

# [MS-SPFEPO]: SharePoint Front-End Protocols Overview

---

## Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

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

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

## Abstract

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

## Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.01	Major	Initial Availability
06/27/2008	1.0	Editorial	Revised and edited technical content
10/06/2008	1.01	Editorial	Revised and edited technical content
12/12/2008	1.02	Editorial	Revised and edited technical content
07/13/2009	1.03	Major	Revised and edited the technical content
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	Major	Updated and revised the technical content
02/19/2010	2.0	Major	Updated and revised the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.05	Minor	Clarified the meaning of the technical content.
09/27/2010	2.06	Editorial	Changed language and formatting in the technical content.
11/15/2010	2.06	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.06	No change	No changes to the meaning, language, or formatting of the technical content.

<b>Date</b>	<b>Revision History</b>	<b>Revision Class</b>	<b>Comments</b>
03/18/2011	2.06	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.06	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	3.0	Major	Significantly changed the technical content.
04/11/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1 Introduction</b>	<b>7</b>
1.1 Glossary	8
1.2 References	10
<b>2 Functional Architecture</b>	<b>16</b>
2.1 Overview	16
2.1.1 Deployment Topology	16
2.1.1.1 SharePoint Products and Technologies Farm	17
2.1.1.2 Web Infrastructure	18
2.1.1.2.1 Web Topology	18
2.1.1.2.2 Service Applications	19
2.1.1.2.3 Web Components	20
2.1.1.2.4 Web Parts	20
2.1.1.3 Data Storage	21
2.1.1.3.1 Physical Storage	21
2.1.1.3.2 Logical Storage	21
2.1.1.4 Back-End Protocols	22
2.1.1.5 Front-End Protocols	22
2.1.2 Core Platform	23
2.1.2.1 Site Administration	23
2.1.2.2 List Data	24
2.1.2.3 File Operations	24
2.1.2.4 Security and Identity	24
2.1.2.4.1 Authentication	24
2.1.2.4.2 Authorization	25
2.1.2.5 Authoring Integration	25
2.1.2.6 Extensibility	25
2.1.2.7 SharePoint Client-Side Object Model	25
2.1.3 Extended Capabilities	26
2.1.3.1 Excel Services	26
2.1.3.2 Access Services	27
2.1.3.3 InfoPath Forms Services	27
2.1.3.4 User Profile Service	28
2.1.3.5 Search Services	28
2.1.3.6 Business Connectivity Services	29
2.1.3.7 Content Management Service	29
2.1.3.8 PerformancePoint Services	30
2.1.3.9 Analytics Services	30
2.1.3.10 Office Online Web Query Service	31
2.2 Protocol Summary	31
2.2.1 Core Platform Protocols	31
2.2.1.1 Site Administration	31
2.2.1.2 List Data	32
2.2.1.3 File Operations	34
2.2.1.4 Security and Identity	35
2.2.1.5 Authoring Integration	35
2.2.1.6 Extensibility	37
2.2.1.6.1 Content Import and Export	37
2.2.1.6.2 Workflow Integration	37
2.2.1.6.3 Web Page and Web Part Customization	38

2.2.1.7	SharePoint Client-Side Object Model .....	38
2.2.2	Extended Capabilities Protocols .....	38
2.2.2.1	Excel Services .....	38
2.2.2.2	Access Services .....	39
2.2.2.3	InfoPath Forms Services .....	40
2.2.2.4	User Profile Service .....	41
2.2.2.5	Search Services .....	41
2.2.2.6	Business Connectivity Services .....	42
2.2.2.7	Content Management Service .....	42
2.2.2.8	PerformancePoint Services .....	43
2.2.2.9	Analytics Services .....	44
2.2.2.10	Office Online Web Query Service .....	44
2.3	Environment .....	44
2.3.1	Dependencies on This System .....	44
2.3.2	Dependencies on Other Systems/Components .....	44
2.4	Assumptions and Preconditions .....	45
2.5	Use Cases .....	46
2.5.1	Core Platform Use Cases .....	47
2.5.1.1	Create a Site Collection .....	47
2.5.1.2	Create a Site .....	48
2.5.1.3	Create a List and Add a View .....	48
2.5.1.4	Get List Data .....	49
2.5.1.5	Create a File in a SharePoint Library from a Client Computer .....	50
2.5.1.6	Export and Import Site Content .....	52
2.5.1.7	Add a Web Part to a Page and Update Its Properties .....	53
2.5.2	Extended Capabilities Use Cases .....	54
2.5.2.1	Excel Services .....	54
2.5.2.1.1	Publish a Workbook to a SharePoint Library .....	54
2.5.2.1.2	Interact with a Workbook by Using a Browser .....	55
2.5.2.1.3	Interact with a Workbook by Using an Application .....	56
2.5.2.2	Access Services .....	58
2.5.2.2.1	Publish a Database Application to SharePoint Products and Technologies....	58
2.5.2.2.2	Interact with a Database Application that Has Been Published to SharePoint Products and Technologies .....	59
2.5.2.3	InfoPath Forms Services .....	60
2.5.2.3.1	Publish a Form Template and Enable Browser Rendering .....	60
2.5.2.4	User Profile Service .....	62
2.5.2.4.1	Retrieve a User's Profile .....	62
2.5.2.4.2	Tag a Document .....	63
2.5.2.5	Search Services .....	64
2.5.2.5.1	Issue a Search Query .....	64
2.5.2.6	Business Connectivity Services .....	65
2.5.2.6.1	Create an External Content Type .....	65
2.5.2.6.2	Read an External Content Type .....	66
2.5.2.6.3	Update an External Content Type .....	67
2.5.2.6.4	Deploy a Business Connectivity Services Solution .....	68
2.5.2.6.5	Update a Business Connectivity Services Solution .....	69
2.5.2.7	Content Management Service .....	70
2.5.2.7.1	Get Term Set Data and Add a Term .....	70
2.5.2.8	PerformancePoint Services .....	71
2.5.2.8.1	Interact with Published Dashboards on a Web Part Page .....	71
2.5.2.9	Analytics Services .....	72
2.5.2.9.1	Log an Event and Retrieve Data .....	72

2.6	Versioning, Capability Negotiation, and Extensibility .....	73
2.7	Error Handling .....	73
2.8	Coherency Requirements .....	73
2.9	Security.....	73
2.10	Additional Considerations .....	73
<b>3</b>	<b>Examples.....</b>	<b>74</b>
3.1	Example 1: Create a New Site and Access List Data .....	74
3.2	Example 2: Create a File from a Client Computer.....	75
<b>4</b>	<b>Microsoft Implementations .....</b>	<b>79</b>
4.1	Product Behavior .....	79
<b>5</b>	<b>Change Tracking.....</b>	<b>80</b>
<b>6</b>	<b>Index .....</b>	<b>81</b>

# 1 Introduction

This document provides an informative overview of the front-end protocols that are implemented by Microsoft® Office SharePoint® Server 2007, Windows® SharePoint® Services 3.0, Microsoft® SharePoint® Server 2010, and Microsoft® SharePoint® Foundation 2010, referred to collectively in this document as SharePoint Products and Technologies, for the purpose of communicating with client and server applications.

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 the client and server applications and the front-end Web servers for SharePoint Products and Technologies. Where appropriate, the document also describes the relationship between the protocols and the example scenarios in which they are used. The 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 [\[MS-WSSTS\]](#). For an overview of basic SharePoint Products and Technologies file, print, user/group administration, see [\[MS-WSSO\]](#).

Some of the **Web services** and file-format protocols have changed significantly between Office SharePoint Server 2007 and SharePoint Server 2010. The new protocols are considered completely different than their predecessors, and cross-compatibility between versions is not supported. New versions of protocol documents that target SharePoint Server 2010 feature the same titles as their predecessors, with the addition of "Version 2." This overview document covers both the original and the "Version 2" protocols, and, where appropriate, mentions the appropriate protocol name or document short name.

In addition to the front-end protocols that are described in this document, SharePoint Products and Technologies implements a number of back-end protocols, as described in [\[MS-SPBEPO\]](#). The front-end protocols are implemented at a higher level in the product architecture than the back-end protocols. Because the front-end protocols use logic within SharePoint Products and Technologies to enforce appropriate rules and data consistency, they are less complex to implement and are thus preferred for most interoperability scenarios.

## 1.1 Glossary

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

- authentication**
- authentication mode**
- authorization**
- directory service (DS)**
- GUID**
- Hypertext Transfer Protocol (HTTP)**
- principal**
- server**
- share**
- XML**

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

- Active Directory Service Interfaces (ADSI)**
- alert**
- alert subscription**
- application server**
- ASP.NET**
- Asynchronous JavaScript + XML (AJAX)**
- atom feed**
- audience**
- back-end database server**
- blog**
- browser-enable**
- browser-enabled form template**
- Business Connectivity Services (BCS)**
- Business Data Connectivity (BDC)**
- calculate**
- Central Administration site**
- change log**
- Collaborative Application Markup Language (CAML)**
- column**
- content database**
- content source**
- crawl**
- crawler**
- dashboard**
- data connection**
- data macro**
- decomposition tree**
- deployment package**
- distribution list**
- document**
- document library**
- Document Workspace site**
- EntityInstance**
- external data**
- external list**
- farm**
- file**
- form**



**form digest validation**  
**form file**  
**form library**  
**form server**  
**form template**  
**form template (.xsn) file**  
**forms authentication**  
**front-end Web server**  
**group**  
**hierarchy**  
**home page**  
**HTTP GET**  
**Hypertext Markup Language (HTML)**  
**identity**  
**index server**  
**Internet Information Services (IIS)**  
**inverted index**  
**item**  
**line-of-business (LOB) system**  
**list**  
**list item**  
**Meeting Workspace site**  
**membership**  
**metadata index**  
**metadata store**  
**ODBC**  
**OLAP**  
**OLE DB**  
**Online Analytical Processing (OLAP)**  
**Open Database Connectivity (ODBC)**  
**permission**  
**picture library**  
**property promotion**  
**scorecard**  
**search query**  
**search service application**  
**Secure Store Service (SSS)**  
**Security Assertion Markup Language (SAML)**  
**service application**  
**Shared Services Provider (SSP)**  
**Simple Object Access Protocol (SOAP)**  
**site**  
**site collection**  
**site definition**  
**site membership**  
**site template**  
**Slide Library**  
**social data**  
**social networking**  
**social rating**  
**social tag**  
**solution gallery**  
**solution package**  
**TCP/IP**  
**term**

**term set**  
**term store**  
**Uniform Resource Identifier (URI)**  
**Uniform Resource Locator (URL)**  
**Universal Data Connection (.udc, .udcx) file**  
**user profile**  
**User Profile Service**  
**user profile store**  
**view**  
**Web application**  
**Web control**  
**Web Distributed Authoring and Versioning Protocol (WebDAV)**  
**Web Part**  
**Web server**  
**Web service**  
**Web site**  
**workbook**  
**workflow**

## 1.2 References

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

We conduct frequent surveys of the informative references to assure their continued availability. If you have any issue with finding an informative reference, please contact [dochelp@microsoft.com](mailto: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-ACCDT] Microsoft Corporation, "[Access Template File Format Specification](#)".

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

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

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

[MS-ALERTSS] Microsoft Corporation, "[Alerts Service Protocol Specification](#)".

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

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

[MS-AXL] Microsoft Corporation, "[Access Application Transfer Protocol Structure Specification](#)".

[MS-BCSDPFFS] Microsoft Corporation, "[Business Connectivity Services Deployment Package File Format Specification](#)".

[MS-BCSDWPS] Microsoft Corporation, "[Business Connectivity Services Deployment Web Service Protocol Specification](#)".

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

[MS-BDCMP] Microsoft Corporation, "[Business Data Catalog Metadata Web Service Protocol Specification](#)".

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

[MS-CONATB] Microsoft Corporation, "[Content Area Toolbox Web Service Protocol Specification](#)".

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

[MS-CPSWS] Microsoft Corporation, "[SharePoint Claim Provider Web Service Protocol Specification](#)".

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

[MS-CSOMSPT] Microsoft Corporation, "[SharePoint Client-Side Object Model Protocol Specification](#)".

[MSDN-ASPWP] Microsoft Corporation, "ASP.NET Web Parts Overview", <http://msdn.microsoft.com/en-us/library/hhy9ewf1.aspx>

[MSDN-EXCELS] Microsoft Corporation, "Introducing Excel Services", <http://msdn.microsoft.com/en-us/library/ms519581.aspx>

[MSDN-IIS] Microsoft Corporation, "Internet Information Services", <http://msdn.microsoft.com/en-us/library/aa286507.aspx>

[MSDN-MOSS2007-SDK] Microsoft Corporation, "Microsoft Office SharePoint Server 2007 SDK", <http://msdn.microsoft.com/en-us/library/bb931736.aspx>

[MSDN-MOSSWF] Microsoft Corporation, "Workflows in Office SharePoint Server 2007", <http://msdn.microsoft.com/en-us/library/ms549489.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-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)

[MSDN-WSSWF] Microsoft Corporation, "Workflow Development for Windows SharePoint Services", <http://msdn.microsoft.com/en-us/library/ms414613.aspx>

[MSDN-WSSWP] Microsoft Corporation, "Web Part in Windows SharePoint Services", <http://msdn.microsoft.com/en-us/library/ms476318.aspx>

[MS-DOCTRANS] Microsoft Corporation, "[Document Transforms Service Protocol Specification](#)".

[MS-DSPSTSS] Microsoft Corporation, "[Data-Source Adapter SharePoint Team Services Web Service Protocol Specification](#)".

[MS-DWSS] Microsoft Corporation, "[Document Workspace Web Service Protocol Specification](#)".

[MS-EAWF] Microsoft Corporation, "[Group Approval Web Service Protocol Specification](#)".

[MS-ECTPWPS] Microsoft Corporation, "[External Content Type Picker Web Service Protocol Specification](#)".

[MS-EMMWC] Microsoft Corporation, "[Metadata Shared Service WCF Service Protocol Specification](#)".

[MS-EMMWS] Microsoft Corporation, "[Microsoft Enterprise Managed Metadata Web Service Protocol 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-FORMS] Microsoft Corporation, "[Forms Service Protocol Specification](#)".

[MS-FPSE] Microsoft Corporation, "[FrontPage Server Extensions Remote Protocol Specification](#)".

[MS-FSDAP] Microsoft Corporation, "[Forms Services Design and Activation Web Service Protocol Specification](#)".

[MS-FSFD] Microsoft Corporation, "[Forms Services Feature Detection Protocol Specification](#)".

[MS-FSPP] Microsoft Corporation, "[Forms Services Proxy Web Service Protocol Specification](#)".

[MS-FSSHHTTP] Microsoft Corporation, "[File Synchronization via SOAP over HTTP Protocol Specification](#)".

[MS-FSSHHTTPB] Microsoft Corporation, "[Binary Requests for File Synchronization via SOAP Protocol Specification](#)".

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

[MS-IMAGS] Microsoft Corporation, "[Imaging Service Protocol Specification](#)".

[MS-INFODCF] Microsoft Corporation, "[InfoPath Data Connection File Download Protocol Specification](#)".

[MS-IPDSP] Microsoft Corporation, "[InfoPath Digital Signing Protocol](#)".

[MS-IPFF] Microsoft Corporation, "[InfoPath Form Template Format](#)".

[MS-IPFF2] Microsoft Corporation, "[InfoPath Form Template Format Version 2](#)".

[MS-IPFFX] Microsoft Corporation, "[InfoPath Form File Format Specification](#)".

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

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

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

[MS-METAWEB] Microsoft Corporation, "[MetaWeblog Extensions Protocol Specification](#)".

[MS-OCPROTO] Microsoft Corporation, "[Office Client Protocols Overview](#)".

[MS-OFBA] Microsoft Corporation, "[Office Forms Based Authentication Protocol Specification](#)".

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

[MS-OFFICIALFILE] Microsoft Corporation, "[Official File Web Service Protocol Specification](#)".

[MS-OMS] Microsoft Corporation, "[Office Mobile Service Protocol Specification](#)".

[MS-OOWQWS] Microsoft Corporation, "[Office Online Web Query Web Service Protocol Specification](#)".

[MS-OSALER] Microsoft Corporation, "[Alerts Interoperability Protocol Specification](#)".

[MS-OSCO] Microsoft Corporation, "[Office Server ClickOnce Manifest Structure Specification](#)".

[MS-OUTSPS] Microsoft Corporation, "[Lists Client Sync Protocol Specification](#)".

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

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

[MS-PLSP] Microsoft Corporation, "[Published Links Web Service Protocol Specification](#)".

[MS-PPSAS] Microsoft Corporation, "[PerformancePoint Services AuthoringService Protocol Specification](#)".

[MS-PPSDECO] Microsoft Corporation, "[PerformancePoint Services Decomposition Tree Protocol Specification](#)".

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

[MS-PRSTFR] Microsoft Corporation, "[ADO XML Persistence Format Protocol Specification](#)".

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

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

[MS-SHDACCWS] Microsoft Corporation, "[Shared Access Web Service Protocol Specification](#)".

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

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

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

[MS-SLIDELI] Microsoft Corporation, "[Slide Library Web Service Protocol Specification](#)".

[MS-SLXV] Microsoft Corporation, "Silverlight XAML Vocabulary Specification 2008", October 2008, [http://msdn.microsoft.com/en-us/library/dd361850\(v=PROT.10\).aspx](http://msdn.microsoft.com/en-us/library/dd361850(v=PROT.10).aspx)

[MS-SPAFWI] Microsoft Corporation, "[SharePoint Activity Feed Web Interfaces Protocol Specification](#)".

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

[MS-SPDIAG] Microsoft Corporation, "[SharePoint Diagnostics Web Service Protocol Specification](#)".

[MS-SPEMAWS] Microsoft Corporation, "[SharePoint E-Mail Web Service Protocol Specification](#)".

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

[MS-SPRSS] Microsoft Corporation, "[RSS 2.0 Format Extensions Specification](#)".

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

[MS-SPSTWS] Microsoft Corporation, "[SharePoint Security Token Service Web Service Protocol Specification](#)".

[MS-SSAS] Microsoft Corporation, "[SQL Server Analysis Services Protocol Specification](#)".

[MS-STSSYN] Microsoft Corporation, "[StsSync Data Structure](#)".

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

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

[MS-TMPLDISC] Microsoft Corporation, "[Template Discovery Web Service Protocol Specification](#)".

[MS-UDCX] Microsoft Corporation, "[Universal Data Connection 2.0 XML File Format](#)".

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

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

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

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

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

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

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

[MS-WDV] Microsoft Corporation, "[Web Distributed Authoring and Versioning \(WebDAV\) Protocol: Client Extensions](#)".

[MS-WDVME] Microsoft Corporation, "[Web Distributed Authoring and Versioning \(WebDAV\) Protocol: Microsoft Extensions](#)".

[MS-WDVMODUUJ] Microsoft Corporation, "[Office Document Update Utility Extensions Protocol Specification](#)".

[MS-WDVSE] Microsoft Corporation, "[Web Distributed Authoring and Versioning \(WebDAV\) Protocol: Server Extensions](#)".

[MS-WEBDAVE] Microsoft Corporation, "[Web Distributed Authoring and Versioning Error Extensions Protocol Specification](#)".

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

[MS-WPPS] Microsoft Corporation, "[Web Part Pages Web Service Protocol Specification](#)".

[MS-WSSCAML] Microsoft Corporation, "[Collaborative Application Markup Language \(CAML\) Structure Specification](#)".

[MS-WSSCAP] Microsoft Corporation, "[Windows SharePoint Services Collaborative Application Protocol Specification](#)".

