]
User Profile Admin Stored Procedures Version 2 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 2013 Preview.	[MS-UPASP2]
User Profile Import and Export Stored Procedures Version 2 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 2013 Preview.	[MS-UPIESP2]
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-UPIEWS]
User Profile Service Application Caching Stored Procedures Version 2 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 2013 Preview.	[MS-UPSCSP2]
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 (2) 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.	[MS-UPWCFWS]

Protocol name	Description	Short name
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.	[MS-UPSCWS]
User Profile Synchronization (UPS): Configuration Data Structure	Identifies the data structures used in configuring the User Profile Synchronization service.	[MS-UPSCDS]
User Profile Synchronization (UPS): Configuration Database Data Access Protocol Specification	Identifies the communication between the User Profile Synchronization Configuration Service acting as a client and the User Profile Synchronization Configuration Database acting as a server.	[MS-UPSCDAP]
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]
User Activity Feed Cache Web Service Protocol	Identifies the communication that allows protocol clients to store data related to social feed posts and feed metadata in an in-memory cache on a middle-tier application server.	[MS-UPFCWS]

2.2.2.5 Search Services

The protocols in the following table enable search services in Microsoft® SharePoint® Server 2013 Preview.

Protocol name	Description	Short name
Fast Query Language Version 2 Protocol Specification	Specifies the advanced query language to execute search queries	[MS-FQL2]
Keyword Query Language Structure Protocol Specification	This document specifies the structure of the Keyword Query Language (KQL). KQL is a language for expressing search criteria.	[MS-KQL]
Search Query Shared Service Protocol Specification	Allows protocol clients to execute queries, retrieve search results, and record clickthroughs on search results. This is a WCF based protocol.	[MS-QSSWS]
Site Data Web Service Protocol Specification	Allows the protocol client to retrieve all items in the web site including all metadata. It supposed full and incremental indexing.	[MS-SITEDATS]
SPSCrawl Stored Procedures Version 3 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 2013 Preview.	[MS-SPSCLSP3]
Search Lotus Notes Web Service Protocol Specification	Enables the protocol client to enumerate and retrieve content and properties from a Lotus Notes Domino Server.	[MS-NOTESWS]
Search Protocol Specification	Allows protocol clients to execute queries against the search service. This query protocol is based on SOAP.	[MS-SEARCH]
Search Client Query Protocol Specification	Allows the protocol clients to execute queries against the search service. This query protocol is based on CSOM, and is designed for clients outside the search system.	[MS-SRCHCSOM]

2.2.2.6 Business Connectivity Services

The protocols in the following table enable information about interfaces and line-of-business (LOB) systems in Microsoft® SharePoint® Server 2013 Preview.

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]
Business Data Connectivity Database Version 2 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-BDCDPS2]
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 Version 2	Provides secure access to a database of encrypted user credentials. Enterprises have a variety of data stored in various line-of-business	[MS-SSDPS2]

Protocol name	Description	Short name
Protocol Specification	(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.	
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 Specification	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 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.	[MS-BDCMFFS]

2.2.2.7 Content Management Service

The protocols in the following table enable remote import in Microsoft® SharePoint® Server 2013 Preview.

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]
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]
eDiscovery Service Application Protocol Specification	Enables a protocol client to manage actions that can be applied to the supported sources of content. Actions can be invoked and deleted. The protocol also provides the means to get the status of the actions that have already been sent to the protocol server. A typical scenario for using this protocol is an application used to centrally manage the different sources of content.	[MS-EDSA]
SharePoint Analytics Service Application Protocol Specification	This protocol enables a protocol client to send usage events to the protocol server. The protocol server performs implementation specific aggregation of those events and produces historical usage information based on those events. This protocol further enables a protocol client to retrieve the aggregated usage information from the protocol server. A typical scenario for using this protocol is a reporting application that allows the users to visualize the usage of an item during its lifetime.	[MS-SPASA]

2.2.2.8 PerformancePoint Services

The protocols in the following table enable communications for PerformancePoint Services in Microsoft® SharePoint® Server 2013 Preview.

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 2013 Preview.

Protocol name	Description	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 Version 2 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-WORDSSP2]

2.2.2.10 Session State

The protocol in the following table enables stored procedures for the ASP.Net Session State Service in Microsoft® SharePoint® Server 2013 Preview.

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 2013 Preview.

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 Service

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 2013 Preview.

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 Visio Graphics Service

The protocol in the following table enables the Visio Graphics Service in Microsoft® SharePoint® Server 2013 Preview.

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 formats in the following table enable the Visio Graphics Service in SharePoint Server 2013 Preview.

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

2.2.2.13 Work Management

The protocol in the following table is used by Work Management features in Microsoft® SharePoint® Server 2013 Preview.

Protocol name	Description	Short name
Work Management Service Web Service Protocol Specification	Allows protocol clients to refresh or synchronize a user's task (3) information through a middle-tier application server. The protocol client sends a request for a user to the protocol server, and the server responds with an acknowledgement of the request and refreshes the task (3) information asynchronously.	[MS-WMS]

2.2.2.14 PowerPoint Automation Services

The protocol in the following table enables presentation file conversion in Microsoft® SharePoint® Server 2013 Preview.

