

[MS-DWSS]:

Document Workspace Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
6/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
7/13/2009	1.02	Major	Revised and edited the technical content
8/28/2009	1.03	Editorial	Revised and edited the technical content
11/6/2009	1.04	Editorial	Revised and edited the technical content
2/19/2010	2.0	Minor	Updated the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.0	Major	Significantly changed the technical content.
4/11/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.1	Minor	Clarified the meaning of the technical content.
9/12/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	3.2	Minor	Clarified the meaning of the technical content.
2/11/2013	4.0	Major	Significantly changed the technical content.
7/30/2013	4.1	Minor	Clarified the meaning of the technical content.
11/18/2013	4.1	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	5.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
4/30/2014	5.1	Minor	Clarified the meaning of the technical content.
7/31/2014	5.2	Minor	Clarified the meaning of the technical content.
10/30/2014	5.3	Minor	Clarified the meaning of the technical content.
3/16/2015	6.0	Major	Significantly changed the technical content.
6/30/2015	7.0	Major	Significantly changed the technical content.
2/26/2016	8.0	Major	Significantly changed the technical content.
4/14/2016	9.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	7
1.1	Glossary	7
1.2	References	9
1.2.1	Normative References	9
1.2.2	Informative References	10
1.3	Overview	10
1.4	Relationship to Other Protocols	12
1.5	Prerequisites/Preconditions	12
1.6	Applicability Statement	13
1.7	Versioning and Capability Negotiation	13
1.8	Vendor-Extensible Fields	13
1.9	Standards Assignments.....	13
2	Messages.....	14
2.1	Transport	14
2.2	Common Message Syntax	14
2.2.1	Namespaces	14
2.2.2	Messages.....	14
2.2.3	Elements	15
2.2.3.1	Assignees	15
2.2.3.2	Error	15
2.2.3.3	ID	17
2.2.3.4	List	17
2.2.3.5	Member	18
2.2.3.6	Roles.....	19
2.2.4	Complex Types.....	19
2.2.4.1	MemberData	20
2.2.4.2	Schema	20
2.2.5	Simple Types	21
2.2.5.1	ListType.....	21
2.2.5.2	RoleType.....	21
2.2.6	Attributes	22
2.2.7	Groups	22
2.2.8	Attribute Groups.....	22
3	Protocol Details.....	23
3.1	Server Details.....	23
3.1.1	Abstract Data Model.....	23
3.1.2	Timers	23
3.1.3	Initialization.....	23
3.1.4	Message Processing Events and Sequencing Rules	23
3.1.4.1	CanCreateDwsUrl	24
3.1.4.1.1	Messages	24
3.1.4.1.1.1	CanCreateDwsUrlSoapIn	24
3.1.4.1.1.2	CanCreateDwsUrlSoapOut.....	25
3.1.4.1.2	Elements	25
3.1.4.1.2.1	CanCreateDwsUrl	25
3.1.4.1.2.2	CanCreateDwsUrlResponse	25
3.1.4.2	CreateDws	26
3.1.4.2.1	Messages	26
3.1.4.2.1.1	CreateDwsSoapIn	27
3.1.4.2.1.2	CreateDwsSoapOut	27
3.1.4.2.2	Elements.....	27
3.1.4.2.2.1	CreateDws	27
3.1.4.2.2.2	CreateDwsResponse.....	28