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

- [MS-WSSHP] Microsoft Corporation, "[HTTP Windows SharePoint Services Headers Protocol Specification](#)".
- [MS-WSSO] Microsoft Corporation, "[Windows SharePoint Services Overview](#)".
- [MS-WSSREST] Microsoft Corporation, "[ListData Data Service Protocol Specification](#)".
- [MS-WSSTS] Microsoft Corporation, "[Windows SharePoint Services Technical Specification](#)".
- [MS-WWSP] Microsoft Corporation, "[Workflow Web Service Protocol Specification](#)".
- [RFC2518] Goland, Y., Whitehead, E., Faizi, A., Carter, S., and Jensen, D., "HTTP Extensions for Distributed Authoring WebDAV", RFC 2518, February 1999, <http://www.ietf.org/rfc/rfc2518.txt>
- [RFC2616] Fielding, R., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.ietf.org/rfc/rfc2818.txt>
- [SAMLToken1.1] Lawrence, K., Kaler, C., Monzillo, R., et al., "Web Services Security: SAML Token Profile 1.1", February 2006, <http://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTokenProfile.pdf>
- [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>
- [WSFederation] Kaler, C., Nadalin, A., Bajaj, S., et al., "Web Services Federation Language (WS-Federation)", Version 1.1, December 2006, <http://specs.xmlsoap.org/ws/2006/12/federation/ws-federation.pdf>
- [WSTrust] IBM, Microsoft, Nortel, VeriSign, "WS-Trust V1.0", December 2005, <http://www.oasis-open.org/committees/download.php/15980/oasis-wssx-ws-trust-1.0.pdf>

## 2 Functional Architecture

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

### 2.1 Overview

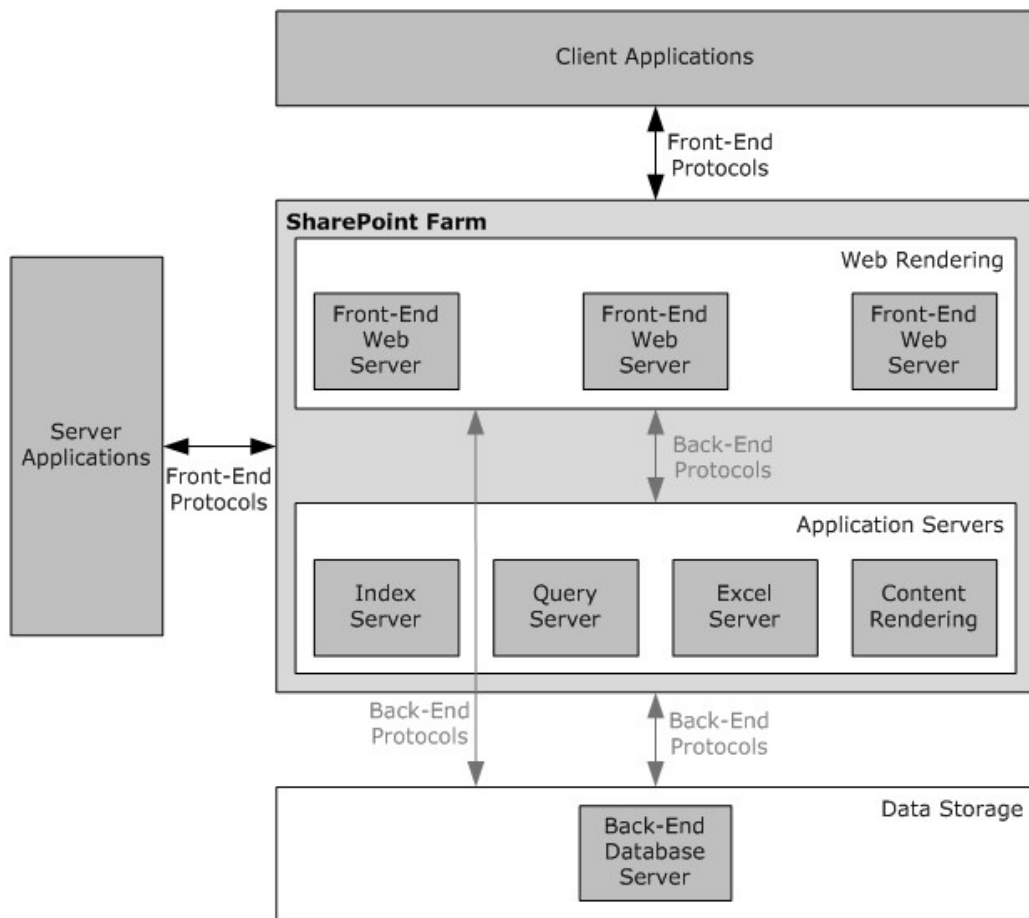
SharePoint Products and Technologies comprises a Web-based platform that provides both an infrastructure for building and deploying applications and a set of built-in capabilities and features. The following sections describe the high-level topology, architecture, and deployment of SharePoint Products and Technologies as they relate to the front-end protocols.

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

#### 2.1.1 Deployment Topology

SharePoint Products and Technologies can be deployed in a wide range of topologies to provide high availability and various degrees of scale-out, depending on the requirements of particular deployment scenarios. A particular deployment of one or more **servers (2)** is called a farm. The following figure shows the protocol topology for SharePoint Products and Technologies. This figure shows the components deployed on separate servers (2), but the components can also run compatibly on a single server (2).





**Figure 1: SharePoint Products and Technologies deployment topology**

Servers (2) of each type can be added to a deployment of SharePoint Products and Technologies as needed by the specific usage scenario. These additional servers (2), such as front-end Web servers, supply additional computing resources to 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 Web Infrastructure

SharePoint Products and Technologies provides the ability to create and manage **Web sites (2)**. It does so within the framework of two operating-system components: **Internet Information Services (IIS)** and **ASP.NET**, which provide core services for HTTP, process management, and programmability. These components exist within the Windows Server® 2003 operating system with Service Pack 1 and The Windows Server® 2008 operating system with Service Pack 2 (SP2). For more information, see [\[MSDN-IIS\]](#).

#### 2.1.1.2.1 Web Topology

In IIS, a **Web application (1)** acts as both a container for Web sites (2) and a process isolation boundary. A deployment of SharePoint Products and Technologies includes three types of Web applications (1) in IIS:

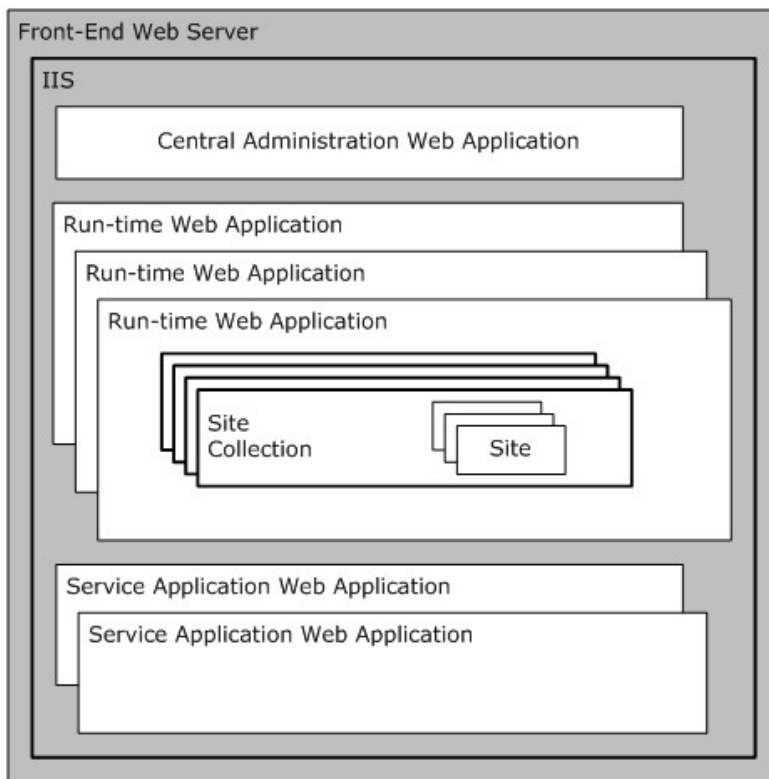
- A **Central Administration site** Web application (1) that handles the administration of Web sites (2) and of the farm running SharePoint Products and Technologies.
- A service application Web application (1) that enables key services spanning multiple Web sites (2).
- Zero or more run-time Web applications (1) that handle user requests.

A functioning farm includes at least one run-time Web application (1). Such a Web application (1) is necessary for creating Web sites (2) for user content and applications. In a common usage scenario, a browser accesses each Web application (1) via an independent TCP/IP port. For example, the Central Administration site Web application (1) might function over port 1002 and be accessed via the <http://myserver:1002> **URL**. The run-time Web application (1) might function over TCP/IP port 80, which is the default for HTTP.

This separation among the Web applications (1) provides both simplicity and security:

- Simplicity for users who do not need to see the complexity that is associated with administration and service applications.
- Security, because the user content, service applications, and Central Administration site all run in separate processes with separate **permissions** and **authentication (2)**.

The following figure shows the configuration of Web applications (1) in IIS.



**Figure 2: Configuration of Web applications (1) in IIS**

In this configuration, the run-time Web applications (1) render all of the user content and contain one or more site collections. The architecture of SharePoint Products and Technologies allows for a large number of site collections in a single deployment. A single deployment can also support more than one run-time Web application (1). Groups of site collections can thus be organized onto separate process boundaries, which in turn can be useful for supporting different modes of authentication (2) and for separating content into different levels of security on the front-end Web server.

### 2.1.1.2.2 Service Applications

A service application, also known as a **Shared Services Provider (SSP)** in Microsoft® Office SharePoint® Server 2007, exposes functionality that has been configured for providing extended capabilities across site collections and Web applications (1) or potentially across multiple farms running SharePoint Products and Technologies. A service application can be implemented as a Web service. Service applications that are implemented as Web services are configured and hosted by the service application Web application (1), as described in the overview of Web topology (section [2.1.1.2.1](#)).

For example, a full-text search can be used to aggregate content that exists both across many site collections and outside of SharePoint Products and Technologies. To enable such a search, the **search service application** defines the search capability, which each run-time Web application then accesses individually.

Every farm running SharePoint Products and Technologies includes zero or more service applications. The service applications that are provided with SharePoint Products and Technologies include:

- Excel® Services in Microsoft® SharePoint® Server 2010: Storing, calculating, and displaying spreadsheets, as described in section [2.1.3.1](#).
- Microsoft® Access Services: Storing, creating, and managing database applications, as described in section [2.1.3.2](#).
- User profiles: Creating **user profiles** and **audiences**, as described in section [2.1.3.4](#).
- Search services: Full-text querying, as described in section [2.1.3.5](#).
- Indexing: Building content indexes for full-text querying, as described in section [2.1.3.5](#).
- **Business Connectivity Services (BCS)**: Querying data from external systems, as described in section [2.1.3.6](#).
- Usage analytics: Recording and reporting on the user utilization of SharePoint **sites (2)**, as described in section [2.1.3.9](#).

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

### 2.1.1.2.3 Web Components

Because of their integration with IIS and ASP.NET, SharePoint Products and Technologies implements a full **Web server** that is capable of storing and delivering user content over HTTP and HTTP extensions. Web pages on SharePoint sites use the ASP.NET framework for rendering and executing Web content. Such Web pages include those that are installed by default as well as those that are customized. In addition, SharePoint Products and Technologies uses ASP.NET **Web Parts** for customizing and reusing Web content.

### 2.1.1.2.4 Web Parts

By allowing users to easily add and change information that appears on individual Web pages, ASP.NET Web Parts provide a powerful and easy-to-use mechanism for customizing the presentation of Web sites (2).

For example, a user might decide to change the layout of components on a site's (2) **home page** by removing some content and changing the page's layout or appearance. With the user interface (UI) components displayed on the Web site (2), the user can customize the page by changing settings on the Web Parts and by adding and removing Web Parts. Web Parts can provide this level of flexibility, because they are represented as separate resources that are not stored on the base ASP.NET page. Instead, Web Parts are ASP.NET components for which the executable code is deployed on the front-end Web server. All the settings and customizations for a Web Part are stored in a content database and associated with the location of the particular page on the particular site (2). For information about the types of databases provided by SharePoint Products and Technologies, see section [2.1.1.3.1](#).

A detailed technical specification of Web Parts as they are used in SharePoint Products and Technologies is available. For more information, see [\[MS-WSSTS\]](#) section 2.1.2.15.2. Information

about the development and utilization of Web Parts is also available. For more information, see [\[MSDN-WSSWP\]](#). Finally, a general overview of ASP.NET Web Parts is available. For more information, see [\[MSDN-ASPWP\]](#).

### 2.1.1.3 Data Storage

The following sections provide a summary of the physical and logical storage that is provided by SharePoint Products and Technologies. For more information, see [\[MS-WSSO\]](#) section 2.1.2.

#### 2.1.1.3.1 Physical Storage

All the information that is stored by SharePoint Products and Technologies is contained in relational databases in one or more instances of Microsoft® SQL Server® in the data storage layer. Individual front-end Web servers do not serve as repositories for user data. Instead, they store only the configuration or application information that can be easily redeployed. Application servers can store some constructed data that is derived from user data, such as a set of full-text indexes. Such constructed data can take time to reconstruct. The data storage is intended to be completely abstracted from SharePoint site users and administrators. In other words, normal usage of the farm running SharePoint Products and Technologies and interoperability with the front-end protocols does not require any direct interaction with the data storage environment.

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 [\[MS-SPBEPO\]](#) 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 [\[MS-SPBEPO\]](#) section 2.1.2.1.1. For information about the user of the configuration database, see [\[MS-WSSO\]](#) section 2.1.2.6.
- **Central Administration content database:** Stores content associated with the Central Administration site.
- **Shared Services Administration content database:** Microsoft® Office SharePoint® Server 2007 stores content for the Shared Services Administration Web site in this database. This database is not used in Microsoft® SharePoint® Server 2010.
- **Service applications databases:** Stores content and configuration information for service applications, such as the **User Profile Service**. The capabilities of service applications are described in [2.1.1.2.2](#).

#### 2.1.1.3.2 Logical Storage

Logically, all the user content that is stored by SharePoint Products and Technologies is located within the container hierarchy. A simplified perspective is as follows:

- **Site collections:** A site collection serves as the top-level boundary of storage and security for SharePoint Products and Technologies. A site collection also serves as the unit of scale, which means that the architecture of SharePoint Products and Technologies allows for a large number of site collections. Each site collection can contain a hierarchy of sites (2) and can directly contain **lists (1)** and **document libraries**.
- **Sites (2):** A site (2) provides a mechanism for subdividing the URL namespace, security, and organization of content within a site collection. Each site (2) can contain sites (2), lists (1), and document libraries.

- **Lists (1) and libraries:** Lists (1) and libraries act as the basic containers for content that is stored in SharePoint Products and Technologies. In other words, users store and access individual pieces of content from lists (1) and libraries. A list (1) can contain **list items**, and a library can contain **documents**. A SharePoint library stores documents with their associated metadata. Client applications can interact with the documents in SharePoint libraries by using either file operation protocols (section [2.1.2.3](#)) or authoring integration protocols (section [2.1.2.5](#)). Both lists (1) and libraries can be organized further into a hierarchy of folders.
- **Folders:** Folders act as organizational tools for categorizing the **items** or documents that exist in a list (1) or library. Folders enable browsing and permissions management. A folder can contain subfolders as well as individual list items and documents.
- **List items and documents:** In SharePoint Products and Technologies, the basic units of content are the list item and the document. In a list (1), the unit of content is the list item, which can be an individual task, announcement, or contact entry. In a library, the unit of content is the document with its associated metadata.

Each instance of the types of content described in this section will have its own URL that can be used to directly access and link to the content. In addition, users can create lists (1) and customize them with new **columns (2)**, **forms (2)**, and **views**.

For more information, see [\[MS-WSSTS\]](#) section 2.1.2.

#### 2.1.1.4 Back-End Protocols

The back-end protocols described in [\[MS-SPBEPO\]](#) are implemented by SharePoint Products and Technologies servers for communication within the farm and to enable communication between the SharePoint Products and Technologies components. For example, 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.

#### 2.1.1.5 Front-End Protocols

A front-end Web server running SharePoint Products and Technologies implements the front-end protocols that are described in this document. These protocols enable communication between SharePoint Products and Technologies and other client and server applications. Intra-farm communication, as described in [\[MS-SPBEPO\]](#), is not handled by the front-end protocols.

The front-end protocols are implemented by using **HTTP** or extensions to HTTP, such as **SOAP** Web services. For more information, see [\[SOAP1.1\]](#), [\[SOAP1.2/1\]](#), [\[SOAP1.2/2\]](#), [\[RFC2616\]](#), and [\[RFC2818\]](#).

Because HTTP and SOAP are used, the front-end protocols can be used by Web applications (1), tunneled through corporate network firewalls, and accessed through load-balancing technologies. In common usage scenarios, large-scale deployments of SharePoint Products and Technologies use network load balancers to distribute external requests among the front-end Web servers. These network load balancers operate at the **TCP/IP** transport level and do not require special knowledge of the front-end protocols to function.

The front-end protocols are divided into two main groups: core platform and extended capabilities.

The core platform protocols cover the following areas:

- Site (2) administration
- List (1) data
- **File** operations
- Security and **identity**
- Authoring integration
- Extensibility
- SharePoint client-side object model

The extended capabilities protocols cover the following areas:

- Excel® Services
- Microsoft® Access Services
- InfoPath® Forms Services
- User Profile Service
- Search services
- BCS
- Content Management Service
- PerformancePoint Services in Microsoft® SharePoint® Server 2010
- Analytics Services
- Office Online Web Query Web Service

Sections [2.1.2](#) and [2.1.3](#) summarize the core platform and extended capabilities groups of front-end protocols.

## 2.1.2 Core Platform

The core platform provides administrative services, a security and identity model, and the basic physical and logical storage for content.

### 2.1.2.1 Site Administration

Users access data and capabilities by interacting with individual SharePoint sites through a Web browser or other client application. Sites (2) are organized hierarchically, so a site (2) can contain zero or more subsites. A **hierarchy** of sites (2) is contained in a site collection, and multiple site collections can be contained in a single farm running SharePoint Products and Technologies. Technical details about site collections and sites (2) are available. For more information, see [\[MS-WSSTS\]](#) section 2.1.2.

The client application, whether it is a Web browser or other client application, communicates with the farm by means of HTTP-based protocols, as described in section [2.1.1.2.3](#). Every site collection and every site (2) within a farm has its own URL, so access patterns frequently involve users connecting directly to a site (2) by means of the URL and then navigating or searching within the site's content hierarchy.

SharePoint Products and Technologies provides generic site (2) administration capabilities, such as the basic creation, configuration, and inspection of all types of sites (2) and site collections. SharePoint Products and Technologies also supports additional site (2) administration functionality for specific types of sites (2), such as **Document Workspace sites** and **Meeting Workspace sites**. These sites (2) are built from specific **site definitions** and target specific usage scenarios. Document Workspace sites facilitate the collaboration of multiple authors on a single document with supporting collateral. A Meeting Workspace site provides a Web site (2) for organizing content that is associated with a meeting, such as agendas, participants, follow-up details, and meeting collateral.

Another site (2) administration capability entails linking the **membership** of a SharePoint site to an e-mail **distribution list**. If the deployment has been configured appropriately with a directory provider, a client application can use membership information to create and configure a distribution list in an external directory.

### 2.1.2.2 List Data

The list (1) data protocols enable core SharePoint list and library data access for creating and working with lists (1).

### 2.1.2.3 File Operations

The file operations protocols that are implemented by SharePoint Products and Technologies provide basic browse, open, edit, save, and rename file operations. These capabilities are exposed through clients such as Windows® Internet Explorer®, the 2007 Microsoft® Office system, and Microsoft® Office 2010 suites. At a high level, two primary technologies provide this interoperability: Microsoft® FrontPage® Server Extensions and **WebDAV**. Earlier versions of Microsoft® Office and Microsoft® Windows® implemented both FrontPage Server Extensions and WebDAV, whereas the newer versions of these clients are moving towards implementations that use only WebDAV. A description of how the 2007 Office system and Office 2010 use these protocols to communicate with SharePoint Products and Technologies is available. For more information, see [\[MS-OCPROTO\]](#) section 2.1.2.

SharePoint Products and Technologies also supports the ability to extract document metadata and link information from specific document file formats. This is called document **property promotion**. For information, see [\[MS-WSSQ\]](#) section 2.1.2.3.6.

### 2.1.2.4 Security and Identity

Security for SharePoint Products and Technologies can be divided into two separate but related areas: authentication (2) and **authorization**. Authentication (2) is the process of reliably validating the identity claims of an individual user. Authorization is the process of correctly determining which permissions that a user is supposed to have within the farm running SharePoint Products and Technologies.

#### 2.1.2.4.1 Authentication

For user authentication (2), SharePoint Products and Technologies relies on external authorities to appropriately validate a user's identity and, optionally, **group (2)** or role membership. By default, this external authority is Active Directory® Domain Services (AD DS), but SharePoint Products and Technologies supports pluggable security authentication (2) that allows any appropriately configured authentication provider to validate a user's identity. Such a provider, which is called an authentication system, provides a list of users and group (2) memberships that are available to a Web application (1) in SharePoint Products and Technologies. A technical specification of an authentication system is available. For more information, see [\[MS-WSSTS\]](#) section 2.1.4. An



example of how SharePoint Products and Technologies authenticates a specific user is also available. For more information, see [\[MS-WSSO\]](#) section 2.9.2.

Microsoft® SharePoint® Server 2010 also implements claims-based identity, an authentication system based on different industry standards as described in [\[MS-SPSTWS\]](#) section 1. This capability exists in addition to Windows Challenge/Response (NTLM) and the Kerberos protocol. The implementation of claims-based identity is standards based. Users who deploy SharePoint Server 2010 can use any protocol or service that implements these standards to provide identity to Microsoft® SharePoint® 2010 Products and Technologies. In addition, when applications running on SharePoint Server 2010 need to make external Web service calls, SharePoint Server 2010 will issue a **Security Assertion Markup Language (SAML)** token that identifies the user and application pool identity. For more information, see [\[WSTrust\]](#), [\[WSFederation\]](#), and [\[SAMLToken1.1\]](#).