Protocol name	Description	Short name
PowerPoint Automation Services Conversion Web Service Protocol Specification	Allows a protocol client to convert a presentation from one file format to another file format. It allows the protocol client to pass a presentation file to the protocol server and to receive from the protocol server a converted file in a different format.	[MS-PASCWS]

2.2.2.15 Translation Services

The protocols in the following table are used by Translation Services in Microsoft® SharePoint® Server 2013 Preview.

Protocol name	Description	Short name
Translation Services Web Service Protocol Specification	This protocol enables a protocol client to create and manage a persistent queue of document translations. It also allows the protocol client to initiate the machine translation of documents in that queue from one language to another.	[MS-TSWS]
Translation Services Stored Procedures Protocol	This protocol allows clients to add, modify, and delete translation jobs , translation groups and translation items from a database on the protocol server, as well as retrieve those translation jobs, translation groups and translation items by using predefined criteria.	[MS-TSSPROC]

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: [\[MS-SPFEPO2\]](#)

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
- wss File, Print, User/Group Administration: [\[MS-WSSO\]](#) section 1

Active Directory® Domain Services (AD DS): [\[MS-ADTS\]](#) section 1

Microsoft® SharePoint® Server 2013 Preview depends on the following systems/components to function:

- Windows Server 2008 R2 Sp1
- .NET Framework 4.0
- Windows PowerShell 2.0
- SQL Server 2008 R2 SP1

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.
- 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 2013 Preview 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 3 Protocol [MS-WSSFO3]. 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-SPFEPO2] section 2.5.1.5.

The example in section 3.1 also provides more details.

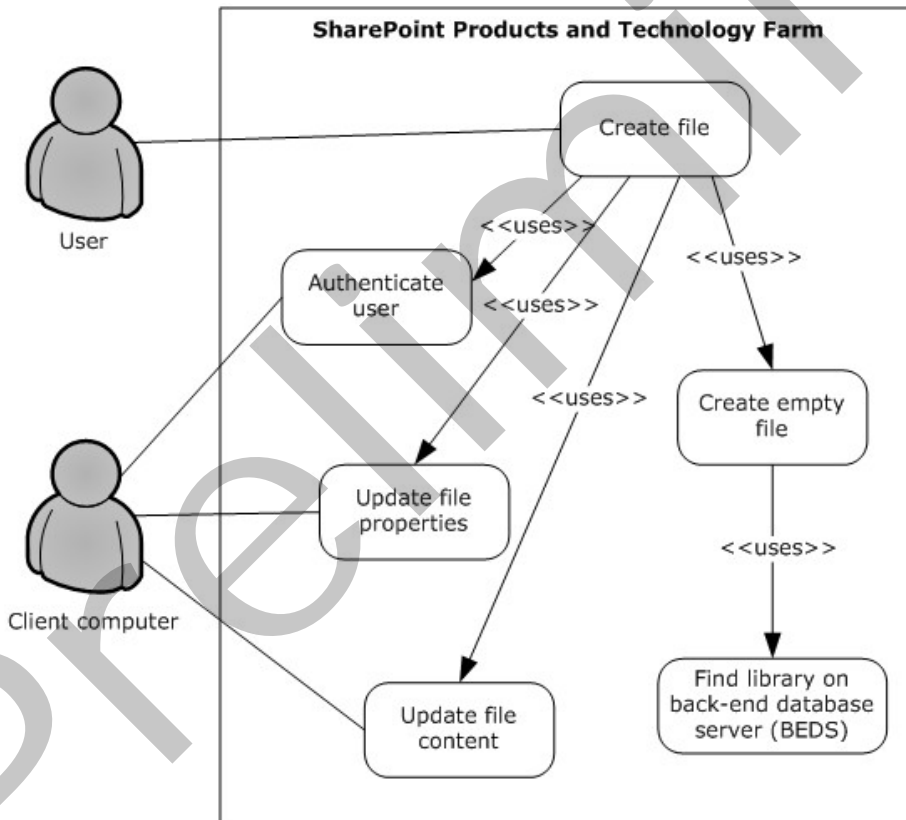


Figure 10: Process for creating a file from the client in a SharePoint library

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 SharePoint Foundation 2013 Preview authenticates the user.
3. SharePoint Foundation 2013 Preview finds the location of the document library and verifies that the user has access to it.
4. SharePoint Foundation 2013 Preview 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.