3.1.4.3	CreateFolder	30
3.1.4.3.1	Messages	30
3.1.4.3.1.1	CreateFolderSoapIn	30
3.1.4.3.1.2	CreateFolderSoapOut	31
3.1.4.3.2	Elements	31
3.1.4.3.2.1	CreateFolder	31
3.1.4.3.2.2	CreateFolderResponse	31
3.1.4.4	DeleteDws	32
3.1.4.4.1	Messages	32
3.1.4.4.1.1	DeleteDwsSoapIn	32
3.1.4.4.1.2	DeleteDwsSoapOut	33
3.1.4.4.2	Elements	33
3.1.4.4.2.1	DeleteDws	33
3.1.4.4.2.2	DeleteDwsResponse	33
3.1.4.5	DeleteFolder	34
3.1.4.5.1	Messages	34
3.1.4.5.1.1	DeleteFolderSoapIn	34
3.1.4.5.1.2	DeleteFolderSoapOut	34
3.1.4.5.2	Elements	35
3.1.4.5.2.1	DeleteFolder	35
3.1.4.5.2.2	DeleteFolderResponse	35
3.1.4.6	FindDwsDoc	36
3.1.4.6.1	Messages	36
3.1.4.6.1.1	FindDwsDocSoapIn	36
3.1.4.6.1.2	FindDwsDocSoapOut	36
3.1.4.6.2	Elements	37
3.1.4.6.2.1	FindDwsDoc	37
3.1.4.6.2.2	FindDwsDocResponse	37
3.1.4.7	GetDwsData	38
3.1.4.7.1	Messages	38
3.1.4.7.1.1	GetDwsDataSoapIn	38
3.1.4.7.1.2	GetDwsDataSoapOut	38
3.1.4.7.2	Elements	39
3.1.4.7.2.1	GetDwsData	39
3.1.4.7.2.2	GetDwsDataResponse	39
3.1.4.8	GetDwsMetaData	41
3.1.4.8.1	Messages	41
3.1.4.8.1.1	GetDwsMetaDataSoapIn	41
3.1.4.8.1.2	GetDwsMetaDataSoapOut	41
3.1.4.8.2	Elements	42
3.1.4.8.2.1	GetDwsMetaData	42
3.1.4.8.2.2	GetDwsMetaDataResponse	43
3.1.4.8.2.3	ListInfo	45
3.1.4.9	RemoveDwsUser	46
3.1.4.9.1	Messages	46
3.1.4.9.1.1	RemoveDwsUserSoapIn	47
3.1.4.9.1.2	RemoveDwsUserSoapOut	47
3.1.4.9.2	Elements	47
3.1.4.9.2.1	RemoveDwsUser	47
3.1.4.9.2.2	RemoveDwsUserResponse	48
3.1.4.10	RenameDws	48
3.1.4.10.1	Messages	48
3.1.4.10.1.1	RenameDwsSoapIn	49
3.1.4.10.1.2	RenameDwsSoapOut	49
3.1.4.10.2	Elements	49
3.1.4.10.2.1	RenameDws	49
3.1.4.10.2.2	RenameDwsResponse	50
3.1.4.11	UpdateDwsData	50

3.1.4.11.1	Messages	50
3.1.4.11.1.1	UpdateDwsDataSoapIn	51
3.1.4.11.1.2	UpdateDwsDataSoapOut	51
3.1.4.11.2	Elements	51
3.1.4.11.2.1	UpdateDwsData	51
3.1.4.11.2.2	UpdateDwsDataResponse	52
3.1.5	Timer Events	52
3.1.6	Other Local Events	52
4	Protocol Examples	53
4.1	CanCreateDwsUrl	53
4.1.1	Request	53
4.1.2	Response	53
4.2	CreateDws	53
4.2.1	Request	54
4.2.2	Response	54
4.3	CreateFolder	55
4.3.1	Request	55
4.3.2	Response	55
4.4	DeleteDws	55
4.4.1	Request	55
4.4.2	Response	56
4.5	DeleteFolder	56
4.5.1	Request	56
4.5.2	Response	56
4.6	FindDwsDoc	57
4.6.1	Request	57
4.6.2	Response	57
4.7	RemoveDwsUser	58
4.7.1	Request	58
4.7.2	Response	58
4.8	RenameDws	58
4.8.1	Request	58
4.8.2	Response	59
4.9	GetDwsData	59
4.9.1	Request	59
4.9.2	Response	59
4.10	GetDwsMetaData	61
4.10.1	Request	61
4.10.2	Response	62
5	Security	67
5.1	Security Considerations for Implementers	67
5.2	Index of Security Parameters	67
6	Appendix A: Full WSDL	68
7	Appendix B: Product Behavior	77
8	Change Tracking	79
9	Index	81

1 Introduction

The Document Workspace Web Service Protocol is used to create, edit, and delete workspaces and folders for a SharePoint site configured as a Document Workspace. After the workspace is created, folders can be created in the workspace to organize the shared work. Workspaces are template sites intended for easier creation and maintenance than an entire shared Web site.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

absolute URL: The full Internet address of a page or other World Wide Web resource. The absolute URL includes a protocol, such as "http," a network location, and an optional path and file name — for example, <http://www.treyresearch.net/>.

Active Directory: A general-purpose network directory service. **Active Directory** also refers to the Windows implementation of a directory service. **Active Directory** stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in **Active Directory**. **Active Directory** is either deployed as **Active Directory Domain Services (AD DS)** or Active Directory Lightweight Directory Services (AD LDS). [\[MS-ADTS\]](#) describes both forms. For more information, see [\[MS-AUTHSOD\]](#) section 1.1.1.5.2, Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Kerberos, and DNS.