#### 2.1.2.4.2 Authorization

Authentication system, group (2), and user identities are known as security **principals (3)**, which are granted permissions on specific objects in SharePoint Products and Technologies. Permissions can be set directly on standard types of content, such as site collections, sites (2), lists (1), libraries, folders, and items. In addition, permissions can be inherited from containers by the objects that exist in those containers. To simplify the administrative overhead that is associated with assigning permissions, SharePoint Products and Technologies implements the concept of groups (2), allowing a collection of principals (3) to be assigned the same permissions and roles. A description of how authorization is implemented is available. For more information, see [\[MS-WSSO\]](#) section 2.9.1. A technical description of permissions is also available. For more information, see [\[MS-WSSTS\]](#) section 2.1.4.

#### 2.1.2.5 Authoring Integration

In addition to basic file operations, SharePoint Products and Technologies implements several protocols that can be used by authoring applications, such as Microsoft® Office 2010 suites, to provide specific, enhanced experiences for content creation. For example, SharePoint Products and Technologies implements protocols that facilitate scenarios in which a client application produces content that is targeted for the server (2). Such content might include a message for a **blog (1)** or slides for a **Slide Library**.

Microsoft® SharePoint® Server 2010 also implements protocols to help synchronize document content between client applications. These protocols allow a client application to incrementally upload or download changes to documents along with any metadata associated with the changes from SharePoint libraries. These protocols enable coauthoring features in Microsoft® Word 2010 and Microsoft® PowerPoint® 2010 as well as document synchronization in Microsoft® SharePoint® Workspace 2010.

For more information, see [\[MS-OCPROTO\]](#) sections [2.1.10](#), [2.1.12](#), and [2.1.18](#).

#### 2.1.2.6 Extensibility

SharePoint Products and Technologies implements support for content import and export, **workflow (1)** integration, and Web page and Web Part customization. For information about workflow (1) in SharePoint Products and Technologies, see [\[MSDN-MOSSWF\]](#) and [\[MSDN-WSSWF\]](#).

#### 2.1.2.7 SharePoint Client-Side Object Model

Microsoft® SharePoint® Foundation 2010 implements a client-side object model. The SharePoint Client-Side Object Model enables a client application to make batch requests to a front-end Web server running SharePoint Products and Technologies to perform operations on core platform objects

such as lists (1), sites (2), and users. The operations can include method calls, property read operations, and property write operations.

Through the SharePoint Client-Side Object Model, a client application has access to much of the same functionality that is exposed by the SOAP Web services mentioned in section [2.1.1.5.5](#). The SharePoint Client-Side Object Model differs from the SOAP Web services in that it is optimized to support the placement of a batch of operations into a single request by the client application. The operations can also span object types, whereas the SOAP Web services are organized by individual object type. For more information, see [\[MS-CSOMSPT\]](#) section 3.2.1.

### 2.1.3 Extended Capabilities

The extended capabilities build on the core platform that was described in the preceding sections.

#### 2.1.3.1 Excel Services

Microsoft® SharePoint® Server 2010 implements Excel® Services in Microsoft® Office SharePoint® Server 2010, which is used to create, update, format, **calculate**, and save **workbooks** on a server (2). Excel Services 2010 provides the following core capabilities:

- The ability to publish (1) workbooks from a client application, such as Microsoft® Excel® 2010, to the server (2). Users can then view and interact with these workbooks via a Web browser. This ability enables browser-based interaction with the workbooks from machines that do not have a spreadsheet application installed.
- The ability to programmatically access workbook contents via Excel Web Services, which is a Web services protocol. This ability enables client applications to load, edit, interact with, calculate, and save workbooks that have been published (1) to SharePoint Server 2010.

Microsoft® Excel® Calculation Services 2010 provides the core calculation and workbook processing for Excel Services 2010. Excel Calculation Services 2010 can be deployed on a separate application server (2), as described in section [2.1.1.1.1](#). A description of the architecture of this application server and the protocol communication with it is available. For more information, see [\[MS-SPBEPO\]](#) section 2.1.3.1. General information about Excel Calculation Services 2010 is also available. For more information, see [\[MSDN-EXCELS\]](#).

As in the usage scenario for viewing workbooks in the browser, it is possible to programmatically access and manipulate server (2)-based workbooks by using Web services. Operations available via the Excel Web Services API include opening, setting ranges in, refreshing **external data** in, recalculating, getting ranges from, and saving a workbook.

Workbooks that are loaded by Excel Calculation Services 2010 can reference external data. The data can be in relational or multidimensional databases. Similar to a client spreadsheet application, Excel Calculation Services 2010 uses the following data access APIs to retrieve information from such data sources:

- **Open Database Connectivity (ODBC)**
- **OLE DB**
- OLE DB for **Online Analytical Processing (OLAP)**

If a workbook contains elements that reference external data, such as PivotTable or data access formulas, Excel Calculation Services 2010 will perform the data retrieval by using one of the preceding methods.

A common way to extend these features to new data sources is to install an appropriate **ODBC** driver for the data source, thereby making it available to Excel Services 2010. ODBC is a data access API that is commonly used. ODBC is not a wire protocol. Many ODBC drivers already exist for different data sources. When communicating with Microsoft® SQL Server® data sources, the OLE DB providers included with the Microsoft® .NET Framework are used. These providers use the Tabular Data Stream (TDS) protocol to communicate with the server (2) running SQL Server. General descriptions of OLE DB and ODBC are available. For more information, see [\[MSDN-OLEDB\]](#) and [\[MSDN-ODBC\]](#).

Excel Calculation Services 2010 can communicate not only with strictly relational data sources but with multidimensional databases. For this purpose, SharePoint Products and Technologies can install an **OLAP** driver that communicates with SQL Server by using the SQL Server Analysis Services Protocol. For more information, see [\[MS-SSAS\]](#).

Two types of front-end protocols are used for interoperability with Excel Services 2010. In addition to the Excel Services Publishing Protocol, as described in [\[MS-ESURL\]](#), workbook publishing from client authoring applications to Excel Services 2010 also uses the file operations protocols as described in section [2.2.1.3](#).

### 2.1.3.2 Access Services

Microsoft® SharePoint® Server 2010 implements Microsoft® Access Services, which is used to create database applications and manage data. Access Services provides the following core capabilities:

- The ability to publish (1) database applications from a client application, such as Microsoft® Access® 2010, to the server (2). Users can then view and edit data in these databases via a Web browser. This ability enables browser-based viewing and interaction with the databases from machines that do not have a database application installed.
- The ability to programmatically access database contents via Access Services Web Services, which is a Web services protocol. This ability enables client applications to retrieve and interact with databases that have been published (1) to SharePoint Server 2010.

A user can publish (1) Access 2010 database applications to SharePoint Server 2010. Access Services creates a SharePoint site for each published (1) Access 2010 database application. Access 2010 creates a list (1) for each table in the Access 2010 database application and uploads the data from the database to the lists (1).

Access Services Data Server retrieves and manages data from the storage in SharePoint Products and Technologies for the database applications. For more information, see [\[MS-ADS\]](#).

A user can also get application reference files for the Access 2010 database application. The application reference files connect to the lists (1), forms (2), and reports in the database application that was published (1) and allows the user to edit them in Access 2010. For more information, see [\[MS-AXL\]](#) section 1.3.7.

### 2.1.3.3 InfoPath Forms Services

InfoPath® Forms Services in Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® Server 2010 enables users to fill out **form templates** in a Web browser. This ability requires that InfoPath Forms Services be enabled and that the **browser-enabled form template** be published (1) to the server (2). For more information, see [\[MS-OCPROTO\]](#) section 2.2.3.

Office SharePoint Server 2007 and SharePoint Server 2010 also implements the ability of form-editing applications, such as Microsoft® InfoPath® 2010, to store forms (2) as content in document libraries. This ability relies on the file operations protocols as described in section [2.2.1.3](#).

#### 2.1.3.4 User Profile Service

Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® Server 2010 implement a User Profile Service that stores information about a user population. This information might include names, e-mail addresses, and telephone numbers. The User Profile Service aggregates information from multiple data sources and provides a uniform interface to this information at both the user level and the programming level. The User Profile Service exposes a number of capabilities including consistent ways to query across the user data, categorize users into groups (2), create indexes of user data for full-text searches, access a consistent **change log** of all the modified user data, and synchronize the user data with multiple SharePoint sites in a farm.

The User Profile Service is implemented as a service application, as described in section [2.1.1.2.2](#). Multiple site collections can thus refer to one central User Profile Service for information about users.

A profile is intended to be a description of a person. In common usage scenarios, each profile represents one person. In some situations, it represents one identity (for example, a user account). The profile consists of a set of identifiers (for example, a user name, an e-mail address, and an alias) and a list (1) of values for additional properties. Office SharePoint Server 2007 and SharePoint Server 2010 also implement support for properties that contain multiple values, such as a property that contains a list (1) of skills for a person, and for properties that are linked to specific, restricted lists (1) of allowed values. Access to properties can be restricted so that only a specific group (2) of related users can see a particular value.

In addition to the core concept of a profile, Office SharePoint Server 2007 and SharePoint Server 2010 implement support for grouping profiles together. One mechanism for doing so is to import key grouping information from other sources, such as e-mail distribution lists (1) or site memberships. Office SharePoint Server 2007 and SharePoint Server 2010 also allow the calculation of collections of user profiles based on profile metadata information. Such a collection is called an audience. For example, all the managers of a company can comprise an audience. Web pages or client applications can then target information to this audience.

In SharePoint Server 2010, the User Profile Service also implements support for adding **social data** to any **URI** and for retrieving social activity events information, such as service anniversaries, distribution list memberships, and social data activity on a URI. The URI can point to a resource or to content inside or outside of SharePoint Products and Technologies. Social data can be in the form of numerical ratings, a note that is represented by lines of text, and keywords or **terms**. Keywords or terms are 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 with something familiar to the user (for example, "social computing" for a document that is related to social computing), and writing a descriptive line of text as a note about some content.

#### 2.1.3.5 Search Services

Search services provide the ability for a user to quickly see a list of the most relevant documents in a large document collection, based on a specified keyword query. A large document collection can have tens of millions of documents. This quick query resolution requires the creation and maintenance of two basic data structures: the **inverted index** and the property store, which is known as the **metadata index** in Microsoft® SharePoint® Server 2010. SharePoint Products and Technologies implements the protocols that are necessary for creating these data structures, including those for crawling (traversing the URL space of the documents that the user wants to

search and synchronizing the changes in the URL space), retrieving the documents, parsing the documents, performing tokenization, creating the inverted index, and propagating the index. SharePoint Products and Technologies also implements the front-end protocols that are necessary for executing queries against these data structures and for administering the search configuration.

Windows® SharePoint® Services, Microsoft® SharePoint® Foundation 2010, and Microsoft® SharePoint® Server all implement search protocols. In Windows SharePoint Services and SharePoint Foundation 2010, the protocols are limited to the subset that is necessary for crawling and providing a search capability across the content that is stored in Windows SharePoint Services and SharePoint Foundation 2010 respectively. The SharePoint Products and Technologies search service provides search capabilities across the content in all the Windows SharePoint Services and SharePoint Foundation 2010 sites in the enterprise as well as across other Web sites (2), file shares, and other **content sources**. Unless otherwise specified, the concepts, configuration, and protocols apply to search services in Windows SharePoint Services, SharePoint Foundation 2010, and SharePoint Server.

### 2.1.3.6 Business Connectivity Services

Business Connectivity Services (BCS) provides access to data from server applications that are external to the farm running SharePoint Products and Technologies, thus bridging the gap between SharePoint sites and business applications. Through Business Connectivity Services, Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® 2010 Products and Technologies have built-in support for displaying data from databases and Web services. Business Connectivity Services also facilitates the integration of data from line-of-business applications, such as SAP, Siebel, and other databases, into lists (2), Web Parts, search services, user profiles, and custom applications. However, Business Connectivity Services does not specify a protocol for communicating with each line-of-business application. The SharePoint 2010 Products implementation of Business Connectivity Services supports the ability to write back to these data sources, as well.

BCS consists of two parts: a **Business Data Connectivity (BDC)** (referred to as Business Data Catalog in Office SharePoint Server 2007) **metadata store** and an object model. The BDC service uses metadata to describe the APIs of the line-of-business applications. After the API description of a line-of-business application is registered in the metadata store, it provides dynamic data-access connections to the back-end data without the need for additional code.

The metadata model abstracts the underlying physical sources and provides a consistent and uniform way to work with different line-of-business applications. The metadata for each data source defines the programming objects that applications interact with and the methods that are available for client applications. Although the metadata model provides a consistent way to work with different data sources, BCS does not specify a generic protocol for communicating with each source.

For more information, see [\[MSDN-MOSS2007-SDK\]](#).

Note that Office SharePoint Server 2007 and Microsoft® SharePoint® Server 2010 also implement a capability for securely storing user credentials for authenticating to a line-of-business application. This capability is called the **Secure Store Service (SSS)**.

### 2.1.3.7 Content Management Service

Microsoft® Office SharePoint® Server 2007 and Microsoft® SharePoint® Server 2010 implement enterprise content management capabilities on top of the core storage, platform, and file operations infrastructure that is described in this document. These capabilities include document management and versioning, metadata management, tools for propagating and managing content across multiple installations, records management tools, and support for converting documents from one format to another.

Office SharePoint Server 2007 continues to implement the obsolete Area Web Service. Microsoft does not implement any protocol clients for this protocol. SharePoint Server 2010 does not implement the Area Web Service; nor will this protocol be supported in future releases. It is advisable for implementers of this protocol to discontinue usage.

### 2.1.3.8 PerformancePoint Services

PerformancePoint Services in Microsoft® SharePoint® Server 2010 allows users to author and publish (1) business intelligence data to a Web Part page in SharePoint Products and Technologies. This page, also known as a **dashboard** in PerformancePoint Services 2010, provides users with the ability to analyze their business data based on business rules and metrics. Dashboards can be filled with content representing business performance indicators (such as grids, charts, **scorecards**, and other reports). This content is composed of metadata (such as the names of dimensions, measures, or data source references), calculation definitions, and formatting information (such as colors or layout information).

The Web services for PerformancePoint Services 2010 enable a protocol client to:

- Create, read, update, validate, and delete a variety of business logic content in a content store in SharePoint Products and Technologies.
- Create, read, update, and delete annotations that exist on a scorecard.
- Retrieve metadata, such as the names of dimensions or measures, and data sets via data source references.
- Retrieve a list (2) of content subtypes and component information from a configuration store.

A typical scenario for using these services involves creating an extensible content-editing application that enables users to create, publish (1), and maintain some of the content that is used in a larger business-intelligence system. A manager for a large business, for example, might want to create a chart that shows the daily sales volume for a specific product. The chart would contain active areas that would allow the manager to click those areas for additional information, such as the breakdown of the net profit and cost for that product.

### 2.1.3.9 Analytics Services

Analytics Services allows users to access the following three categories of analytical data about the way a farm is being used:

- **Traffic data:** Data about Web traffic. Examples include the most-visited Web pages, trends about Web page visits, top visitors, and trends about the number of unique visitors.
- **Search reports:** Data about search queries and search results. Examples include the top queries, failed queries, and number of queries.
- **Inventory reports:** Data about the usage of various entities, such as storage, libraries, and templates.

The usage data can be retrieved for entities at various levels in the farm, such as site (2), site collection, and Web application (1). To generate usage data, a client application logs specific events about an entity to the system. After the events are logged, the client application can retrieve usage data for that entity. For example, a client application might log all the pages that were visited on a particular site (2). The client application can then retrieve a report about the traffic volume per day for that particular site (2), the top pages visited for that site (2), and the top visitors for that particular site (2).

Microsoft® SharePoint® Server 2010 employs a usage database to store usage data of various kinds, including user request information, performance counters, data about slow or expensive queries, and other relevant performance data. This database can be extended by non-Microsoft features that are built upon SharePoint Products and Technologies. For more information see [\[MS-UTSP\]](#).

### 2.1.3.10 Office Online Web Query Service

The Office Online Web Query Web Service enumerates details about files in the Office Online repository. Microsoft® SharePoint® 2010 Products and Technologies has a user interface for displaying **solution packages** stored in the Office Online repository, downloading a solution package, and storing it in a site collection's **solution gallery**.

## 2.2 Protocol Summary

The tables in this section provide a comprehensive list of the member protocols of the SharePoint Front-End Protocols System. The member protocols are grouped according to their primary purpose.

### 2.2.1 Core Platform Protocols

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

#### 2.2.1.1 Site Administration

The protocols in the following table enable site (2) administration in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies. These protocols are generic across all site collections and sites (2).

Protocol name	Description	Short name
Administration Web Service Protocol	Provides the ability to create and delete site collections and to retrieve information about a site collection, such as which languages are supported by the site collection.	<a href="#">[MS-ADMINS]</a>
Webs Web Service Protocol	Provides the ability to access and modify sites (2) and the attributes of those sites (2) in a site collection. This protocol can also be used to modify the design and layout of the sites (2).	<a href="#">[MS-WEBSS]</a>

The protocols in the following table enable site (2) administration for Document Workspace sites and Meeting Workspace sites in SharePoint 2007 Products and Technologies and in SharePoint 2010 Products.

Protocol name	Description	Short name
Document Workspace Web Service Protocol	Provides the ability to create, edit, and delete Document Workspace sites. This protocol also allows the protocol client to organize the content in the Document Workspace sites by using folders.	<a href="#">[MS-DWSS]</a>
Meetings Web Services Protocol	Provides the ability to create, edit, and delete Meeting Workspace sites.	<a href="#">[MS-MEETS]</a>

The protocol in the following table enables **site membership** and directory integration in SharePoint 2007 Products and Technologies and in SharePoint 2010 Products.

Protocol name	Description	Short name
SharePoint E-Mail Web Service Protocol	Provides the ability to create and configure distribution lists. SharePoint Products and Technologies includes a simple implementation of this Web service that can be configured to use <b>Active Directory Service Interfaces (ADSI)</b> to transmit requests to Active Directory® Domain Services (AD DS).	<a href="#">[MS-SPHEMAWS]</a>

### 2.2.1.2 List Data

The protocols in the following table enable core SharePoint list and library data access in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Lists Web Service Protocol	Provides the ability to create, edit, delete, and configure lists (1) and list (1) data and to configure settings and metadata that are specific to lists (1).	<a href="#">[MS-LISTSWS]</a>
Views Web Service Protocol	Provides methods for adding a list (1) view and for retrieving a specific view or the collection of list (1) views.	<a href="#">[MS-VIEWS]</a>
Forms Service Protocol	Provides methods for getting a list (1) of forms (2) from the server (2) and for retrieving an individual form (2) from that list (1). Protocol clients can use these methods to display viewing and editing forms (2) for individual list items to users.	<a href="#">[MS-FORMS]</a>

The protocol in the following table enables core SharePoint list and library data access only in SharePoint 2010 Products.

Protocol name	Description	Short name
ListData Data Service Protocol	Provides the ability to create, edit, delete, and retrieve list (2) data.	<a href="#">[MS-WSSREST]</a>

The structures in the following table enable SharePoint list and library data access in SharePoint 2007 Products and Technologies and in SharePoint 2010 Products.

Structure name	Description	Short name
Collaborative Application Markup Language (CAML) Structure	Describes the use of <b>Collaborative Application Markup Language (CAML)</b> to describe queries, views, and lists (1) across multiple external and internal protocols that are implemented by SharePoint Products and Technologies.	<a href="#">[MS-WSSCAML]</a>
ADO XML Persistence Format Protocol	Describes the subset of the ADO XML Persistence Format used to represent tabular data, such as list (2) data returned by [MS-LISTSWS].	<a href="#">[MS-PRSTFR]</a>



The protocol in the following table enables **alerts (2)** for SharePoint list and library data in SharePoint 2007 Products and Technologies and in SharePoint 2010 Products.

Protocol name	Description	Short name
Alerts Web Service Protocol	Allows a protocol client to list and delete <b>alert subscriptions</b> , which specify when and how notifications are sent to users when changes are made to content that is stored on the server (2).	<a href="#">[MS-ALERTSS]</a>

The protocol in the following table enables the Office Mobile Service Protocol. SharePoint 2010 Products can use this protocol to send an alert (1) to a protocol server, and the protocol server sends the alert (1) to a mobile phone.

Protocol name	Description	Short name
Office Mobile Service Protocol	Enables a protocol client, such as SharePoint 2010 Products, to transmit mobile messages to a protocol server.	<a href="#">[MS-OMS]</a>

The protocols in the following table enable additional ways to access list (2) data in SharePoint 2007 Products and Technologies and in SharePoint 2010 Products. These protocols are implemented for historical interoperability with Microsoft® Office Excel® 2003 and with Microsoft® Office FrontPage® 2003, which was superseded by Microsoft® Office SharePoint® Designer 2007. For common usage scenarios, implementing these protocols is not advised because the same capabilities are available through the Lists Web Service Protocol, as described in [\[MS-LISTSWS\]](#).