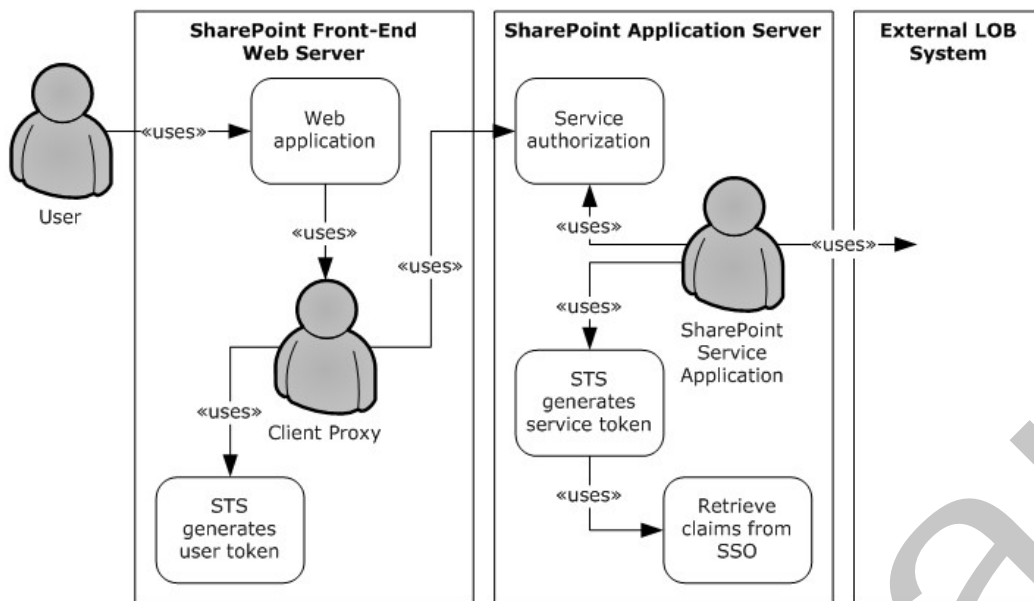


Figure 11: Process for using claims-based identity to connect line-of-business applications to SharePoint Service applications

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 service token 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.

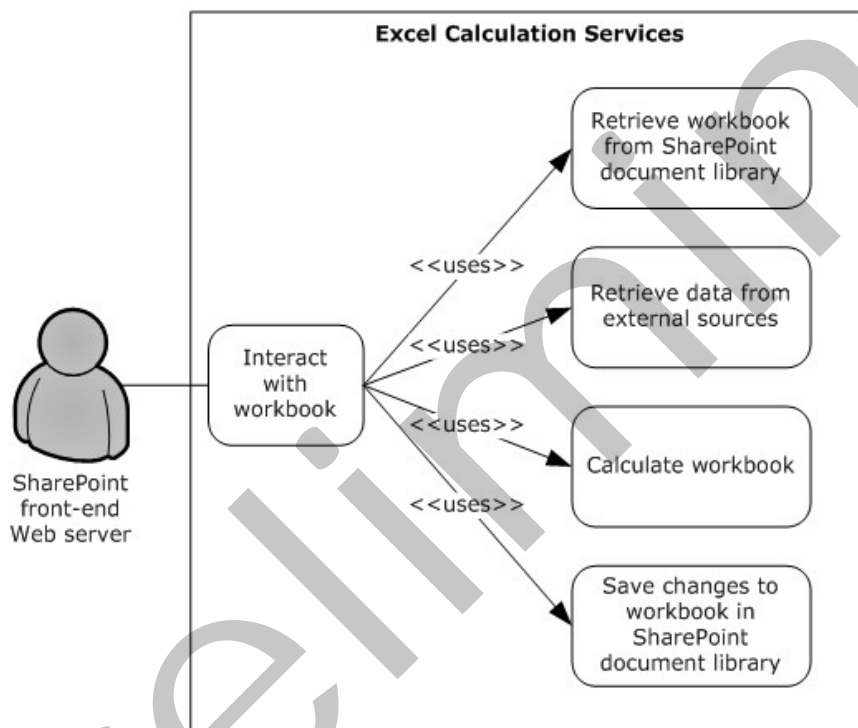


Figure 12: Process for managing requests from a front-end Web server

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

Microsoft® Access Services supports two different kinds of applications, an Access web application and an Microsoft® Access Services Preview web app. The following section describes how Access Services manages data operations with an Access web application.

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 Access web application published to a SharePoint library.

The following diagram illustrates this process.

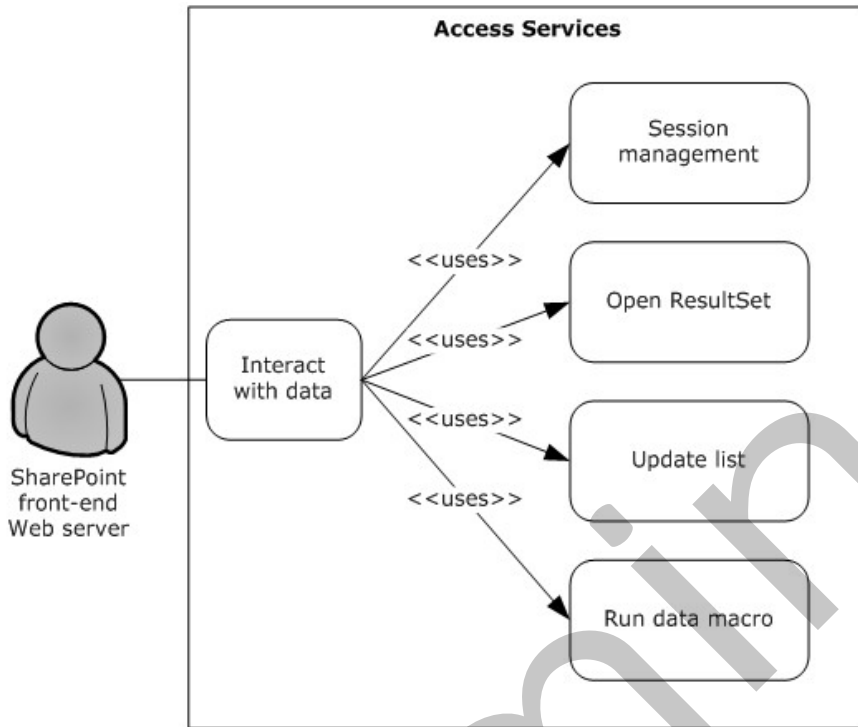


Figure 13: Process for managing the data operation request

Preconditions

- Access Services is enabled on the SharePoint Products and Technologies farm.
- A user interacts with the Access web 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 2013 Preview. This use case complements the use case, "Publish a Form Template and Enable Browser Rendering," in [MS-SPFEPO2] 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.