Active Directory Domain Services (AD DS): A directory service (DS) implemented by a domain controller (DC). The DS provides a data store for objects that is distributed across multiple DCs. The DCs interoperate as peers to ensure that a local change to an object replicates correctly across DCs. For more information, see [\[MS-AUTHSOD\]](#) section 1.1.1.5.2 and [\[MS-ADTS\]](#). For information about product versions, see [\[MS-ADTS\]](#) section 1. See also **Active Directory**.

ActiveX Data Objects (ADO): A data access interface that connects to, retrieves, manipulates, and updates data in Object Linking and Embedding (OLE) database-compliant data sources.

alert: An Internet message that is sent to subscribers automatically to notify them when user-defined criteria are met. Alerts are generated automatically when items such as documents, webpages, list items, sites, or other resources on a server are changed.

authenticated user: A built-in security group specified in [\[MS-WSO\]](#) whose members include all users that can be authenticated by a computer.

check out: The process of retrieving a writable copy of a file or project from a source repository. This locks the file for editing to prevent other users from overwriting or editing it inadvertently. See also check in.

Collaborative Application Markup Language (CAML): An XML-based language that is used to describe various elements, such as queries and views, in sites that are based on SharePoint Products and Technologies.

current user: The user who is authenticated during processing operations on a front-end web server or a back-end database server.

document library: A type of list that is a container for documents and folders.

Document Workspace site: A SharePoint site that is based on a Document Workspace site template and has a template identifier value of "1". A Document Workspace site is used for planning, posting, and working together on a document or a set of related documents.

folder: A file system construct. File systems organize a volume's data by providing a hierarchy of objects, which are referred to as folders or directories, that contain files and can also contain other folders.

friendly name: A name for a user or object that can be read and understood easily by a human.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [\[RFC4122\]](#) or [\[C706\]](#) must be used for generating the **GUID**. See also universally unique identifier (UUID).

group: A named collection of users who share similar access permissions or roles.

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

item: A unit of content that can be indexed and searched by a search application.

link: An attribute (2) value that refers to a directory object and whose Attribute-Schema object specifies an even value for the linkId attribute. Also referred to as forward link.

list: A container within a SharePoint site that stores list items. A list has a customizable schema that is composed of one or more fields.

list item: An individual entry within a SharePoint list. Each list item has a schema that maps to fields in the list that contains the item, depending on the content type of the item.

member: A user in the **Members group** of a site.

Members group: A default group of users on a SharePoint site. By default, the Members group is assigned the Contribute permission level.

page: A file that consists of HTML and can include references to graphics, scripts, or dynamic content such as Web Parts.

personal view: A view of a list that is created by a user for personal use. The view is unavailable to other users.

role: A symbolic name that defines a class of users for a set of components. A role defines which users can call interfaces on a component.

site: A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

site-relative URL: A URL that is relative to the site that contains a resource and does not begin with a leading slash (/).

SOAP action: The HTTP request header field used to indicate the intent of the SOAP request, using a **URI** value. See [\[SOAP1.1\]](#) section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a SOAP message to its recipient. See [\[SOAP1.2-1/2007\]](#) section 5.3 for more information.

SOAP fault: A container for error and status information within a SOAP message. See [\[SOAP1.2-1/2007\]](#) section 5.4 for more information.

SOAP operation: An action that can be performed by a Simple Object Access Protocol (SOAP) service, as described in [\[SOAP1.1\]](#).

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [\[RFC3986\]](#).

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [\[RFC1738\]](#).

user identifier: An integer that uniquely identifies a security principal (2) as distinct from all other security principals (2) and site groups within the same site collection.

Web Part: A reusable component that contains or generates web-based content such as **XML**, HTML, and scripting code. It has a standard property schema and displays that content in a cohesive unit on a webpage. See also Web Parts Page.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

XML: The Extensible Markup Language, as described in [\[XML1.0\]](#).

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [\[RFC3986\]](#). A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [\[XMLNS-2ED\]](#).

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring XML schemas.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

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

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

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/rfc/rfc2119.txt>

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.rfc-editor.org/rfc/rfc2616.txt>

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

1.2.2 Informative References

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

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

1.3 Overview

This protocol provides methods to create and update **Document Workspace sites**.