Protocol name	Description	Short name
Data-Source Adapter SharePoint Team Services Web Service Protocol	Enables a protocol client to obtain structured tabular data from a server (2) and provides access to metadata about the server (2) and how the tabular data is organized.	<a href="#">[MS-DSPSTSS]</a>
Windows SharePoint Services Collaborative Application Protocol	Enables a protocol client to access and manipulate lists (1) and list (2) views on the server (2).	<a href="#">[MS-WSSCAP]</a>

The protocols in the following table enable synchronization with Microsoft® Office Outlook® 2007 or Microsoft® Outlook® 2010 in SharePoint 2007 Products and Technologies and in SharePoint 2010 Products.

Protocol name	Description	Short name
Lists Client Sync Protocol	Determines the processing and logic to host local copies of SharePoint list items in Office Outlook 2007 or Outlook 2010.	<a href="#">[MS-OUTSPS]</a>
StsSync Structure	Instantiates a connection between Office Outlook 2007 or Outlook 2010 and a SharePoint list by using a URL format.	<a href="#">[MS-STSSYN]</a>
Alerts Interoperability Protocol	Extracts data from an alert (2) sent from SharePoint Products and Technologies to Office Outlook 2007 or Outlook 2010.	<a href="#">[MS-OSALER]</a>

### 2.2.1.3 File Operations

The protocols in the following table enable file operations in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
HTTP Extensions for Distributed Authoring -- WEBDAV	Provides file access and content management.	<a href="#">[RFC2518]</a>
Web Distributed Authoring and Versioning (WebDAV) Protocol: Client Extensions	Provides file access and content management over the Internet. These extensions to WebDAV provide enhanced features, such as new error codes and additional TCP ports.	<a href="#">[MS-WDV]</a>
Web Distributed Authoring and Versioning (WebDAV) Protocol: Microsoft Extensions	Provides extensions and limitations to WebDAV as implemented by SharePoint Products and Technologies.	<a href="#">[MS-WDVME]</a>
Web Distributed Authoring and Versioning (WebDAV) Protocol: Server Extensions	Provides extensions to WebDAV, including headers that both enable the file types that are not currently manageable and optimize protocol interactions for file system clients.	<a href="#">[MS-WDVSE]</a>
Office Document Update Utility Extensions Protocol	Provides extensions to WebDAV, including header updates, a property that enables optimized protocol interaction for synchronization, and a property that allows the server (2) to send the virus infection status of a document to client applications. The Document Workspace feature of Microsoft® Office clients uses these extensions to synchronize documents with SharePoint Products and Technologies.	<a href="#">[MS-WDVMODUU]</a>
Web Distributed Authoring and Versioning Error Extensions Protocol	Describes the extended error codes that SharePoint Products and Technologies can return to a client application. The extended errors include: Document checked out Minor version limit extended Missing necessary document properties	<a href="#">[MS-WEBDAVE]</a>
FrontPage Server Extensions Remote Protocol	Provides file access functionality that is similar to that of WebDAV, allowing a Web site (2) to be presented as a <b>share</b> .	<a href="#">[MS-FPSE]</a>
Hypertext Transfer Protocol -- HTTP/1.1	Provides application-level methods for distributed, collaborative, and hypermedia information systems. This generic, stateless, protocol can be used for many tasks, including those for file access. If other protocols are not supported, Microsoft Office clients use <b>HTTP GET</b> for read-only access to files.	<a href="#">[RFC2616]</a>

### 2.2.1.4 Security and Identity

The protocols in the following table enable user authentication (2) and identification in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Authentication Web Service Protocol	Identifies the <b>authentication mode</b> that is in use by a Web application (1), and enables user authentication (2) when <b>forms authentication</b> is the authentication mode.	<a href="#">[MS-AUTHWS]</a>
People Web Service Protocol	Enables a protocol client to resolve and find principals (3).	<a href="#">[MS-PEOPS]</a>
Office Forms Based Authentication Protocol Specification	Enables a protocol client to authenticate to a server (2) configured to use Office Forms Based Authentication.	<a href="#">[MS-OFBA]</a>

The protocols in the following table enable claims-based identity only in SharePoint 2010 Products.

Protocol name	Description	Short name
SharePoint Claim Provider Web Service Protocol	Enables a protocol client to enumerate, search for, and resolve the security principals (3) that are available in SharePoint 2010 Products.	<a href="#">[MS-CPSWS]</a>
SharePoint Security Token Service Web Service Protocol	Describes the restrictions placed on the messages for interoperating with the SharePoint 2010 Products implementation of a Security Token Service.	<a href="#">[MS-SPSTWS]</a>

The protocols in the following table enable SharePoint Products and Technologies authorization in SharePoint 2007 Products and Technologies and in SharePoint 2010 Products.

Protocol name	Description	Short name
Permissions Web Service Protocol	Enables client applications to manage permissions to server (2) sites (2) and lists (1) through a Web service. This protocol describes the messages for adding, modifying, retrieving, and removing permissions.	<a href="#">[MS-PERMS]</a>
UserGroup Web Service Protocol	Enables the protocol client to define, modify, and retrieve information about users or groups (2) and the role definitions to which they are assigned in a specific site (2) or site collection.	<a href="#">[MS-UGS]</a>

### 2.2.1.5 Authoring Integration

The protocol in the following table enables extended HTTP headers in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
HTTP Windows SharePoint	Enables previously undefined behaviors by means of new headers and messages, which are extensions to HTTP mechanisms. The new	<a href="#">[MS-</a>

Protocol name	Description	Short name
Services Headers Protocol	application scenarios that these extensions introduce include authenticating client connections, communicating error conditions, sending complex data, interacting with information rights management systems, interacting with antivirus systems, and interacting with <b>crawlers</b> .	<a href="#">WSSHP1</a>

The protocols in the following table enable scenario-specific content authoring in SharePoint 2007 Products and Technologies and in SharePoint 2010 Products.

Protocol name	Description	Short name
MetaWeblog Extensions Protocol	Describes extensions and modifications to the MetaWeblog API (MWA). These extensions and modifications allow client applications to get and set the text and attributes of posts on a blog (1).	<a href="#">[MS-METAWEB]</a>
Slide Library Web Service Protocol	Enables a protocol client to send a request to the protocol server (2) and to then receive information about the existence of named slides, about all slides, about slides that are specified by their identifiers, or about slides that contain specific search strings.	<a href="#">[MS-SLIDELI]</a>
Imaging Service Protocol	Enables a protocol client to retrieve, upload, and organize images in a <b>picture library</b> .	<a href="#">[MS-IMAGS]</a>

The protocol in the following table enables diagnostic submission only in SharePoint 2010 Products.

Protocol name	Description	Short name
SharePoint Diagnostics Web Service Protocol	Enables a protocol client to submit details about an error report—details such as the call stack, an error message, or the operating environment.	<a href="#">[MS-SPDIAG]</a>

The protocols in the following table enable file synchronization only in SharePoint 2010 Products.

Protocol name	Description	Short name
File Synchronization via SOAP over HTTP Protocol	Enables a client application to make a SOAP Web service request that allows for incrementally uploading or downloading file changes, along with the related metadata changes, from a server (2) that supports the protocol. This protocol enables the coauthoring feature in Microsoft® Word 2010 and Microsoft® PowerPoint® 2010.	<a href="#">[MS-FSSHTTP]</a>
Binary Requests for File Synchronization via SOAP Protocol	Enables a protocol client to synchronize the state of a structured file that is hosted by a protocol server.	<a href="#">[MS-FSSHTTPB]</a>

The protocol in the following table enables coauthoring only in SharePoint 2010 Products.

Protocol name	Description	Short name
Shared Access Web	Enables a client application to determine whether a	<a href="#">[MS-</a>

Protocol name	Description	Short name
Service Protocol	document is being coauthored.	<a href="#">SHDACCWS</a>

### 2.2.1.6 Extensibility

SharePoint Products and Technologies implements a number of protocols that are designed specifically to provide extensibility and customization. These protocols can be divided into three major areas:

- Content import and export protocols
- Workflow integration protocols
- Web page and Web Part customization protocols

#### 2.2.1.6.1 Content Import and Export

The protocols and structure in the following table enable importing and exporting content from Microsoft® SharePoint 2007 Products and Technologies and from Microsoft® SharePoint® 2010 Products and Technologies.

Protocol or structure name	Description	Short name
Sites Web Service Protocol	Provides methods for importing and exporting a site (2) from a server (2), retrieving <b>site template</b> information, and obtaining a <b>form digest validation</b> value for authenticating a secure submit request that was made to the server (2).	<a href="#">[MS-SITESS]</a>
RSS 2.0 Format Extensions	Provides methods for serializing data from a Web site (2).	<a href="#">[MS-SPRSS]</a>
Deployment Package Format	Defines a set of <b>XML</b> and binary files to encapsulate site content and relationships. This file format enables the portability of user data.	<a href="#">[MS-PRIMEPF]</a>

#### 2.2.1.6.2 Workflow Integration

The protocols in the following table enable workflow development and administration as well as user interaction in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Workflow Web Service Protocol	Starts new workflows (1), changes existing ones, and queries for status. A common usage scenario for this protocol is managing a document approval process.	<a href="#">[MS-WWSP]</a>
Group Approval Web Service Protocol	Submits or approves Group Approval documents. A common usage scenario for this protocol involves a custom Group Approval application that allows users to remotely access Group Approval data.	<a href="#">[MS-EAWF]</a>
Web Part Pages Web Service	Describes a format for workflow definitions that will be deployed to a server (2) running SharePoint Products and Technologies.	<a href="#">[MS-WPPS]</a>

Protocol name	Description	Short name
Protocol		

### 2.2.1.6.3 Web Page and Web Part Customization

As described in section [2.1.1.2.4](#), SharePoint Products and Technologies implements a capability for a Web page to integrate multiple pieces of content through the Web Part mechanism.

The protocols in the following table enable Web Part page customization in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Web Part Pages Web Service Protocol	Enables a protocol client to gather the information that is necessary to author Web pages that use server (2) resources, and allows the authoring of execution logic that will react to changes in the server (2) state.	<a href="#">[MS-WPPS]</a>
Content Area Toolbox Web Service Protocol	Enables a protocol client to enumerate the <b>Web controls</b> that are available for use on a Web page that is stored on a protocol server. A common usage scenario involves a Web page editing application that queries the protocol server for Web controls and then displays those controls so that the user can add them to a Web page.	<a href="#">[MS-CONATB]</a>

### 2.2.1.7 SharePoint Client-Side Object Model

The protocols in the following table enable the SharePoint Client-Side Object Model only in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
SharePoint Client Query Protocol	Enables a protocol client to execute a batch of method invocations, property read operations, and property write operations on logical objects or class types.	<a href="#">[MS-CSOM]</a>
SharePoint Client-Side Object Model API Protocol	Defines the types, methods, and properties of objects such as sites (2) and lists (1).	<a href="#">[MS-CSOMSPT]</a>

## 2.2.2 Extended Capabilities Protocols

This section lists the external protocols that are used to interoperate with the extended capabilities of SharePoint Products and Technologies.

### 2.2.2.1 Excel Services

The protocols in the following table enable Excel® Services in both Microsoft® SharePoint 2007 Products and Technologies and Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Excel Web Services Protocol	Enables a protocol client to access workbooks that are stored on a Web server or in a location that can be reached by using a Universal Naming Convention (UNC) path.	<a href="#">[MS-ESP]</a>
Excel Services Publishing Protocol	Enables a client application, such as Microsoft® Excel® 2010, to form the URL and associated parameters for a query string to display the workbook in the browser after the workbook is published (1) to the protocol server.	<a href="#">[MS-ESURL]</a>

The protocols in the following table enable Excel Services only in SharePoint 2010 Products.

Protocol name	Description	Short name
Excel Services REST Protocol	Enables a client application to access information stored in workbooks that are maintained by a protocol server. The information returned by the protocol can be in one of several formats, such as <b>HTML</b> , image, and as an <b>atom feed</b> . This protocol client also allows for values to be inserted into cells of the workbook.	<a href="#">[MS-ESREST]</a>

The protocols in the following table enable remote data access for Microsoft® Excel® Calculation Services in both SharePoint 2007 Products and Technologies and SharePoint 2010 Products.

Protocol name	Description	Short name
Tabular Data Stream Protocol	Enables remote data access to Microsoft® SQL Server®.	<a href="#">[MS-TDS]</a>
SQL Server Analysis Services Protocol	Enables remote access to OLAP data.	<a href="#">[MS-SSAS]</a>

### 2.2.2.2 Access Services

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

Protocol or structure name	Description	Short name
Access Services Protocol	Enables a client application, such as Microsoft® Access® 2010, to interact with Access 2010 Web applications (1) that have been published (1) to SharePoint 2010 Products.	<a href="#">[MS-ASWS]</a>
Access Application Transfer Protocol Structure	Defines a format for describing data application components.	<a href="#">[MS-AXL]</a>
Access Template File Format	Enables the creation of the structure and content of an Access 2010 database application.	<a href="#">[MS-ACCDT]</a>

### 2.2.2.3 InfoPath Forms Services

The protocols and structures in the following table enable InfoPath® Forms Services in both Microsoft® SharePoint 2007 Products and Technologies and Microsoft® SharePoint® 2010 Products and Technologies.

Protocol or structure name	Description	Short name
Forms Services Design and Activation Web Service Protocol	Allows a protocol client to communicate with a <b>form server</b> to validate, <b>browser-enable</b> , or manipulate settings of form template files in supported scenarios.	[MS-FSDAP]
Forms Services Feature Detection Protocol	Allows a protocol client to detect whether InfoPath Forms Services is present and enabled on the server (2). InfoPath Forms Services is necessary to publish (1) a browser-enabled <b>form template (.xsn) file</b> .	[MS-FSFDPI]
InfoPath Form File Format Structure	Enables a form server, with an associated form template (.xsn) file, to render and edit form data from a <b>form file</b> in a Web browser.	[MS-IPFFX]
Universal Data Connection 2.0 XML File Structure	Provides a container for <b>data connection (3)</b> information.	[MS-UDCX]
InfoPath Data Connection File Download Protocol	Allows a protocol client to retrieve a <b>Universal Data Connection (.udc, .udcx) file</b> .	[MS-INFODCF]
Forms Services Proxy Web Service Protocol	Forwards SOAP messages for a client application, and returns the targeted Web service response.	[MS-FSPP]

The structure in the following table enables InfoPath Forms Services only in SharePoint 2007 Products and Technologies.

Structure name	Description	Short name
InfoPath Form Template Format Structure	Enables a form server to render and edit form data in a Web browser.	[MS-IPFF]

The structure and protocol in the following table enable InfoPath Forms Services only in SharePoint 2010 Products.

Structure name	Description	Short name
InfoPath Form Template Format Version 2 Structure	Enables a form server to render and edit form data in a Web browser.	[MS-IPFF2]
InfoPath Digital Signing Protocol	Enables a form server client to digitally sign a form file.	[MS-IPDSP]



### 2.2.2.4 User Profile Service

The protocols in the following table enable the User Profile Service in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Published Links Web Service Protocol	Retrieves a list (1) of URLs and descriptive details about each URL that a protocol client can inspect to decide whether each URL is useful for its particular application. A common usage scenario involves providing a list (1) of SharePoint sites to be displayed in an Open or Save dialog box.	<a href="#">[MS-PLSP]</a>
User Profile Service Web Service Protocol	Enables a protocol client to retrieve user profile data as well as data that is common among users. This protocol also enables a protocol client to manage other user profile properties, such as links, group (3) memberships, and colleagues. This protocol can be used to manage an enterprise social network and to enable users to search for people within their organization for communication purposes.	<a href="#">[MS-USPWS]</a>
User Profile Change Log Web Service Protocol	Allows client applications to query and synchronize the changes from a user profile store. This protocol provides querying for changes to specific user profiles or to all profiles on the server (2). The protocol is also designed to be used by search engines to incrementally <b>crawl</b> data from the user profile store.	<a href="#">[MS-USRPCH]</a>

The protocols in the following table enable the User Profile Service only in SharePoint 2010 Products.

Protocol name	Description	Short name
User Profile Social Data Web Service Protocol	Enables a protocol client to add, update, remove, and retrieve <b>social tags</b> , notes, and <b>social ratings</b> on documents and Web pages that are stored on the server (2).	<a href="#">[MS-UPSDWS]</a>
SharePoint Activity Feed Web Interfaces Protocol	Enables a protocol client to retrieve <b>social networking</b> activity data for the users of a Web site (2).	<a href="#">[MS-SPAFWI]</a>

### 2.2.2.5 Search Services

The protocols in the following table enable search services in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short name
Search Protocol	Allows a protocol client to issue <b>search queries</b> to a protocol server and to then receive the results.	<a href="#">[MS-SEARCH]</a>
Site Data 2003 Web Service Protocol	Supports the creation of site indexes by external <b>index servers</b> . This protocol allows the creation of indexes from earlier implementations of SharePoint Products and Technologies.	<a href="#">[MS-SITED3S]</a>
Site Data Web Service Protocol	Supports the creation of site indexes by external indexing services.	<a href="#">[MS-SITEDATS]</a>
SPSCrawl Web	Allows protocol clients to read the value of any item within the	<a href="#">[MS-</a>

Protocol name	Description	Short name
Service Protocol	context of a site or service application.	<a href="#">SPSCRWL</a>

### 2.2.2.6 Business Connectivity Services

The protocol in the following table enables Business Connectivity Services (BCS) in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short Name
Business Data Catalog Data Web Service Protocol	Describes an interface that protocol clients can use to search a <b>line-of-business (LOB) system</b> for an <b>EntityInstance</b> that matches a user input string.	<a href="#">[MS-BDCDP]</a>
Business Data Catalog Metadata Web Service Protocol	Describes an interface for protocol clients to retrieve information about interfaces of other line-of-business applications and annotations of these interfaces.	<a href="#">[MS-BDCMP]</a>

The protocols and structures in the following table enable Business Connectivity Services only in SharePoint 2010 Products.

Protocol or structure name	Description	Short name
Business Data Connectivity Remote Administration Web Service Protocol	Enables protocol clients to create, store, and retrieve information about the interfaces of LOB systems and the annotations of these interfaces.	<a href="#">[MS-BDCRAWPS]</a>
Business Connectivity Services Deployment Package File Format	Describes the content of a deployment package (2) for a BCS solution; the extensions that are specific to BCS for the Visual Studio Tools for Office ClickOnce Schema, as described in <a href="#">[MS-OSCO]</a> ; and a file format that is specific to BCS for representing the properties of an <b>external list</b> .	<a href="#">[MS-BCSDPFFS]</a>
Office Server ClickOnce Manifest Structure	Used to create a <b>deployment package (2)</b> for deploying one or more customizations and their related components to a computer.	[MS-OSCO]
Business Connectivity Services Deployment Web Service Protocol	Describes the utility functions that can be used in the creation of development or administration tools and applications interacting with BCS solutions.	<a href="#">[MS-BCSDWPS]</a>
External Content Type Picker Web Service Protocol	Enables protocol clients to find and resolve EntityInstances on a protocol server.	<a href="#">[MS-ECTPWPS]</a>

### 2.2.2.7 Content Management Service

The protocols in the following table enable content management in Microsoft® SharePoint 2007 Products and Technologies and in Microsoft® SharePoint® 2010 Products and Technologies.

Protocol name	Description	Short Name
Copy Web Service Protocol	Provides a method for copying files—along with their associated properties, such as the creation date and author—to one or more locations on a single server (2) or on multiple servers (2) running SharePoint Products and Technologies.	<a href="#">[MS-COPYS]</a>
Document Transforms Service Protocol	Enables a protocol client to call a file conversion service that converts a file from one format to another.	<a href="#">[MS-DOCTRANS]</a>
Official File Web Service Protocol	Enables the submission of files to a repository for the purpose of long-term retention according to a records management policy.	<a href="#">[MS-OFFICIALFILE]</a>
Publishing Web Service Protocol	Provides access to functionality that is related to publishing and content management. This protocol, which is used in Web content management scenarios, allows for the control of page layout, page status, and multilingual translation.	<a href="#">[MS-PUBWS]</a>
Versions Web Service Protocol	Enables a protocol client to programmatically view and manage the versions of files on the server (2). This protocol enables retrieving all the versions of a file, deleting the previous versions of a file, and restoring a file to a previous version.	<a href="#">[MS-VERSS]</a>
SpellCheck Web Service Protocol	Enables a protocol client to check the spelling of a set of words.	<a href="#">[MS-SPLCHK]</a>

The protocols in the following table enable content management only in SharePoint 2010 Products.