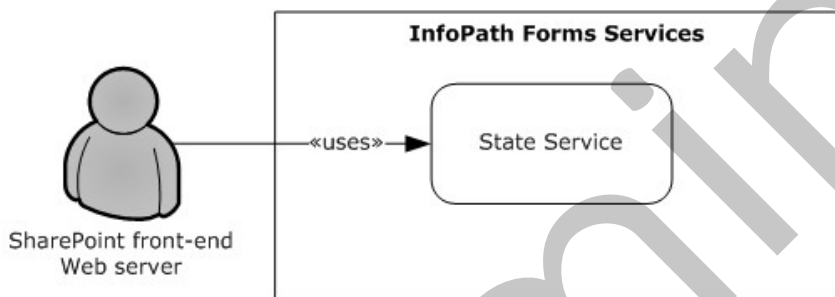


Figure 14: Process for using the State Service

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 Without Error

- 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.

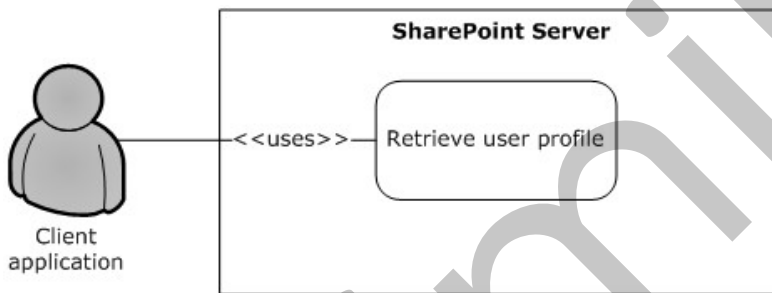


Figure 15: Process for retrieving a user's profile

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.

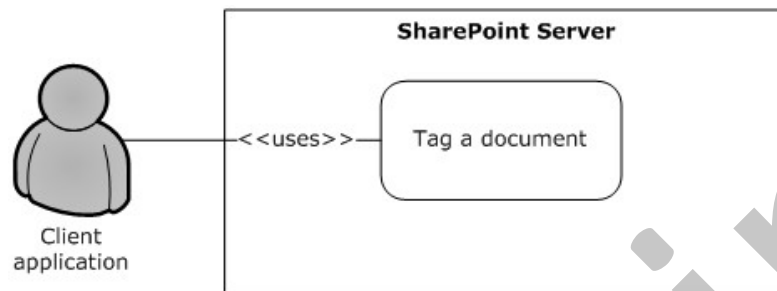


Figure 16: Process for tagging a document

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 Crawl and Query

This use case describes how a SharePoint Products and Technologies Search service can crawl a site (2) and a user subsequently queries a particular search service application. Once the crawl is finished, the most common scenario is for the SharePoint Products and Technologies Search service engine to build a full-text index to use for queries.

The following diagram illustrates this process.

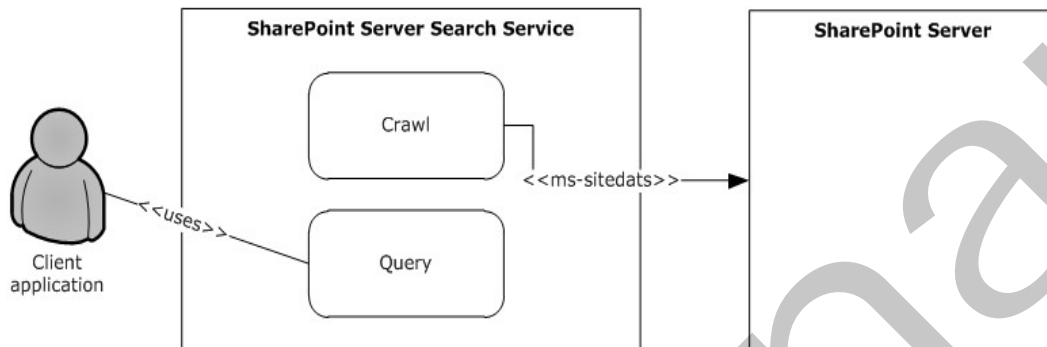


Figure 17: Process for crawling and querying

Precondition

- The client application needs to have the rights to be able to access one of the web service described by [\[MS-SITEDATS\]](#).
- The search service needs to have access to the SharePoint server site data web service.

Steps

1. The search service initiates a crawl, fetches the content from the web site and indexes it.
2. The client application issues queries against the search service to get results across all instances of available SharePoint servers.