Protocol name	Description	Short name
Metadata Shared Service WCF Service Protocol	Describes a set of operations for reading, writing, and managing metadata in a centrally managed metadata store.	<a href="#">[MS-EMMWCF]</a>
Microsoft Enterprise Managed Metadata Web Service Protocol	Describes a set of operations that enable protocol clients to read, write, and manage metadata in a centrally managed metadata store.	<a href="#">[MS-EMMWS]</a>
Template Discovery Web Service Protocol	Enables a protocol client to retrieve links to the document templates that are associated with a document library on the protocol server.	<a href="#">[MS-TMPLDISC]</a>

### 2.2.2.8 PerformancePoint Services

The protocols in the following table enable PerformancePoint Services in Microsoft® SharePoint® Server 2010.

Protocol name	Description	Short name
PerformancePoint Services Authoring Service Protocol	Describes a Web service that enables reading, writing, and performing basic operations on dashboards that are hosted on a protocol server. The methods of this protocol center mainly on the ability to author and edit such dashboards.	<a href="#">[MS-PPSAS]</a>
PerformancePoint Services	Allows a protocol client to perform contribution analysis on a data value by using a data source or data source context that is known	<a href="#">[MS-PPSDECO]</a>

Protocol name	Description	Short name
Decomposition Tree Protocol	to the protocol server. This protocol allows a protocol client to retrieve either grouped, named actions that apply to a specific data value or named properties and values that are associated with that node, and to then retrieve additional values that contribute to the value. The protocol server returns data points and provides or denies summary data for the data points that it does not return.	

### 2.2.2.9 Analytics Services

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

Protocol name	Description	Short name
Microsoft Office SharePoint Server (MOSS) Analytics Reporting Protocol Specification	Provides methods for logging events for analyzing and retrieving analytical data.	<a href="#">[MS-MAR]</a>

### 2.2.2.10 Office Online Web Query Service

The protocol in the following table enables Office Online Web Query.

Protocol name	Description	Short name
Office Online Web Query Web Service Protocol	Allows a protocol client, such as Microsoft® SharePoint® 2010 Products and Technologies, to enumerate details about files in the Office Online repository.	<a href="#">[MS-OOWQS]</a>

## 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 Front-End Protocols system:

- The Office Client Protocols system, as described in [\[MS-OCPROTO\]](#).

### 2.3.2 Dependencies on Other Systems/Components

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

- The SharePoint Back-End Protocols system, as described in [\[MS-SPBEPO\]](#)
- The Windows SharePoint Services system, as described in [\[MS-WSSO\]](#)
- The Windows system, as described in [\[MS-SYS\]](#) section 1
- The Active Directory system, as described in [\[MS-ADTS\]](#) section 1

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

- The Windows Server® 2003 operating system with Service Pack 1
  - Internet Information Services (IIS) 6.0
- The Microsoft® .NET Framework version 3.0
  - ASP.NET 2.0
- Microsoft® SQL Server® 2000 with Service Pack 3a

Microsoft® SharePoint® 2010 Products and Technologies depends on the following systems and components to function:

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

## 2.4 Assumptions and Preconditions

The SharePoint Front-End Protocols system necessitates the following assumptions and preconditions:

- The protocol clients can access the front-end Web servers running SharePoint Products and Technologies via one or more established IP addresses.
- The functional components of the servers (2) running SharePoint Products and Technologies are started collectively, and the front-end Web servers running SharePoint Products and Technologies accept requests from the protocol clients.
- The front-end Web servers can access the back-end database servers and have the appropriate permissions to access data in the content and configuration databases.
- The front-end Web servers and the back-end Web servers are running either the same versions of SharePoint Products and Technologies or versions that are within an acceptable range. For more information, see [\[MS-WSSO\]](#) section 2.6.
- If Active Directory® Domain Services (AD DS) provides authentication (2), the **directory service (DS)** is accessible to the servers (2) running SharePoint Products and Technologies. All the necessary ports and gateways of any intermediate firewalls, routers, or connection points between components of the system are open for communication.

The scope of this list of assumptions and preconditions is intended to be implementation-independent and limited to the system level.

## 2.5 Use Cases

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

Core platform use cases:

- Create a site collection
- Create a site
- Create a list and add a view
- Get list data
- Create a file in a SharePoint library from a client computer
- Export and import site content
- Add a Web Part to a page and update its properties

Extended capabilities use cases:

- Excel® Services: Publish a workbook to a SharePoint library
- Excel Services: Interact with a workbook by using a browser
- Excel Services: Interact with a workbook by using an application
- Microsoft® Access Services: Publish a database application to SharePoint Products and Technologies
- Access Services: Interact with a database application that has been published to SharePoint Products and Technologies
- InfoPath® Forms Services: Publish a form template and enable browser rendering
- User Profile Service: Retrieve a user's profile
- User Profile Service: Tag a document
- Search services: Issue a search query
- BCS: Create an external content type
- BCS: Read an external content type
- BCS: Update an external content type
- BCS: Deploy a BCS solution
- BCS: Update a BCS solution
- Content Management Service: Get term set data and add term
- PerformancePoint Services in Microsoft® SharePoint® Server 2010: Interact with published dashboards on a Web Part page
- Analytics Services: Log an event and retrieve data

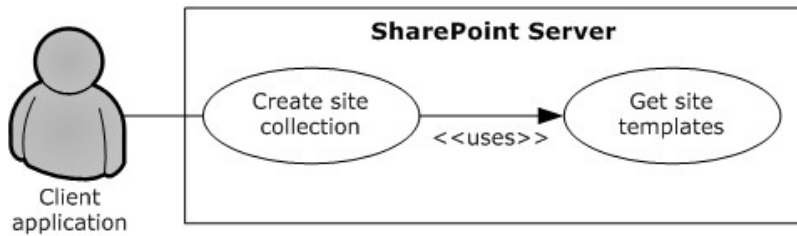
These use cases are not intended to provide a thorough and complete model of the system for any implementation.

## 2.5.1 Core Platform Use Cases

### 2.5.1.1 Create a Site Collection

This use case describes how a client application creates a new site collection. The client application specifies a site definition or site template to use when creating the new site collection.

The following diagram illustrates this process.



**Figure 3: Process for creating a site collection**

#### Preconditions

- The client application can communicate with the Central Administration site.
- The client application has been authenticated as a member of the Farm Administrators group (2).

#### Steps

1. The client application requests a list (1) of available site definitions and site templates from the server (2).
2. The server (2) returns the list (1) of available site definitions and site templates.
3. The client application selects a Web template and specifies the URL, title, and description for the site collection.
4. The client application calls the method to create a new site collection using the specified information.
5. The server (2) creates a site collection with the specified information.

#### Error

- If the specified URL is invalid or already in use, the server (2) will not create a new site collection.

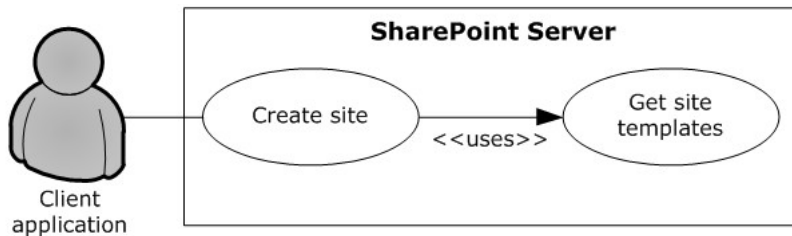
#### Post-conditions

- A site collection is created at the URL that was specified by the client application.
- The site collection is not created, because the specified URL is invalid or already in use.

### 2.5.1.2 Create a Site

This use case describes how a client application creates a new site (2). The client application specifies a site definition or site template to use when creating the new site (2).

The following diagram illustrates this process.



**Figure 4: Process for creating a site**

#### Preconditions

- The client application can communicate with the SharePoint site.
- The client application has been authenticated as a member of the Site Owner group (2).

#### Steps

1. The client application requests a list (1) of available site definitions and site templates from the server (2).
2. The server (2) returns the list (1) of available site definitions and site templates.
3. The client application selects a Web template and specifies the URL, title, and description for the site (2).
4. The client application calls the method to create a new site using the specified information.
5. The server (2) creates a site (2) with the specified information.

#### Error

- If the specified URL is invalid or already in use, the server (2) will not create a new site collection.

#### Post-conditions

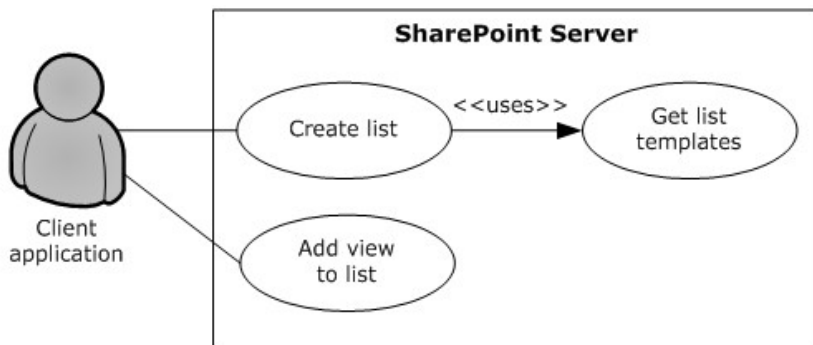
- A site (2) is created at the URL that was specified by the client application.
- The site (2) is not created, because the specified URL is invalid or already in use.

### 2.5.1.3 Create a List and Add a View

This use case describes how a client application creates a new list (1) and adds a view. The client application specifies the list template to use when creating the new list (1).

The following diagram illustrates this process.





**Figure 5: Process for creating a list (1) and adding a view**

**Preconditions**

- The client application can communicate with the SharePoint site.
- The client application has been authenticated as a member of the Contributor group (2).

**Steps**

1. The client application requests a list (1) of list templates from the server (2).
2. The server (2) returns the list (1) of list templates.
3. The client application selects a list template and specifies the title and description for the new list (1).
4. The client application calls the method to create a new list (1) and passes the specified information.
5. The server (2) creates a list (1) with the specified information.
6. The client application selects the list (1) that was created and specifies a view name and view fields.
7. The client application calls the method to add a view to the list (1) and passes the specified information.
8. The server (2) adds a view to the list (1).

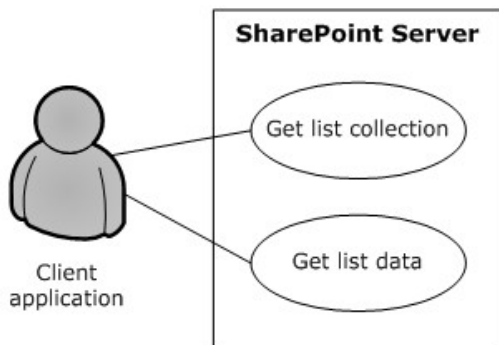
**Post-conditions**

- A new list (1) exists on the site (2).
- The new list (1) contains the new view.

**2.5.1.4 Get List Data**

This use case describes how a client application gets list data for display, building reports, or other common scenarios.

The following diagram illustrates this process.



**Figure 6: Process for getting list (1) data**

**Preconditions**

- The client application can communicate with the SharePoint site.
- The client application has been authenticated and has read permission on the list (1).

**Steps**

1. The client application requests a list of the lists (1) that exist on the site (2) from the server (2).
2. The server (2) returns the list of lists (1) on the site (2).
3. The client application selects a list (1).
4. The client application requests data for the selected list (1).
5. The server (2) returns the data for the list (1).

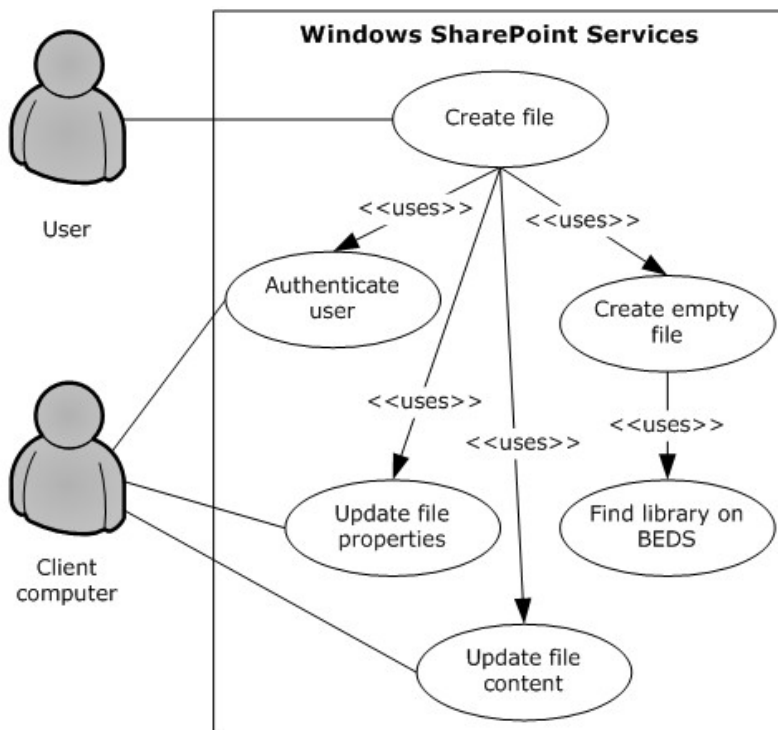
**Post-condition**

- The selected list (1) is displayed.

**2.5.1.5 Create a File in a SharePoint Library from a Client Computer**

This use case describes the simplest way to create a file by using the SharePoint Front-End Protocols system. The actor is the user, who creates a text file named hello.txt that contains the text "hello" in a SharePoint library.

The following diagram illustrates this file creation process.



**Figure 7: Process for creating a file in a SharePoint library from a client computer**

### Preconditions

- The user has permissions for an existing SharePoint library.
- The user is logged on to a client computer running the Windows® 7 operating system with an authenticated Microsoft® Windows® session and can access the Windows® SharePoint® Services 3.0 or Microsoft® SharePoint® Foundation 2010 site that contains the library. (Client computers running other versions of Windows might have different steps.)
- In the **Command Prompt** window, the user types the following command: **echo hello**  
`>\\server\site\doclib\hello.txt` (where *server* is the name of the server (2), *site* is the name of the site (2), and *doclib* is the name of the document library).

### Steps

1. The front-end Web server running Windows SharePoint Services 3.0 or SharePoint Foundation 2010 authenticates the user.
2. Windows SharePoint Services 3.0 or SharePoint Foundation 2010 finds the location of the document library on the back-end database server ("BEDS" in the figure) and verifies that the user has access to it.
3. Windows SharePoint Services 3.0 or SharePoint Foundation 2010 creates an empty file in the library and confirms success to the client computer.
4. The client computer updates the file properties and file contents.

### Errors

- If the user does not have permissions, the client computer notifies the user that access was denied.
- If the client computer cannot connect to the front-end Web server, the client computer notifies the user of the error.
- If the client computer cannot update the file properties or file contents, the client computer notifies the user of the error.

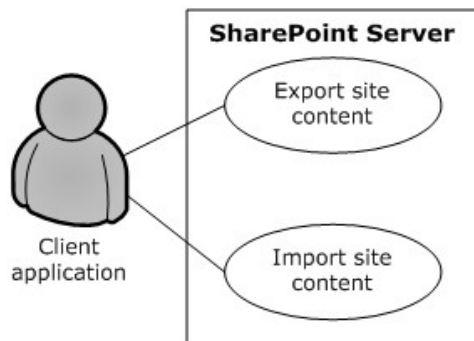
#### Post-conditions

- 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.

### 2.5.1.6 Export and Import Site Content

This use case describes how a client application can export and import site content. Client applications export and import site content to migrate content from one site to another.

The following diagram illustrates the export and import process.



**Figure 8: Process for exporting and importing site (2) content**

#### Preconditions

- The client application can communicate with the SharePoint site.
- The client application has been authenticated to the server (2), has open Web permissions and manage Web permissions for the source site (2), and has site (2) creation permissions for the target site (2).

#### Steps

1. The client application calls the method to export the site and passes the URL of the site (2) to export, the file name for the content migration package, and the directory in which to place the package.
2. The server (2) packages the site (2) content and places the content migration package in the specified directory.
3. The client application calls the method to import the site (2) and passes the URL for the resulting site (2) and the URL of the content migration package.
4. The server (2) imports the content from the content migration package to the specified URL.

## Errors

- If a site (2) at a specified URL is not accessible, the server (2) will return an error.
- If the directory from which to write and read the content migration package is not accessible, the server (2) will return an error.
- If the client application lacks sufficient permissions to export or import the site (2), the server (2) will return an error.

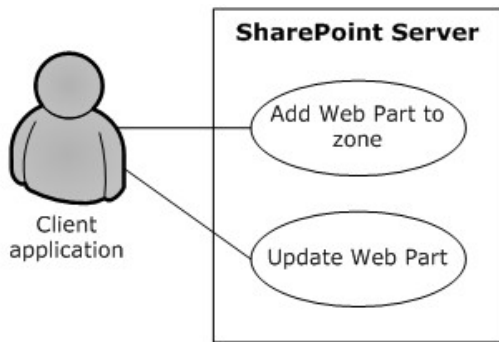
## Post-conditions

- The destination site (2) contains all the content from the content migration package.
- The content is not exported or imported, because an error occurred.

### 2.5.1.7 Add a Web Part to a Page and Update Its Properties

This use case describes how a client application adds a Web Part to a page and updates the properties of that Web Part. Web Parts function as a mechanism for customizing the presentation of a Web site (2).

The following diagram illustrates the Web Part add and update process.



**Figure 9: Process for adding a Web Part to a page and updating its properties**

## Preconditions

- The client application can communicate with the SharePoint site.
- The SharePoint site has a page that contains Web Part zones.
- The client application has been authenticated as a member of the Contributor group (2).
- The Web Part to add is available on the server (2).

## Steps

1. The client application specifies the URL of a page, the Web Part zone, the Web Part zone index, and the markup for the Web Part.
2. The client application calls the method to add a Web Part to a Web Part zone.
3. The server (2) adds the Web Part to the Web Part zone.

4. The server (2) returns the **GUID** that uniquely identifies the newly added Web Part.
5. The client application calls the method to update the Web Part and passes the same URL of the page, the GUID that identifies the Web Part, and the new properties.
6. The server (2) updates the properties of the Web Part.

**Error**

- If the Web Part markup is invalid, the server (2) will return an error.

**Post-conditions**

- The page contains a new Web Part with updated properties.
- The page does not contain a new Web Part with updated properties, because the Web Part markup is invalid.

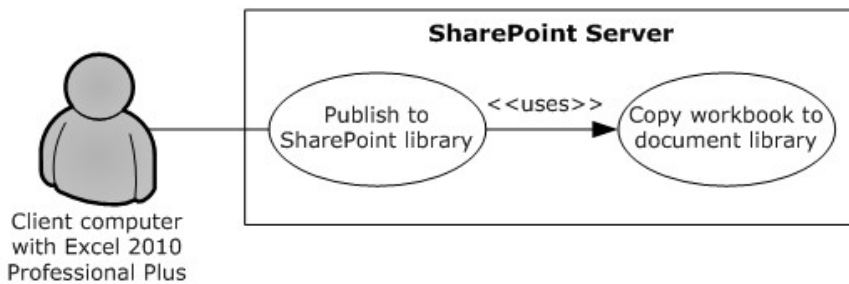
**2.5.2 Extended Capabilities Use Cases**

**2.5.2.1 Excel Services**

**2.5.2.1.1 Publish a Workbook to a SharePoint Library**

This use case describes how a client application publishes (1) a workbook to a SharePoint library.

The following diagram illustrates this process.



**Figure 10: Process for publishing a workbook to a SharePoint library**

**Preconditions**

- Excel® Services in Microsoft® SharePoint® Server 2010 is enabled on a farm running SharePoint Products and Technologies.
- The user has created a workbook in Microsoft® Excel® 2010.
- The user has permission to save files to SharePoint Products and Technologies.

**Steps**

1. The user initiates publishing from Excel 2010.
2. Upon a request from Excel 2010, SharePoint Products and Technologies creates a file in a SharePoint library.

3. The workbook opens in a browser by using a URL that is specified according to the Excel Services Publishing Protocol, as described in [\[MS-ESURL\]](#).

### Errors

- If the user does not have permission to create content in a SharePoint library, an error will be returned.
- If the workbook exceeds the size limit set in SharePoint Central Administration, an error will be returned.
- If the workbook contains unsupported features, the workbook will not open in the browser and an error will be returned.