Errors

- If the client does not have the rights to the search service application, an access denied error can occur.
- If a client issues the query using incorrect syntax a query error can occur.
- If a client does not have permissions to read items that matched the query, it can get 0 results.

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.

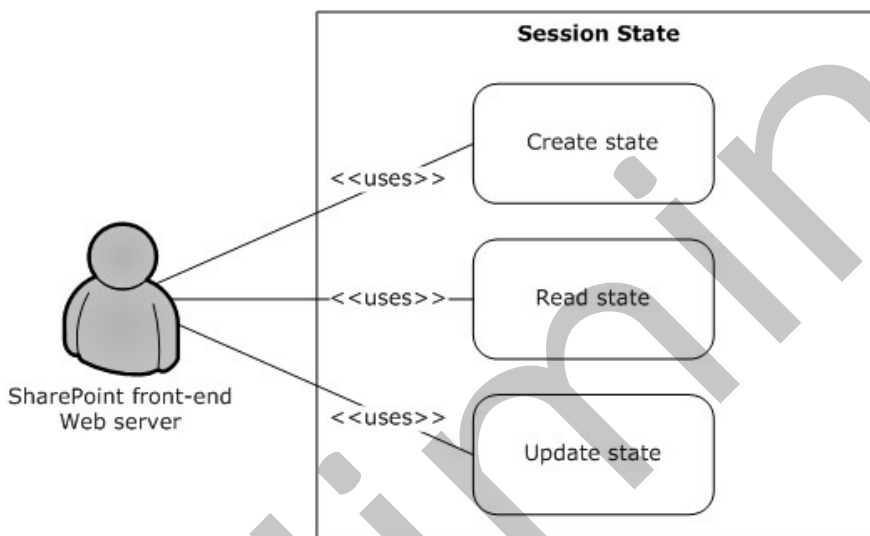


Figure 18: Process for communicating with a back-end database server

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.

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 3 Protocol [\[MS-WSSFO3\]](#), 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 [\[MS-WSSFO3\]](#) 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 [\[MS-WSSFO3\]](#) section 4.2 to provide more details on what happens when a user has not visited the SharePoint Products and Technologies site before. The [\[MS-WSSFO3\]](#) 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 described 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.

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 [\[MS-WSSFO3\]](#) section 4.2 for Microsoft® SharePoint® Server 2013 Preview.

1. 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.
2. 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 [\[MS-WSSFO3\]](#) section 4.6 for SharePoint Server 2013 Preview.

1. 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.
2. 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 [\[MS-WSSFO3\]](#) section 4.9 for SharePoint Server 2013 Preview.

1. The front-end Web server returns a WebDAV response, saying the file was created successfully.
2. 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.
3. In multiple round trips with the back-end database server, the front-end Web server retrieves the file.
4. 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 [\[MS-WSSFO3\]](#) section 4.1 for SharePoint Server 2013 Preview.

1. In response to the HEAD request, the front-end Web server sends a response saying the request was successful.
2. 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.
3. The client sends a WebDAV request to the front-end Web server with an XML payload that has the file properties from the client.

4. 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".
5. 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.
6. The back-end database server returns multiple result sets as part of the process to update the files.
7. 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.

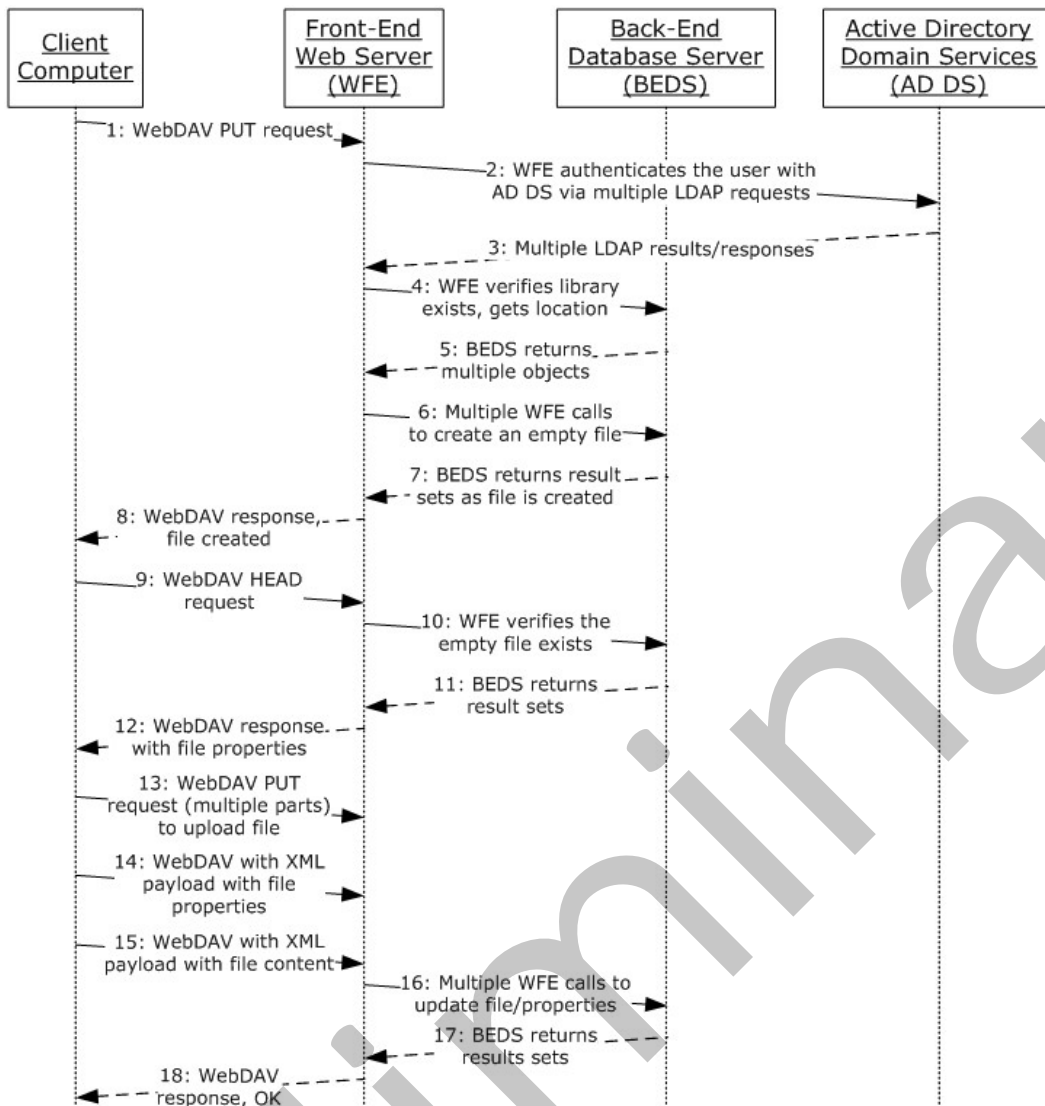


Figure 19: Sequence for creating a file from the client

3.2 Example 2: Crawl and Query for Content

This example describes how a search service runs a crawl operation, and how the query client queries for search results. This example uses the search services use cases in section [2.5.2.5](#).

This scenario assumes the following:

- The user has configured the search service to crawl and index the content of the relevant web sites.
- 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 the search service running a crawl and indexing the web site content, and the user querying content, assuming that no errors occur.

1. The search service initiates a crawl of the user profile store and enumerates user profile content using the messages provided in [\[MS-SPSCLSP3\]](#) section 3.2.5.
2. The search service starts traversing the URL space.
3. For SharePoint Products and Technologies content, the crawler application communicates with the SiteData Web service and retrieves site collection information by calling GETSITE.
4. 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 [\[MS-SITEDATS\]](#) section 4.1, [\[MS-SITEDATS\]](#) section 4.2, and [\[MS-SITEDATS\]](#) section 4.3.

Step 5 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.

- After the search service has finished the crawl, 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 [\[MS-QSSWS\]](#) section 3.1.4.1.2.1.

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:

- Microsoft® SharePoint® Foundation 2013 Preview
- Microsoft® SharePoint® Server 2013 Preview

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

This section identifies changes that were made to the [MS-SPBEPO2] protocol document between the April 2012 and July 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.1.1.4 Back-End Protocols	Updated list of back-end protocols included by extended capabilities.	N	Content updated.
2.1.2.1.1.1.1 Site Collection Lookup	Added reference [RFC2396] to list of informative references in section 1.2.	N	Content updated.
2.1.3.2 Access Services	Updated the captions for the two figures in this section.	N	Content updated.
2.1.3.2 Access Services	Updated service names in Figure 5.	N	Content updated.
2.1.3.4 User Profile Service	Updated introduction and caption of figure for User Profile Service.	N	Content updated.
2.1.3.4.1 Core Service	Revised the list of key pieces of data stored in the User Profile Service database, for clarity.	N	Content removed.
2.1.3.6 Business Connectivity Services	Updated "TDS" to "Tabular Data Stream Protocol" for clarity.	N	Content updated.
2.1.3.7 Content Management Service	Updated "TDS" to "Tabular Data Stream Protocol" for clarity.	N	Content updated.
2.1.3.10.1 ASP.NET Session State	Updated caption of diagram.	N	Content updated.
2.1.3.10.1	Updated label for protocol name in	N	Content

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
ASP.NET Session State	diagram.		updated.
2.1.3.10.1 ASP.NET Session State	Added glossary term and description for "Shared Services Provider (SSP)".	N	Content updated.
2.2.2.4 User Profile Service	Added definition number for term "site".	N	Content updated.
2.5.1.2 Use Claims-Based Identity to Connect Line-of-Business Applications to SharePoint Service Applications	Clarified wording in step 4 of use case process.	N	Content updated.
2.5.2.3.1 Use the State Service	Updated "Post Condition" to "Post Condition Without Error" for clarity.	N	Content updated.
3.2 Example 2: Crawl and Query for Content	Updated the section reference for MS-SPSCLSP3.	N	Content updated.