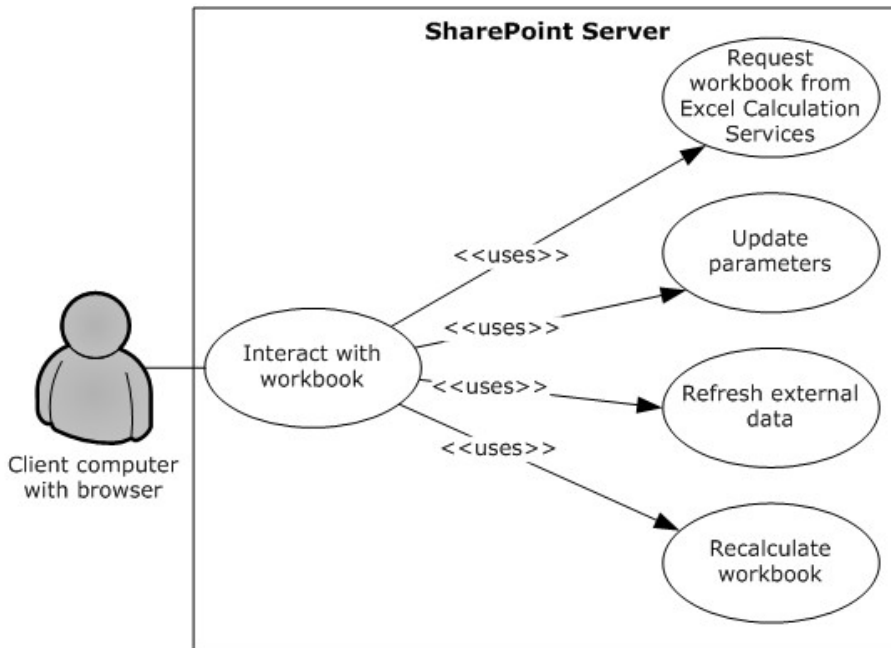
### Post-conditions

- A file exists in a SharePoint library.
- A file does not exist in a SharePoint library, because an error occurred.

#### 2.5.2.1.2 Interact with a Workbook by Using a Browser

This use case describes how a browser uses Excel® Services in Microsoft® SharePoint® Server 2010 to interact with a workbook that has been published (1) to SharePoint Products and Technologies.

The following diagram illustrates this process.



**Figure 11: Process for interacting with a workbook by using a browser**

### Preconditions

- Excel Services 2010 is enabled on a farm running SharePoint Products and Technologies.

- The user has permissions for the requested workbook.
- The user is running a supported browser.

### **Steps**

1. The user requests the workbook by either clicking it in the UI of a SharePoint library, typing the URL of the file in the browser, or opening a Web page that displays all or part of the workbook.
2. The viewable range of the workbook appears in the browser.
3. If parameters exist, the user can enter the values in the parameter pane.
4. If external data connections (2) exist, the user can request that the data be refreshed.
5. The user can request that the workbook be recalculated.

### **Errors**

- If the user does not have permissions for the workbook, an error will be returned.
- If the workbook contains unsupported features that prevent it from opening, an error will be returned.
- If the workbook is connected to external data for which the user does not have permissions, an error will be returned.

### **Post-conditions**

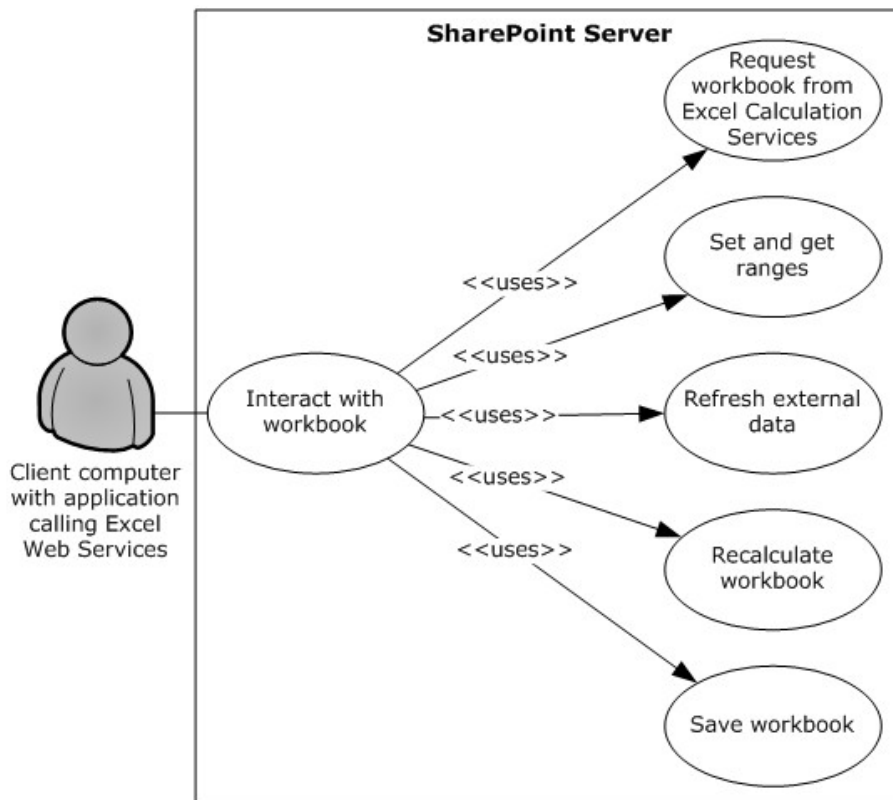
- The values of the workbook are recalculated according to the changes, and the workbook is displayed in the browser.
- The workbook is not displayed in the browser, because an error occurred.

#### **2.5.2.1.3 Interact with a Workbook by Using an Application**

This use case describes how an application uses Excel® Services in Microsoft® SharePoint® Server 2010 to interact with a workbook that has been published (1) to SharePoint Products and Technologies.

The following diagram illustrates this process.





**Figure 12: Process for interacting with a workbook by using an application**

**Preconditions**

- Excel Services 2010 is enabled on a farm running SharePoint Products and Technologies.
- The user who is running the application has permissions for the requested workbook.

**Steps**

1. The application requests the workbook by using the URL of the file.
2. The application can get a range of data from the workbook.
3. The application can set a range of data in the workbook.
4. The application can refresh external data in the workbook.
5. The application can recalculate the workbook.
6. The application can save changes made to the workbook back to the file that is stored in SharePoint Products and Technologies.
7. The application can close the workbook.

**Errors**

- If the user who is running the application does not have permissions for the workbook, an error will be returned.
- If the workbook contains unsupported features that prevent it from opening, an error will be returned.
- If the workbook is connected to external data for which the current user does not have permissions, an error will be returned.

#### Post-conditions

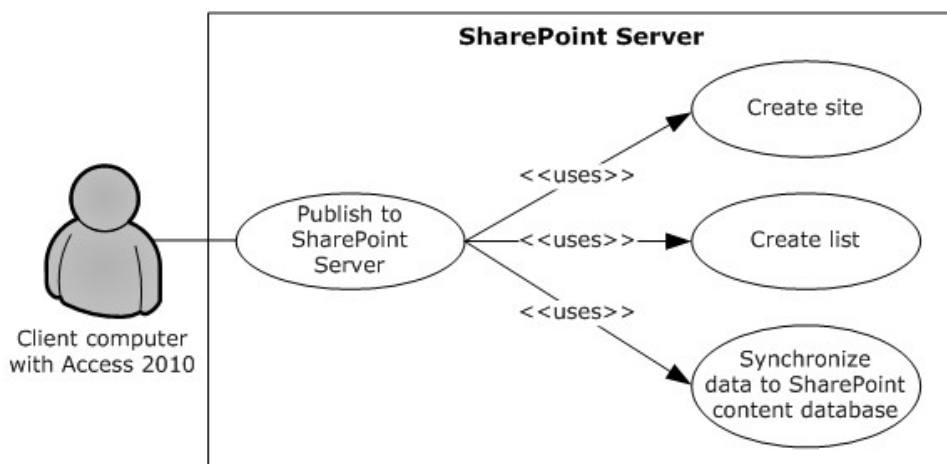
- The application has retrieved any necessary data from the workbook.
- The workbook has been updated as requested by the application.
- The workbook has not been updated, because an error occurred.

### 2.5.2.2 Access Services

#### 2.5.2.2.1 Publish a Database Application to SharePoint Products and Technologies

This use case describes how a client application publishes (1) a database application to SharePoint Products and Technologies.

The following diagram illustrates this process.



**Figure 13: Process used to publish (1) a database application to SharePoint Products and Technologies**

#### Preconditions

- Microsoft® Access Services is enabled on a farm running SharePoint Products and Technologies.
- The user has built a database application in Microsoft® Access® 2010.
- The user has been authenticated and has permission to create a site.

#### Steps

1. The user initiates publishing from Access 2010.

2. Upon a request from Access 2010, Access Services creates a SharePoint site according to the Access Services Site Template, as described in [\[MS-ASWS\]](#) section 3.1.1.1.
3. Upon a request from Access 2010, Access Services creates SharePoint lists to represent tables in the Access 2010 database application.
4. Upon a request from Access 2010, Access Services synchronizes the database object structures.
5. Upon a request from Access 2010, Access Services synchronizes data from the tables to the SharePoint lists.
6. Upon a request from Access 2010, Access Services creates pages in SharePoint Products and Technologies for the forms (2) and reports of the Access 2010 database application. Upon user request, the pages will be rendered in HTML in the browser.

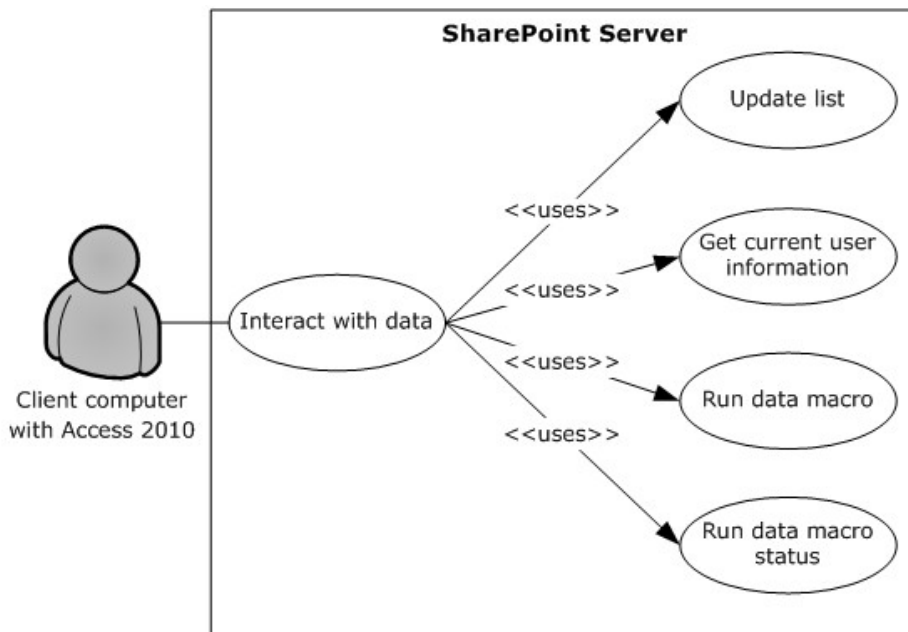
**Post-condition**

- A SharePoint site that contains the published (1) application has been created.

**2.5.2.2.2 Interact with a Database Application that Has Been Published to SharePoint Products and Technologies**

This use case describes how a client application interacts with a database application that has been published (1) to SharePoint Products and Technologies.

The following diagram illustrates this process.



**Figure 14: Process for interacting with a database application that has been published to SharePoint Products and Technologies**

**Preconditions**

- Microsoft® Access Services is enabled on a farm running SharePoint Products and Technologies.

- A user can use Microsoft® Access® 2010 to interact with a database application that has been published (1) to SharePoint Products and Technologies.

### Steps

1. After a database application created with Access 2010 has been published (1) to SharePoint Products and Technologies, the data is stored in the content database. A user can still update the data by using Access 2010 through Web services, as described in [\[MS-ASWS\]](#). Upon a request from Access 2010, Access Services updates the SharePoint lists that store the data.
2. Access 2010 can retrieve the current user information from Access Services.
3. A **data macro** can be triggered from Access 2010. Access Services will run the data macro on SharePoint Products and Technologies and track the status of the data macro instance.

### Error

- If the specified list (1) to be updated does not exist, the server (2) will return an error.

### Post-conditions

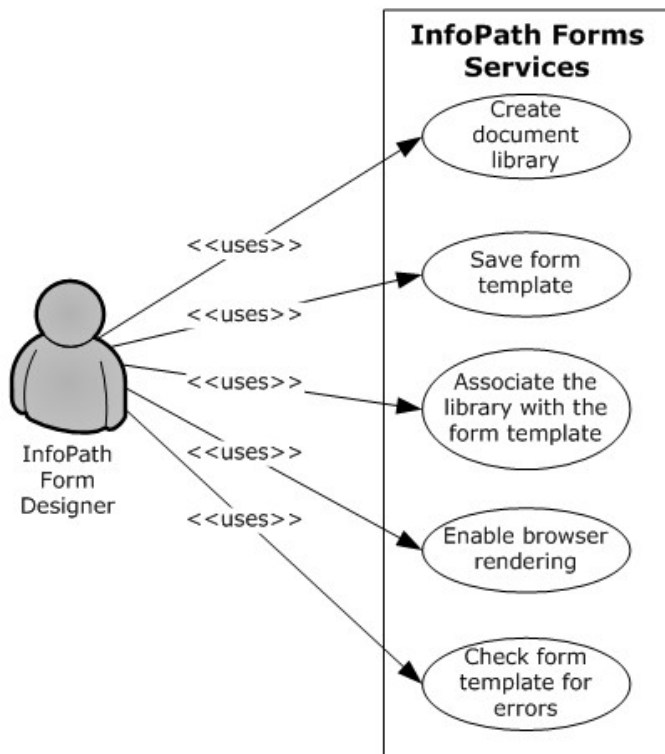
- The data operation that was performed in Access 2010 is synchronized to the SharePoint site that contains the application.
- The data macro that was triggered in Access 2010 is run on the farm running SharePoint Products and Technologies.

## 2.5.2.3 InfoPath Forms Services

### 2.5.2.3.1 Publish a Form Template and Enable Browser Rendering

This use case describes how a user both publishes (1) a form template to a SharePoint Products and Technologies-hosted document library by using the InfoPath Form Designer client application, and enables browser rendering for that form template by means of InfoPath® Forms Services. This allows other end users to fill out such a form using their Web browsers and submit that information back to the SharePoint Products and Technologies server for review, analysis or further action but the original publishing user.

The following diagram illustrates this process.



**Figure 15: Process used to publish (1) a form template and enable browser rendering**

#### Preconditions

- InfoPath Forms Services is enabled.
- The client application can communicate with the SharePoint site.
- The user has been authenticated as a member of the designer group (2).

#### Steps

1. The user initiates publishing from Microsoft® InfoPath® 2010.
2. Upon a request from InfoPath Designer, SharePoint Products and Technologies responds with information about the site (2) that is provided by the user.
3. The user provides parameters such as a **form library** name.
4. Upon a request from InfoPath Designer, SharePoint Products and Technologies creates a document library.
5. InfoPath Designer saves the form template to the document library.
6. Upon a request from InfoPath Designer, SharePoint Products and Technologies associates the SharePoint library with the template.
7. Upon a request from InfoPath Designer, InfoPath Forms Services enables browser rendering for the form template (.xsn) file.

## Error

- If an error occurs during the browser-enable process, InfoPath Designer will issue a call to check the form template for errors.

## Post-conditions

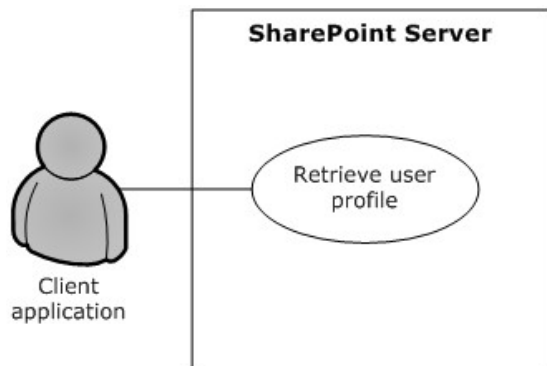
- A published form template is accessible by using the Web browser at a URL that is associated with the document library.
- A published form template is inaccessible, because an error occurred during the browser-enable process.

## 2.5.2.4 User Profile Service

### 2.5.2.4.1 Retrieve a User's Profile

This use case describes how a client application can retrieve a user's profile—that is, all the information in the user's profile that is accessible to the client application. The information that is presented to the client application is based on the privacy level set on the user profile properties, either by the administrator as a default or by the user. For example, a user can set the privacy settings so that only colleagues will be able to see the user's birthday.

The following diagram illustrates the retrieval process.



**Figure 16: Process for retrieving a user's profile**

## Preconditions

- The client application has permissions for accessing and reading 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 the properties of the user profile, filtered by the privacy settings.
3. The User Profile Service returns the properties to the client application, formatted in the site template that has been applied to the user's personal profile site.

## Error

- If the client application does not have access and read permissions for the user's profile, an access denied error will be returned.

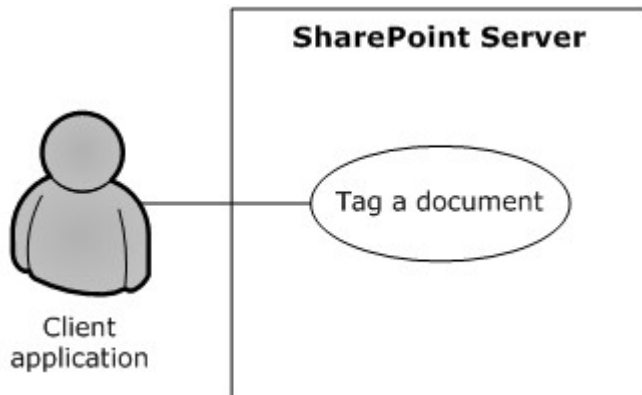
## Post-conditions

- The client application displays the user's profile, which typically includes a picture, information about the user, and any additional properties that have been entered and shared by the user.
- The user's profile is not displayed, because an access denied error was returned.

### 2.5.2.4.2 Tag a Document

This use case describes how a client application applies a tag to a document.

The following diagram illustrates this process.



**Figure 17: Process for tagging a document**

## Preconditions

- The client application has permission to access the location of the document.
- The client application has permission to apply tags by using the User Profile Service.
- The Content Management Service is provisioned and operational. For more information, see section [2.1.3.7](#).

## Steps

1. The client application requests matching terms from the term store of the Content Management Service by providing the initial letters of the tag (for example, "soc" for "social computing" as an intended tag).
2. The Content Management Service returns a set of suggested terms to the client application, as described in section [2.5.2.7.1](#).
3. Upon the acceptance of a suggested term, a GUID for the term is stored in the **user profile store** along with the location of the document, information about the client application's identity, and the time of tag creation.

## Error

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

#### Post-conditions

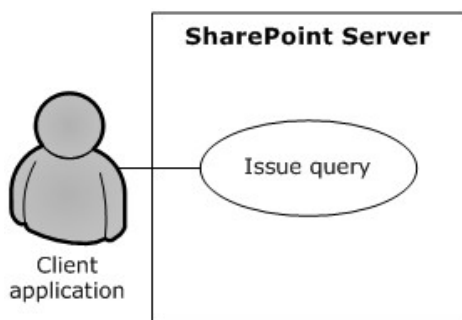
- The client application detects the tag that is applied to the document location.
- The client application cannot tag a document, because tagging access was disabled.

### 2.5.2.5 Search Services

#### 2.5.2.5.1 Issue a Search Query

This use case describes how a client application retrieves the search results for a search query that it issued against a SharePoint site. After the client application receives the results, common uses include displaying them in the UI and allowing the user to select a particular result to view more information.

The following diagram illustrates the query process.



**Figure 18: Process for issuing a search query**

#### Preconditions

- The client application has permissions for accessing and executing queries against the SharePoint site.
- The SharePoint site is configured to use search services from a search service application.

#### Steps

1. The client application requests the results for a particular query.
2. The search service retrieves the URLs and properties that satisfy the query.
3. The search service returns the results to the client application in the format that is specified by the configuration of the Search Center.

#### Errors

- If the client application does not have permissions for the SharePoint site, an access denied error will be returned.
- If the SharePoint site is not set up to use the search services from a search service application, an error will be displayed in the UI when the client application issues a query.



**Post-conditions:**

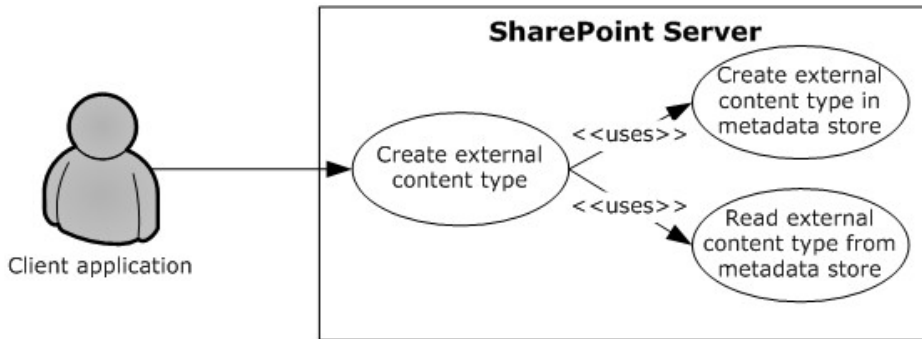
- The client application displays the search results along with properties such as the URL, author, last modified date, file size, title, and summary.
- The search results are not displayed, because an error was returned.