6 Index

A

[Abstract](#) 1
Access Services overview ([section 2.1.3.2](#) 24, [section 2.2.2.2](#) 42)
[Additional considerations](#) 63
[Applicable protocols](#) 39
[Architecture](#) 15
[ASP.NET session state overview](#) 36
[Assumptions](#) 51

B

[Back-end protocols overview](#) 17
Business Connectivity Services overview ([section 2.1.3.6](#) 34, [section 2.2.2.6](#) 46)

C

[Capability negotiation](#) 63
[Change tracking](#) 70
[Coherency requirements](#) 63
Communicate with a back-end database server
 [overview](#) 62
[Communications](#) 50
 [with other systems](#) 50
 [within the system](#) 50
[Component dependencies](#) 50
[Concepts](#) 15
[Configuration overview](#) 39
[Configuration protocols overview](#) 19
Configure a crawl
 run a crawl
 and query for content
 [details](#) 67
Considerations
 [additional](#) 63
 [security](#) 63
Content Management Service overview ([section 2.1.3.7](#) 34, [section 2.2.2.7](#) 47)
Content storage overview ([section 2.1.2.2](#) 22, [section 2.2.1.2](#) 41)
Core platform protocols
 [Configuration](#) 39
 [configuration protocols](#) 19
 content storage ([section 2.1.2.2](#) 22, [section 2.2.1.2](#) 41)
Core platform protocols overview ([section 2.1.2](#) 19, [section 2.2.1](#) 39)
Create a file from the client
 [details](#) 64
Create a file from the client in a SharePoint library
 [overview](#) 52
Create and crawl a content source
 [overview](#) 61

D

[Data storage overview](#) 17
Dependencies
 [with other systems](#) 50
 [within the system](#) 50
Deployment topology
 [back-end protocols](#) 17
 [data storage](#) 17
 [front-end protocols](#) 17
 [overview](#) 15
 [SharePoint farm](#) 16
Design intent
 [communicate with a back-end database server](#) 62
 [create a file from the client in a SharePoint library](#) 52
 [create and crawl a content source](#) 61
 [manage requests from a front-end Web server](#) 55
 [manage the data operation request](#) 57
 [overview](#) 51
 [retrieve a user's profile](#) 59
 [tag a document](#) 60
 [use claims-based identity to connect line-of-business applications to SharePoint Service applications](#) 53
 [use the State Service](#) 58

E

[Environment](#) 50
[Error handling](#) 63
Examples
 configure a crawl
 run a crawl
 [and query for content](#) 67
 [create a file from the client](#) 64
 [overview](#) 64
 [query for content](#) 67
 [run a crawl](#) 67
Excel Services overview ([section 2.1.3.1](#) 23, [section 2.2.2.1](#) 42)
Extended capabilities protocols
 Access Services ([section 2.1.3.2](#) 24, [section 2.2.2.2](#) 42)
 Business Connectivity Services ([section 2.1.3.6](#) 34, [section 2.2.2.6](#) 46)
 Content Management Service ([section 2.1.3.7](#) 34, [section 2.2.2.7](#) 47)
 Excel Services ([section 2.1.3.1](#) 23, [section 2.2.2.1](#) 42)
 InfoPath Forms Services ([section 2.1.3.3](#) 27, [section 2.2.2.3](#) 43)
 PerformancePoint Services ([section 2.1.3.8](#) 34, [section 2.2.2.8](#) 47)
 PowerPoint Automation Services ([section 2.1.3.14](#) 39, [section 2.2.2.14](#) 50)
 search services ([section 2.1.3.5](#) 33, [section 2.2.2.5](#) 45)
 session state ([section 2.1.3.10](#) 36, [section 2.2.2.10](#) 48)

Translation Services ([section 2.1.3.15](#) 39, [section 2.2.2.15](#) 50)
User Code Execution Service ([section 2.1.3.11](#) 38, [section 2.2.2.11](#) 49)
User Profile Service ([section 2.1.3.4](#) 27, [section 2.2.2.4](#) 43)
Visio Graphics Service ([section 2.1.3.12](#) 38, [section 2.2.2.12](#) 49)
Word Automation Services ([section 2.1.3.9](#) 35, [section 2.2.2.9](#) 48)
Work Management ([section 2.1.3.13](#) 39, [section 2.2.2.13](#) 49)
Extended capabilities protocols overview ([section 2.1.3](#) 22, [section 2.2.2](#) 42)
Extended Capabilities Use Cases
[Access Services](#) 56
[User Profile Service](#) 59
Extensibility
[Microsoft implementations](#) 69
[overview](#) 63
[External dependencies](#) 50

F

Farm
[SharePoint overview](#) 16
[Front-end protocols overview](#) 17
[Functional architecture](#) 15
[Functional requirements - overview](#) 15

G

[Glossary](#) 8

H

[Handling requirements](#) 63

I

[Implementations - Microsoft](#) 69
[Implementer - security considerations](#) 63
InfoPath Forms Services overview ([section 2.1.3.3](#) 27, [section 2.2.2.3](#) 43)
[Informative references](#) 9
[Initial state](#) 51
[Introduction](#) 7

M

Manage requests from a front-end Web server
[overview](#) 55
Manage the data operation request
[overview](#) 57
[Microsoft implementations](#) 69

O

Overview
[abstract](#) 1
Access Services ([section 2.1.3.2](#) 24, [section 2.2.2.2](#) 42)

[ASP.NET session state](#) 36
[back-end protocols](#) 17
Business Connectivity Services ([section 2.1.3.6](#) 34, [section 2.2.2.6](#) 46)
[Configuration](#) 39
[configuration protocols](#) 19
Content Management Service ([section 2.1.3.7](#) 34, [section 2.2.2.7](#) 47)
content storage ([section 2.1.2.2](#) 22, [section 2.2.1.2](#) 41)
core platform protocols ([section 2.1.2](#) 19, [section 2.2.1](#) 39)
[data storage](#) 17
[deployment topology](#) 15
[examples](#) 64
Excel Services ([section 2.1.3.1](#) 23, [section 2.2.2.1](#) 42)
extended capabilities protocols ([section 2.1.3](#) 22, [section 2.2.2](#) 42)
[front-end protocols](#) 17
InfoPath Forms Services ([section 2.1.3.3](#) 27, [section 2.2.2.3](#) 43)
PerformancePoint Services ([section 2.1.3.8](#) 34, [section 2.2.2.8](#) 47)
PowerPoint Automation Services ([section 2.1.3.14](#) 39, [section 2.2.2.14](#) 50)
search services ([section 2.1.3.5](#) 33, [section 2.2.2.5](#) 45)
[session state](#) 36
[Session State Services](#) 48
[SharePoint farm](#) 16
[State Service](#) 37
[summary of protocols](#) 39
[synopsis](#) 15
Translation Services ([section 2.1.3.15](#) 39, [section 2.2.2.15](#) 50)
User Code Execution Service ([section 2.1.3.11](#) 38, [section 2.2.2.11](#) 49)
User Profile Service ([section 2.1.3.4](#) 27, [section 2.2.2.4](#) 43)
[User Profile Services – audiences](#) 32
[User Profile Services – content database](#) 32
[User Profile Services – core service](#) 28
[User Profile Services – data import and export](#) 30
[User Profile Services – indexing the profile service data](#) 32
[User Profile Services – personalization and targeting](#) 32
[User Profile Services – social metadata](#) 32
Visio Graphics Service ([section 2.1.3.12](#) 38, [section 2.2.2.12](#) 49)
[Word Automation Services](#) 35
[Work Automation Services](#) 48
Work Management ([section 2.1.3.13](#) 39, [section 2.2.2.13](#) 49)
Overview – Use cases
[Access Services](#) 56
[User Profile Service](#) 59

P

PerformancePoint Services overview ([section 2.1.3.8](#) 34, [section 2.2.2.8](#) 47)
PowerPoint Automation Services overview ([section 2.1.3.14](#) 39, [section 2.2.2.14](#) 50)
[Preconditions](#) 51

Q

Query for content
[details](#) 67

R

[References](#) 9
Requirements
 [coherency](#) 63
 [error handling](#) 63
 [overview](#) 15
 [preconditions](#) 51
Retrieve a user's profile
 [overview](#) 59
Run a crawl
 [details](#) 67

S

Search services overview ([section 2.1.3.5](#) 33, [section 2.2.2.5](#) 45)
[Security considerations](#) 63
Session state overview ([section 2.1.3.10](#) 36, [section 2.2.2.10](#) 48)
[SharePoint farm overview](#) 16
[State Service overview](#) 37
[System architecture](#) 15
[System dependencies](#) 50
 [with other systems](#) 50
 [within the system](#) 50
[System errors](#) 63
System overview
 [abstract](#) 1
 [introduction](#) 7
[System protocols](#) 39
[System requirements - overview](#) 15
System use cases
 [communicate with a back-end database server](#) 62
 [create a file from the client in a SharePoint library](#) 52
 [create and crawl a content source](#) 61
 [manage requests from a front-end Web server](#) 55
 [manage the data operation request](#) 57
 [overview](#) 51
 [retrieve a user's profile](#) 59
 [tag a document](#) 60
 [use Claims-based identity to connect line-of-business applications to SharePoint Service applications](#) 53
 [use the State Service](#) 58

T

[Table of protocols](#) 39
Tag a document

[overview](#) 60
[Tracking changes](#) 70
Translation Services overview ([section 2.1.3.15](#) 39, [section 2.2.2.15](#) 50)

U

[Use cases](#) 51
 [Access Services overview](#) 56
 [communicate with a back-end database server](#) 62
 [create a file from the client in a SharePoint library](#) 52
 [create and crawl a content source](#) 61
 [manage requests from a front-end Web server](#) 55
 [manage the data operation request](#) 57
 [retrieve a user's profile](#) 59
 [tag a document](#) 60
 [use claims-based identity to connect line-of-business applications to SharePoint Service applications](#) 53
 [use the State Service](#) 58
 [User Profile Service overview](#) 59
Use claims-based identity to connect line-of-business applications to SharePoint Service applications
 [overview](#) 53
Use the State Service
 [overview](#) 58
User Code Execution Service overview ([section 2.1.3.11](#) 38, [section 2.2.2.11](#) 49)
User Profile Service overview ([section 2.1.3.4](#) 27, [section 2.2.2.4](#) 43)

V

Versioning
 [Microsoft implementations](#) 69
 [overview](#) 63
Visio Graphics Service overview ([section 2.1.3.12](#) 38, [section 2.2.2.12](#) 49)

W

Word Automation Services overview ([section 2.1.3.9](#) 35, [section 2.2.2.9](#) 48)
Work Management overview ([section 2.1.3.13](#) 39, [section 2.2.2.13](#) 49)