**2.5.2.6 Business Connectivity Services**

**2.5.2.6.1 Create an External Content Type**

This use case describes how a client application creates an external content type and publishes (1) to the BDC metadata store on Microsoft® SharePoint® Server.

The following diagram illustrates this process.



**Figure 19: Process for creating an external content type**

**Preconditions**

- The Business Data Connectivity service is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated as a member of the designer group (2).

**Steps**

1. The protocol client calls the method to create an external content type.
2. The protocol server will establish a connection to the metadata store and create the external content type.
3. The protocol server will return the new external content type to the protocol client.

**Errors**

- If the client application does not have permissions for the SharePoint site, an access denied error will be returned.
- If the SharePoint site is not set up to use Business Connectivity Services (BCS) from a service application, an appropriate error will be returned.
- If the current user does not have permissions to update the BDC metadata store, an appropriate error will be returned.

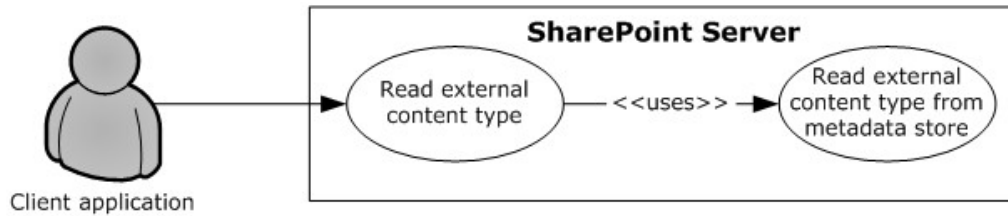
### Post-conditions

- The external content type is successfully published (1) to the BDC metadata store.
- The external content type is not published (1) to the BDC metadata store, because an error was returned.

### 2.5.2.6.2 Read an External Content Type

This use case describes how a client application reads an external content type from the BDC metadata store on Microsoft® SharePoint® Server.

The following diagram illustrates this process.



**Figure 20: Process for reading an external content type**

### Preconditions

- The BDC service is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated as a member of the designer group (2).

### Steps

1. The protocol client calls the method to read an external content type.
2. The protocol server will establish a connection to the metadata store and read the external content type.
3. The protocol server will return the external content type to the protocol client.

### Errors

- If the client application does not have permissions for the SharePoint site, an access denied error will be returned.
- If the SharePoint site is not set up to use BCS from a service application, an appropriate error will be returned.
- If the current user does not have permissions to access the BDC metadata store, an appropriate error will be returned.

### Post-conditions

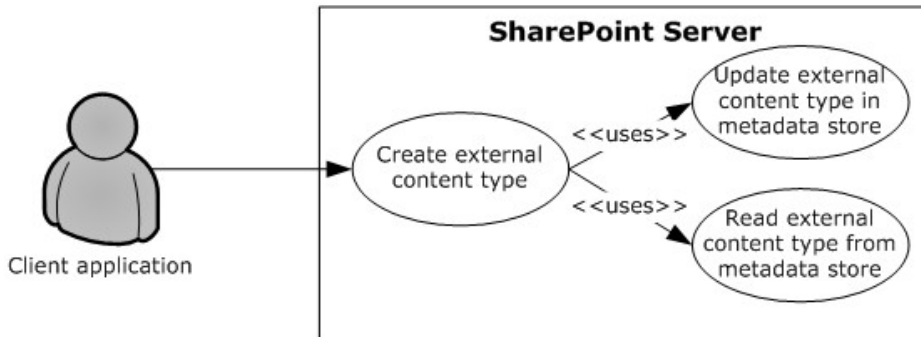
- The external content type is read and returned to the protocol client.

- The external content type is not read or returned to the protocol client, because an error was returned.

### 2.5.2.6.3 Update an External Content Type

This use case describes how a client application updates an external content type store in the BDC metadata store on Microsoft® SharePoint® Server.

The following diagram illustrates this process.



**Figure 21: Process for updating an external content type**

#### Preconditions

- The BDC service is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated as a member of the designer group (2).

#### Steps

1. The protocol client initiates a request for reading an External Content Type by calling the update method.
2. The protocol server will establish a connection to the metadata store and update the external content type.
3. The protocol server will return the updated external content type to the protocol client.

#### Errors

- If the client application does not have permissions for the SharePoint site, an access denied error will be returned.
- If the SharePoint site is not set up to use BCS from a service application, an appropriate error will be returned.
- If the current user does not have permissions to update the BDC metadata store, an appropriate error will be returned.

#### Post-conditions

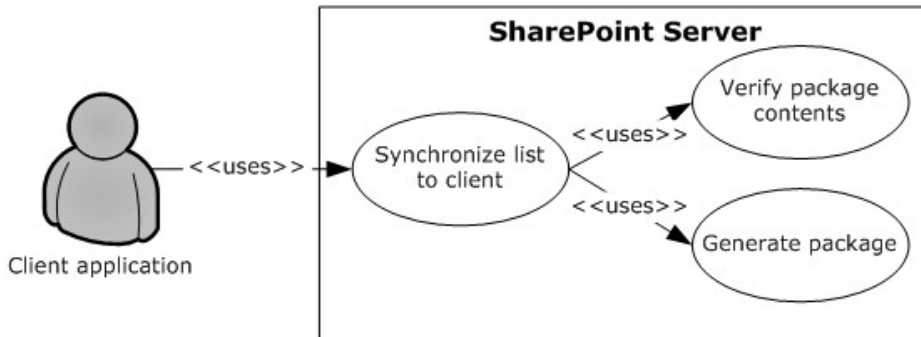
- The external content type is successfully updated in the BDC metadata store.

- The external content type is not updated in the BDC metadata store, because an error was returned.

#### 2.5.2.6.4 Deploy a Business Connectivity Services Solution

This use case describes how a client application can synchronize with a BCS solution published (1) in the form of a list (1) on Microsoft® SharePoint® Server.

The following diagram illustrates this process.



**Figure 22: Process for deploying a Business Connectivity Services solution**

#### Preconditions

- BCS is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated and has read access.

#### Steps

1. The protocol client initiates a request for synchronizing the list (1) with a client application.
2. The protocol server will verify the package existence and will verify that the external content type and the related external content types are current.
3. If they are not current, the protocol server will generate a new package with the current external content type and the related external content types.
4. The protocol server will return the package URL and the solution identifier to the protocol client.

#### Error

- If an error occurs during the package generation process, the protocol server will return an appropriate error code to the protocol client.

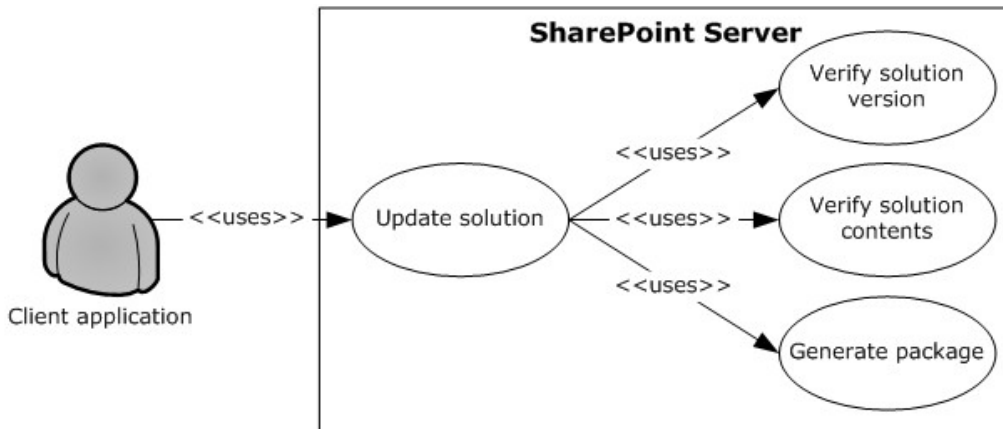
#### Post-conditions

- The solution package is accessible to the protocol client from a URL.
- The solution package is inaccessible to the protocol client from a URL, because an error occurred during the package generation process.

### 2.5.2.6.5 Update a Business Connectivity Services Solution

This use case describes how a client application can update a BCS solution published (1) in the form of a list (1) on Microsoft® SharePoint® Server.

The following diagram illustrates this process.



**Figure 23: Process for updating a Business Connectivity Services solution**

#### Preconditions

- BCS is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated and has read access.

#### Steps

1. The protocol client initiates a request for updating a solution.
2. The protocol server will verify the package existence and will verify that the external content type and the related external content types are current.
3. If they are not current, the protocol server will generate a new package with the current external content type and the related external content types.
4. The protocol server will return the package URL and the solution identifier to the protocol client.

#### Error

- If an error occurs during the package generation process, the protocol server will return an appropriate error code to the protocol client.

#### Post-conditions

- The solution package is accessible to the protocol client from a URL.
- The solution identifier returned to the protocol client is different, indicating a newer version of the solution is available.

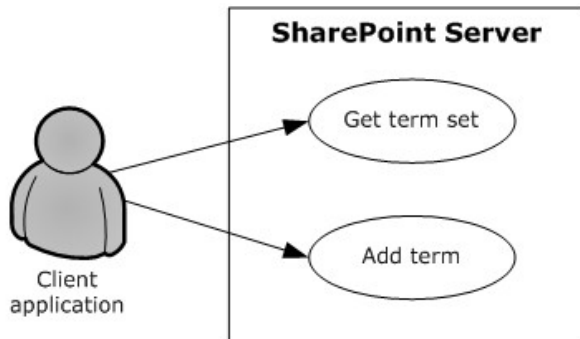
- The solution package is not updated, because an error occurred during the package generation process.

## 2.5.2.7 Content Management Service

### 2.5.2.7.1 Get Term Set Data and Add a Term

This use case describes how a client application first gets the data for a **term set** from a **term store** that is managed by a protocol server and then adds an additional term to that term set. After a client application gets the data, other common uses include displaying the term set and validating user-supplied data against the term set.

The following diagram illustrates the get and add process for a term set.



**Figure 24: Process for getting term set data and adding a term**

#### Preconditions

- The client application can communicate with the term store.
- The client application has been authenticated, has read permission and has write permission for the term store.
- The client application has the identifiers for the term sets in the term store.

#### Steps

1. The client application requests the term set from the term store.
2. The server (2) returns the term set.
3. The client adds a term to the term set in the specified term store.
4. The server (2) returns the added term.

#### Error

- If the term set is not open, the term store will not create a new term.

#### Post-conditions

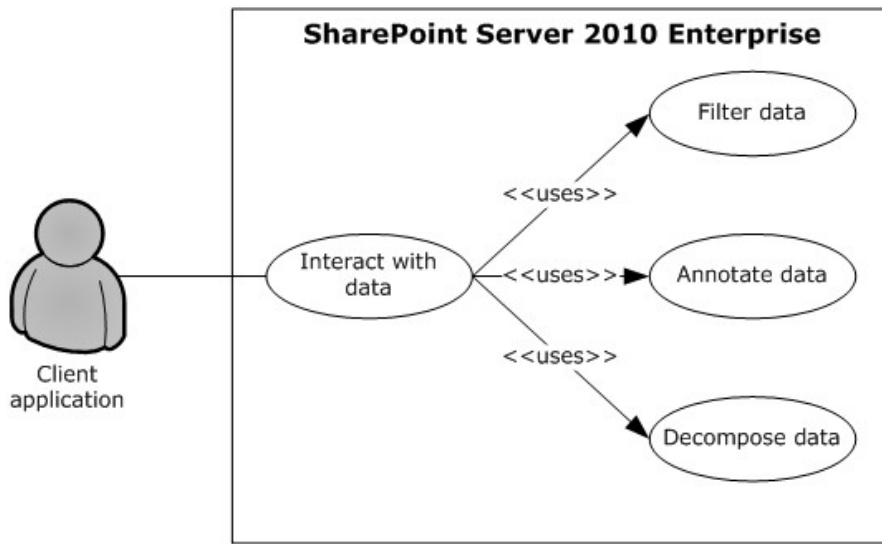
- A new term exists in the term set.
- A new term is not added to the term set, because the term set is not open.

## 2.5.2.8 PerformancePoint Services

### 2.5.2.8.1 Interact with Published Dashboards on a Web Part Page

This use case describes how a client application interacts with published (1) dashboards on a Web Part page in SharePoint Products and Technologies. A user can analyze business data (if the user has the appropriate permissions for the SharePoint list in which that data is stored); comment on the data; filter or run calculations on the data according to business rules; and view the data through the **decomposition tree**, which provides a visual view of how the data unfolds from a rolled-up OLAP hierarchy (as described in [\[MS-PPSDECO\]](#)).

The following diagram illustrates the process of interacting with published dashboards.



**Figure 25: Process for interacting with dashboards on a Web Part page**

#### Preconditions

- PerformancePoint Services in Microsoft® SharePoint® Server 2010 has been set up, and the site collection level feature has been started in Microsoft® SharePoint® Server 2010 Enterprise.
- A dashboard from PerformancePoint Services 2010 has been published (1) to SharePoint Server 2010 Enterprise.
- Microsoft® Silverlight® version 3.0 has been installed on the client computer. For more information, see [\[MS-SLXV\]](#).

#### Steps

1. The browser on the client computer requests a published (1) dashboard from SharePoint Server 2010 Enterprise.
2. PerformancePoint Services 2010 retrieves the requested objects from the SharePoint list service.
3. To return data to the rendered dashboard, PerformancePoint Services 2010 executes a data source query for each object that requires one.

4. The user drills through the rendered data, sending an **Asynchronous JavaScript + XML (AJAX)** request for each data request that the browser initiates.
5. In scorecard objects, the user can right-click a cell and annotate the value in the cell.
6. In charts and analytic grids, the user can analyze data in a decomposition tree that is rendered in Silverlight.

**Error**

- If a user does not have access to a dashboard item (for example, the analytic grid, chart, or scorecard) in the SharePoint list, that user will receive an error in the Web Part zone of the Web Part page.

**Post-conditions**

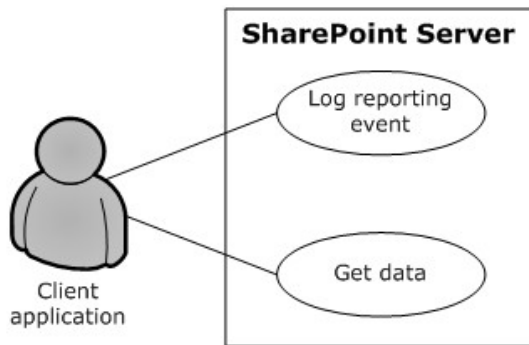
- The user views the result of the dashboard that was acted upon.
- The dashboard result cannot be viewed, because the user does not have access to the dashboard item.

**2.5.2.9 Analytics Services**

**2.5.2.9.1 Log an Event and Retrieve Data**

This use case describes how a client application logs an event for analysis and retrieves the analytical data for reporting purposes.

The following diagram illustrates this process.



**Figure 26: Process for logging an event and retrieving data**

**Preconditions**

- Analytics Services is enabled on the server (2).
- The client application can communicate with the SharePoint site.

**Steps**

1. The client application calls the method to log events when a search query is issued.
2. The server (2) returns the list (1) of events to be logged for analysis.



3. The client application requests data for the top search queries for a specified date.
4. The server (2) returns the top search queries for the specified date.

**Post-condition**

- A search query is displayed for the specified date.

## **2.6 Versioning, Capability Negotiation, and Extensibility**

The SharePoint Products and Technologies front-end and back-end servers (2) perform explicit version verifications.. For more information, see [\[MS-WSSO\]](#) section 2.6.

## **2.7 Error Handling**

The SharePoint Front-End Protocols system does not handle errors at the system level for cross-protocol error states. The individual protocol documents describe the errors that the protocols return and what they mean for the system. How to handle the errors, based on the protocol descriptions, is determined by the implementer.

## **2.8 Coherency Requirements**

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

## **2.9 Security**

A detailed overview of authentication (2), authorization, and the core concepts that are used to manage security in SharePoint Products and Technologies is available. For more information, 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 Front-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 new site (2) and access list data
- Create a file from a client computer

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 New Site and Access List Data

This example describes how to create a new site (2), create a new list (1), and access the list (1) data. This example builds on the use cases covered in "Create a Site" (section [2.5.1.2](#)), "Create a List and Add a View" (section [2.5.1.3](#)), and "Get List Data" (section [2.5.1.4](#)). In this example, a client application creates a new subsite, creates a new list (1) on that subsite, populates the list (1) with data, and gets a filtered set of list items.

This scenario assumes that:

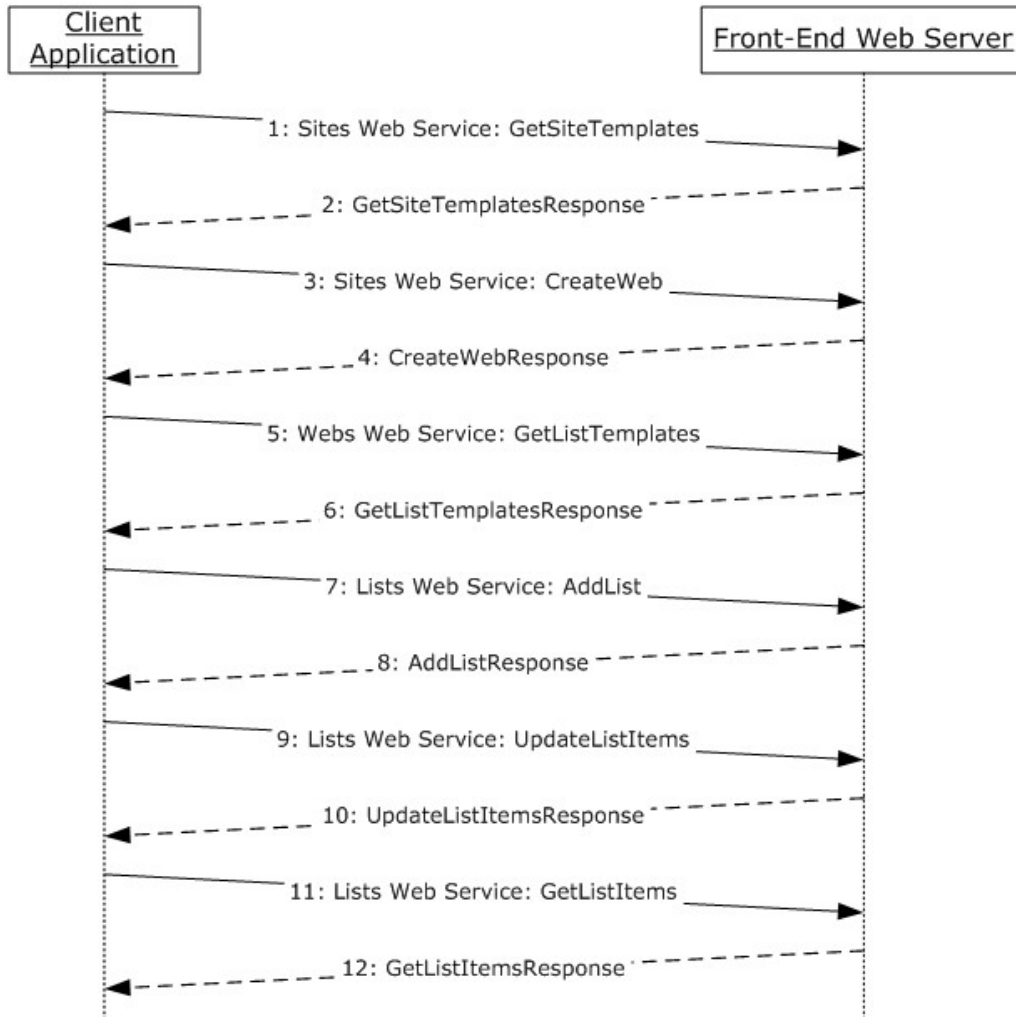
- The client application can communicate with the SharePoint site.
- The client application has permission to create a subsite, create a list (1), read list (1) items, and write list (1) items.

The following steps illustrate the site (2) creation and list access processes, assuming that no errors occur.

1. The client application calls the method in the Site Web Service Protocol (as described in [\[MS-SITESS\]](#)) that retrieves the list (1) of available site templates and site definitions.
2. The server (2) returns the collection of available site templates and site definitions.
3. The client application calls to the method in the Site Web Service Protocol that creates a subsite.
4. The server (2) returns the URL to the newly created subsite.
5. The client application uses the URL of the newly created subsite to call the method in the Webs Web Service Protocol (as described in [\[MS-WEBSS\]](#)) that gets the collection of list templates for the subsite.
6. The server (2) returns the collection of list templates for the new subsite.
7. The client application calls the method in the Lists Web Service Protocol (as described in [\[MS-LISTSWS\]](#)) that adds a list (1).
8. The server (2) returns the properties and fields of the new list (1).

9. The client application calls on the method in the Lists Web Service Protocol that adds items to a list (1).
10. The server (2) returns the list items that were added to the list (1).
11. The client application calls the method in the Lists Web Service Protocol that gets list items, and passes a custom query (as described in [\[MS-WSSCAML\]](#) section 2.2).
12. The server (2) returns the requested list items.

The following figure shows the sequence of steps for this example.



**Figure 27: Sequence for creating a new site and accessing list (1) data**

### 3.2 Example 2: Create a File from a Client Computer

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 in a SharePoint Library from a Client Computer" use case that is described in section [2.5.1.5](#).

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

This example assumes the following:

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

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

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

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

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

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

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

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

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

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

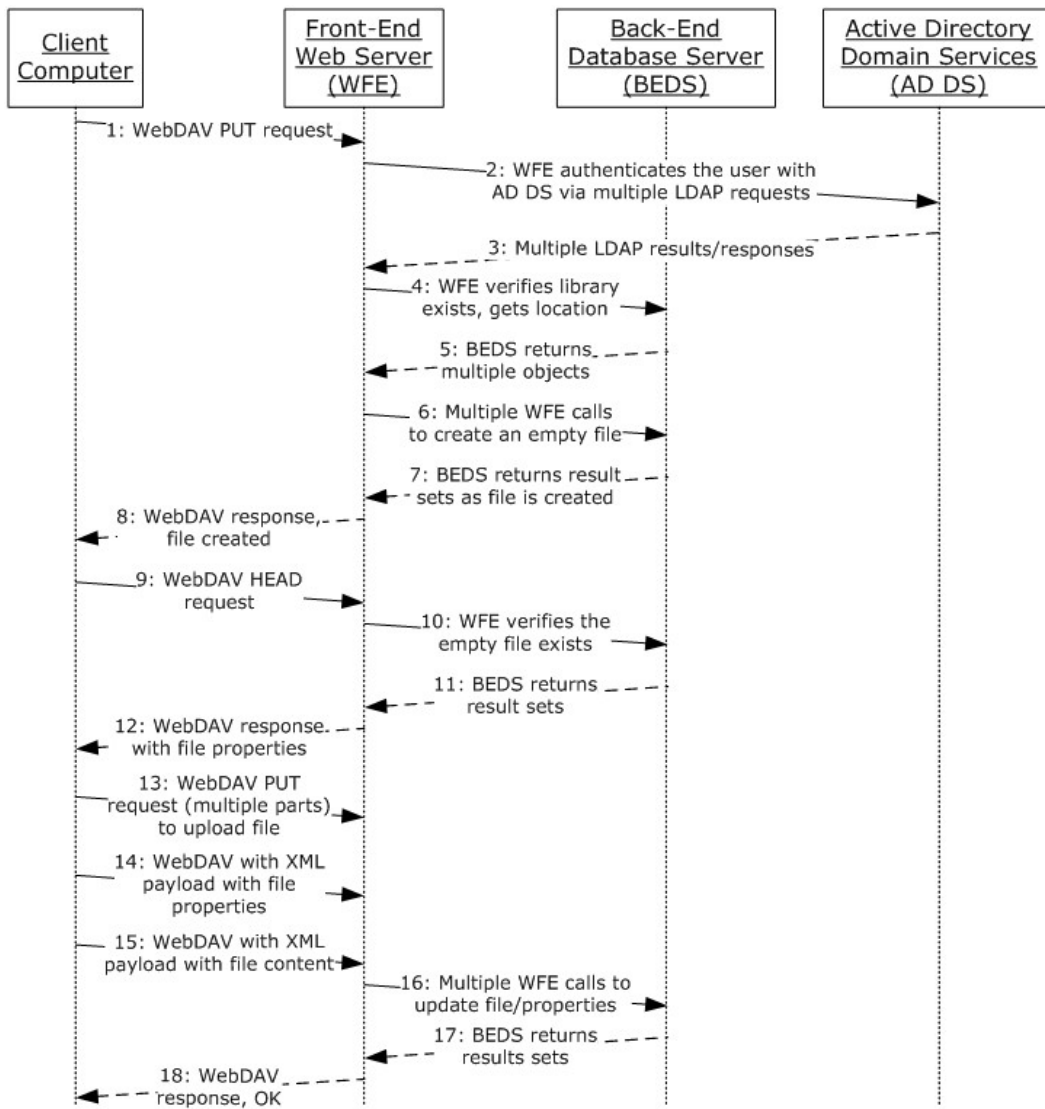
6. In multiple round trips with the back-end database server, the front-end Web server creates an empty file in the document library, and then, if successful, also verifies that the user has permissions to access and write to the document library.
7. The back-end database server returns multiple result sets as part of the process to create the file.

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

8. The front-end Web server returns a WebDAV response, saying the file was created successfully.
9. The client sends a WebDAV HEAD request to the front-end Web server, with the URL to the hello.txt file in the document library, to verify the success of the previous call.
10. In multiple round trips with the back-end database server, the front-end Web server retrieves the file.
11. The back-end database server returns multiple result sets as part of the process to retrieve the file.  
  
For more information about file retrieval see [\[MS-WSSFO2\]](#) section 4.1 for SharePoint Server 2010.
12. In response to the HEAD request, the front-end Web server sends a response saying the request was successful.
13. Then the client sends a WebDAV PUT request to the front-end Web server, which will have multiple parts to upload the file and update its properties.
14. The client sends a WebDAV request to the front-end Web server with an XML payload that has the file properties from the client.
15. The client sends a WebDAV request to the front-end Web server with an XML payload that has the file content; in this example, that content is simply the word "hello".
16. In multiple round trips with the back-end database server, the front-end Web server updates the file and its properties in the document library.
17. The back-end database server returns multiple result sets as part of the process to update the files.
18. On completing the update, the front-end Web server sends a WebDAV response, saying the request was successful.

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

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



**Figure 28: Sequence for creating a file from the client**

## 4 Microsoft Implementations

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

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

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

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

### 4.1 Product Behavior

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

## 5 Change Tracking

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



## 6 Index

### A

- [Abstract](#) 1
- Access list data
  - [details](#) 74
- [Access Services overview](#) 27
- [Access Services protocols summary](#) 39
- Add a term
  - [overview](#) 70
- Add a view
  - [overview](#) 48
- Add a Web Part to a page and update its properties
  - [overview](#) 53
- [Additional considerations](#) 73
- [Analytics Services overview](#) 30
- [Analytics Services protocols summary](#) 44
- [Applicable protocols](#) 31
- [Applicable protocols - core platform](#) 31
  - [authoring integration](#) 35
  - [client-side object model](#) 38
  - [extensibility](#) 37
  - [file operations](#) 34
  - [list data](#) 32
  - [security and identity](#) 35
  - [site administration](#) 31
- [Applicable protocols - extended capabilities](#) 38
  - [Access Services](#) 39
  - [Analytics Services](#) 44
  - [Business Connectivity Services](#) 42
  - [Content Management Service](#) 42
  - [Excel Services](#) 38
  - [InfoPath Forms Services](#) 40
  - [Office Online Web Query Service](#) 44
  - [PerformancePoint Services](#) 43
  - [search services](#) 41
  - [User Profile Service](#) 41
- [Architecture](#) 16
- [Assumptions](#) 45
- [Authentication overview](#) 24
- [Authoring integration overview](#) 25
- [Authoring integration protocols summary](#) 35
- [Authorization overview](#) 25

### B

- [Back-end protocols overview](#) 22
- [Business Connectivity Services overview](#) 29
- [Business Connectivity Services protocols summary](#) 42

### C

- [Capability negotiation](#) 73
- [Change tracking](#) 80
- [Client-side object model overview](#) 25
- [Client-side object model protocols summary](#) 38
- [Coherency requirements](#) 73
- [Communications](#) 44
  - [with other systems](#) 44

- [within the system](#) 44
- [Component dependencies](#) 44
- [Concepts](#) 16
- Considerations
  - [additional](#) 73
  - [security](#) 73
- [Content Management Service overview](#) 29
- [Content Management Service protocols summary](#) 42
- Core platform protocols
  - [authoring integration](#) 25
  - [client-side object model](#) 25
  - [extensibility](#) 25
  - [file operations](#) 24
  - [list data](#) 24
  - [overview](#) 23
  - [security and identity](#) 24
  - [site administration](#) 23
- Create a file from a client computer
  - [details](#) 75
- Create a file from the client
  - [details](#) 75
- Create a file in a SharePoint library from a client computer
  - [overview](#) 50
- Create a new site and access list data
  - [details](#) 74
- Create a site
  - [overview](#) 48
- Create a site collection
  - [overview](#) 47
- Create an external content type
  - [overview](#) 65

### D

- [Data storage overview](#) 21
- Dependencies
  - [with other systems](#) 44
  - [within the system](#) 44
- Deploy a Business Connectivity Services solution
  - [overview](#) 68
- Deployment topology
  - [back-end protocols](#) 22
  - [data storage](#) 21
  - [front-end protocols](#) 22
  - [overview](#) 16
  - [SharePoint farm](#) 17
  - [Web infrastructure](#) 18
- Design intent
  - [add a term](#) 70
  - [add a view](#) 48
  - [add a Web Part to a page and update its properties](#) 53
  - [create a file in a SharePoint library from a client computer](#) 50
  - [create a list and add a view](#) 48
  - [create an external content type](#) 65

[deploy a Business Connectivity Services solution](#) 68  
[enable browser rendering](#) 60  
[export and import site content](#) 52  
[get list data](#) 49  
[get term set data and add a term](#) 70  
[import site content](#) 52  
[interact with a database application that has been published to SharePoint Products and Technologies](#) 59  
[interact with a workbook by using a browser](#) 55  
[interact with a workbook by using an application](#) 56  
[interact with published dashboards on a Web Part page](#) 71  
[issue a search query](#) 64  
[log an event and retrieve data](#) 72  
[overview](#) 46  
[publish a database application to SharePoint Products and Technologies](#) 58  
[publish a form template and enable browser rendering](#) 60  
[publish a workbook to a SharePoint library](#) 54  
[read an external content type](#) 66  
[retrieve a user's profile](#) 62  
[tag a document](#) 63  
[update a Business Connectivity Services solution](#) 69  
[update an external content type](#) 67  
[update Web Part properties](#) 53

## E

Enable browser rendering  
[overview](#) 60  
[Environment](#) 44  
[Error handling](#) 73  
Examples  
[access list data](#) 74  
[create a file from a client computer](#) 75  
[create a file from the client](#) 75  
[create a new site and access list data](#) 74  
[overview](#) 74  
[Excel Services overview](#) 26  
[Excel Services protocols summary](#) 38  
Export and import site content  
[overview](#) 52  
Extended capabilities protocols  
[Access Services](#) 27  
[Analytics Services](#) 30  
[Business Connectivity Services](#) 29  
[Content Management Service](#) 29  
[Excel Services](#) 26  
[InfoPath Forms Services](#) 27  
[Office Online Web Query Service](#) 31  
[overview](#) 26  
[PerformancePoint Services](#) 30  
[search services](#) 28  
[User Profile Service](#) 28  
Extensibility  
[Microsoft implementations](#) 79  
[overview](#) 73

[Extensibility overview](#) 25  
[Extensibility protocols summary](#) 37  
[External dependencies](#) 44

## F

Farm  
[SharePoint overview](#) 17  
[File operations overview](#) 24  
[File operations protocols summary](#) 34  
[Front-end protocols overview](#) 22  
[Functional architecture](#) 16  
[Functional requirements - overview](#) 16

## G

Get list data  
[overview](#) 49  
Get term set data and add a term  
[overview](#) 70  
[Glossary](#) 8

## H

[Handling requirements](#) 73

## I

[Implementations - Microsoft](#) 79  
[Implementer - security considerations](#) 73  
Import site content  
[overview](#) 52  
[InfoPath Forms Services overview](#) 27  
[InfoPath Forms Services protocols summary](#) 40  
[Informative references](#) 10  
[Initial state](#) 45  
Interact with a database application that has been published to SharePoint Products and Technologies  
[overview](#) 59  
Interact with a workbook by using a browser  
[overview](#) 55  
Interact with a workbook by using an application  
[overview](#) 56  
Interact with published dashboards on a Web Part page  
[overview](#) 71  
[Introduction](#) 7  
Issue a search query  
[overview](#) 64

## L

[List data overview](#) 24  
[List data protocols summary](#) 32  
Log an event and retrieve data  
[overview](#) 72  
[Logical storage overview](#) 21

## M

[Microsoft implementations](#) 79

## O

[Office Online Web Query Service overview](#) 31  
[Office Online Web Query Service protocols summary](#) 44  
Overview  
[abstract](#) 1  
[Access Services](#) 27  
[Analytics Services](#) 30  
[authentication](#) 24  
[authoring integration](#) 25  
[authorization](#) 25  
[back-end protocols](#) 22  
[Business Connectivity Services](#) 29  
[client-side object model](#) 25  
[Content Management Service](#) 29  
core platform protocols ([section 2.1.2](#) 23, [section 2.2.1](#) 31)  
[core platform protocols – authoring integration](#) 35  
[core platform protocols – client-side object model](#) 38  
[core platform protocols – extensibility](#) 37  
[core platform protocols – file operations](#) 34  
[core platform protocols – list data](#) 32  
[core platform protocols – security and identity](#) 35  
[core platform protocols – site administration](#) 31  
[data storage](#) 21  
[deployment topology](#) 16  
[examples](#) 74  
[Excel Services](#) 26  
extended capabilities protocols ([section 2.1.3](#) 26, [section 2.2.2](#) 38)  
[extended capabilities protocols – Access Services](#) 39  
[extended capabilities protocols – Analytics Services](#) 44  
[extended capabilities protocols – Business Connectivity Services](#) 42  
[extended capabilities protocols – Content Management Service](#) 42  
[extended capabilities protocols – Excel Services](#) 38  
[extended capabilities protocols – InfoPath Forms Services](#) 40  
[extended capabilities protocols – Office Online Web Query Service](#) 44  
[extended capabilities protocols – PerformancePoint Services](#) 43  
[extended capabilities protocols – search services](#) 41  
[extended capabilities protocols – User Profile Service](#) 41  
[extensibility](#) 25  
[file operations](#) 24  
[front-end protocols](#) 22  
[InfoPath Form Services](#) 27  
[list data](#) 24  
[logical storage](#) 21  
[Office Online Web Query Service](#) 31  
[PerformancePoint Services](#) 30

[physical storage](#) 21  
[search services](#) 28  
[security and identity](#) 24  
[service applications](#) 19  
[SharePoint farm](#) 17  
[site administration](#) 23  
[summary of protocols](#) 31  
[synopsis](#) 16  
[User Profile Service](#) 28  
[Web components](#) 20  
[Web infrastructure](#) 18  
[Web Parts](#) 20  
[Web topology](#) 18

## P

[PerformancePoint Services overview](#) 30  
[PerformancePoint Services protocols summary](#) 43  
[Physical storage overview](#) 21  
[Preconditions](#) 45  
Publish a database application to SharePoint Products and Technologies  
[overview](#) 58  
Publish a form template and enable browser rendering  
[overview](#) 60  
Publish a workbook to a SharePoint library  
[overview](#) 54

## R

Read an external content type  
[overview](#) 66  
[References](#) 10  
Requirements  
[coherency](#) 73  
[error handling](#) 73  
[overview](#) 16  
[preconditions](#) 45  
Retrieve a user's profile  
[overview](#) 62

## S

[Search services overview](#) 28  
[Search services protocols summary](#) 41  
[Security and identity overview](#) 24  
[Security and identity protocols summary](#) 35  
[Security considerations](#) 73  
[Service applications overview](#) 19  
[SharePoint client-side object model protocols summary](#) 38  
[SharePoint farm overview](#) 17  
[Site administration overview](#) 23  
[Site administration protocols summary](#) 31  
[System architecture](#) 16  
[System dependencies](#) 44  
[with other systems](#) 44  
[within the system](#) 44  
[System errors](#) 73  
System overview  
[abstract](#) 1

- [System protocols](#) 31
- [System protocols - core platform](#) 31
  - [authoring integration](#) 35
  - [client-side object model](#) 38
  - [extensibility](#) 37
  - [file operations](#) 34
  - [list data](#) 32
  - [security and identity](#) 35
  - [site administration](#) 31
- [System protocols - extended capabilities](#) 38
  - [Access Services](#) 39
  - [Analytics Services](#) 44
  - [Business Connectivity Services](#) 42
  - [Content Management Service](#) 42
  - [Excel Services](#) 38
  - [InfoPath Forms Services](#) 40
  - [Office Online Web Query Service](#) 44
  - [PerformancePoint Services](#) 43
  - [search services](#) 41
  - [User Profile Service](#) 41
- [System requirements - overview](#) 16
- [System use cases](#)
  - [add a term](#) 70
  - [add a view](#) 48
  - [add a Web Part to a page and update its properties](#) 53
  - [create a file in a SharePoint library from a client computer](#) 50
  - [create a list and add a view](#) 48
  - [create a site](#) 48
  - [create a site collection](#) 47
  - [create an external content type](#) 65
  - [deploy a Business Connectivity Services solution](#) 68
  - [enable browser rendering](#) 60
  - [export and import site content](#) 52
  - [get list data](#) 49
  - [get term set data and add a term](#) 70
  - [import site content](#) 52
  - [interact with a database application that has been published to SharePoint Products and Technologies](#) 59
  - [interact with a workbook by using a browser](#) 55
  - [interact with a workbook by using an application](#) 56
  - [interact with published dashboards on a Web Part page](#) 71
  - [issue a search query](#) 64
  - [log an event and retrieve data](#) 72
  - [overview](#) 46
  - [publish a database application to SharePoint Products and Technologies](#) 58
  - [publish a form template and enable browser rendering](#) 60
  - [publish a workbook to a SharePoint library](#) 54
  - [read an external content type](#) 66
  - [retrieve a user's profile](#) 62
  - [tag a document](#) 63
  - [update a Business Connectivity Services solution](#) 69
  - [update an external content type](#) 67

- [update Web Part properties](#) 53

## T

- [Table of protocols](#) 31
- [Table of protocols - core platform](#) 31
  - [authoring integration](#) 35
  - [client-side object model](#) 38
  - [extensibility](#) 37
  - [file operations](#) 34
  - [list data](#) 32
  - [security and identity](#) 35
  - [site administration](#) 31
- [Table of protocols - extended capabilities](#) 38
  - [Access Services](#) 39
  - [Analytics Services](#) 44
  - [Business Connectivity Services](#) 42
  - [Content Management Service](#) 42
  - [Excel Services](#) 38
  - [InfoPath Forms Services](#) 40
  - [Office Online Web Query Service](#) 44
  - [PerformancePoint Services](#) 43
  - [search services](#) 41
  - [User Profile Service](#) 41
- [Tag a document](#)
  - [overview](#) 63
- [Tracking changes](#) 80

## U

- [Update a Business Connectivity Services solution](#)
  - [overview](#) 69
- [Update an external content type](#)
  - [overview](#) 67
- [Update Web Part properties](#)
  - [overview](#) 53
- [Use cases](#) 46
  - [add a term](#) 70
  - [add a view](#) 48
  - [add a Web Part to a page and update its properties](#) 53
  - [create a file in a SharePoint library from a client computer](#) 50
  - [create a list and add a view](#) 48
  - [create a site](#) 48
  - [create a site collection](#) 47
  - [create an external content type](#) 65
  - [deploy a Business Connectivity Services solution](#) 68
  - [enable browser rendering](#) 60
  - [export and import site content](#) 52
  - [get list data](#) 49
  - [get term set data and add a term](#) 70
  - [import site content](#) 52
  - [interact with a database application that has been published to SharePoint Products and Technologies](#) 59
  - [interact with a workbook by using a browser](#) 55
  - [interact with a workbook by using an application](#) 56
  - [interact with published dashboards on a Web Part page](#) 71

[issue a search query](#) 64  
[log an event and retrieve data](#) 72  
[publish a database application to SharePoint Products and Technologies](#) 58  
[publish a form template and enable browser rendering](#) 60  
[publish a workbook to a SharePoint library](#) 54  
[read an external content type](#) 66  
[retrieve a user's profile](#) 62  
[tag a document](#) 63  
[update a Business Connectivity Services solution](#) 69  
[update an external content type](#) 67  
[update Web Part properties](#) 53  
[User Profile Service overview](#) 28  
[User Profile Service protocols summary](#) 41

## **V**

Versioning

[Microsoft implementations](#) 79  
[overview](#) 73

## **W**

[Web components overview](#) 20  
[Web infrastructure overview](#) 18  
[Web Parts overview](#) 20  
[Web topology overview](#) 18