A Document Workspace site is a convenient and centralized place for collaboration on a project. For example, team members can create and update documents and publish them on the Document Workspace site. The most current version of documents that the team is working on always appears on the Document Workspace site, so team members do not have to send large files by e-mail when content changes.

Each method in the protocol is a **SOAP operation** that accepts a set of parameters as a SOAP request and returns a set of values as a SOAP response. The protocol client sends a request to the protocol server by a SOAP request message, and the protocol server sends return values to the protocol client by a SOAP response message.

The **CanCreateDwsUrl** operation is a simple example of a DWS client-server exchange. The protocol client initiates the exchange with a **CanCreateDwsUrlSoapIn** message. The protocol server processes the request and returns a response in a **CanCreateDwsUrlSoapOut** message, as shown in the following illustration.

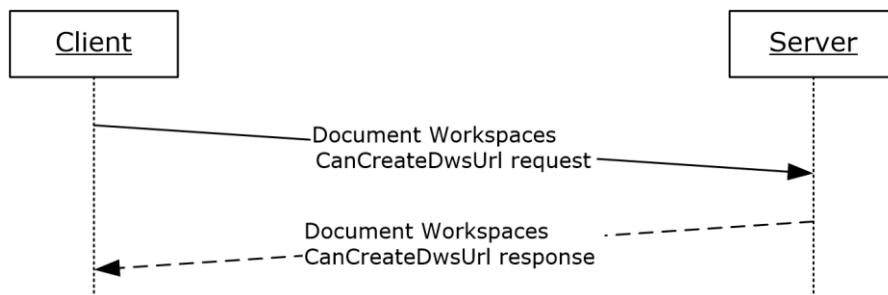


Figure 1: Communication during the CanCreateDwsUrl operation

The **CreateDws** operation creates a Document Workspace site. The protocol client can use the **CanCreateDwsUrl** operation to determine whether the **current user** has the necessary rights on the protocol server to create the workspace. The **CanCreateDwsUrl** operation also ensures that the new **Uniform Resource Locator (URL)** is unique. The protocol client initiates a **CreateDws** operation, providing a URL for the new workspace to be created. This URL becomes the base for the new workspace, that is, further operations on the new workspace are performed by sending the request to a URL in the form of "HTTP://<URL>/_vti_bin/dws.asmx". For example, if the **CreateDws** client request includes the URL "http://my_server/workspaces/user_one", further operations on the new workspace are sent to the SOAP service at "http://my_server/workspaces/user_one/_vti_bin/dws.asmx".

The following illustration shows the typical protocol client-protocol server interaction when creating a Document Workspace.

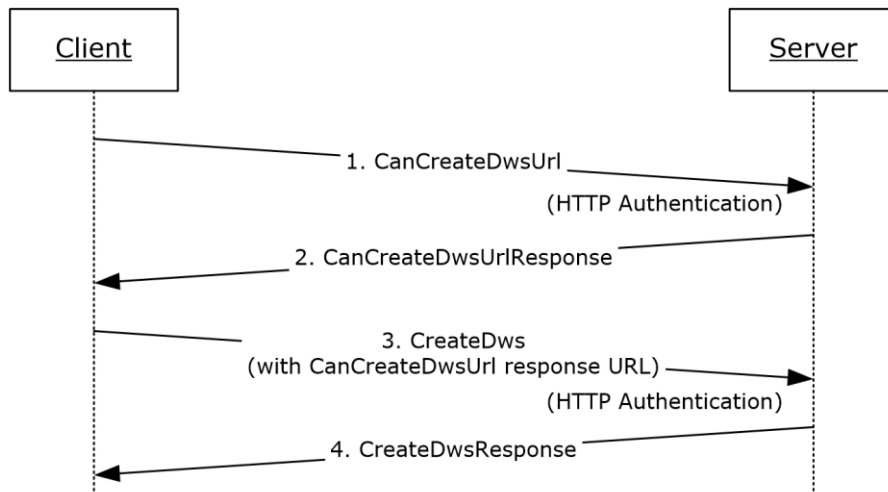


Figure 2: Creating a Document Workspace

Once a workspace has been created, other operations can be performed if they are DWS or other Web services that are based on a **site** URL. A typical operation uses **GetDwsData** to get information about the workspace. The **GetDwsData** operation can then be repeated, using the **LastUpdate** field to determine whether any changes have been made to the workspace. The following illustration shows **GetDwsData** using **LastUpdate**, resulting in a "NoChanges" response from the protocol server.

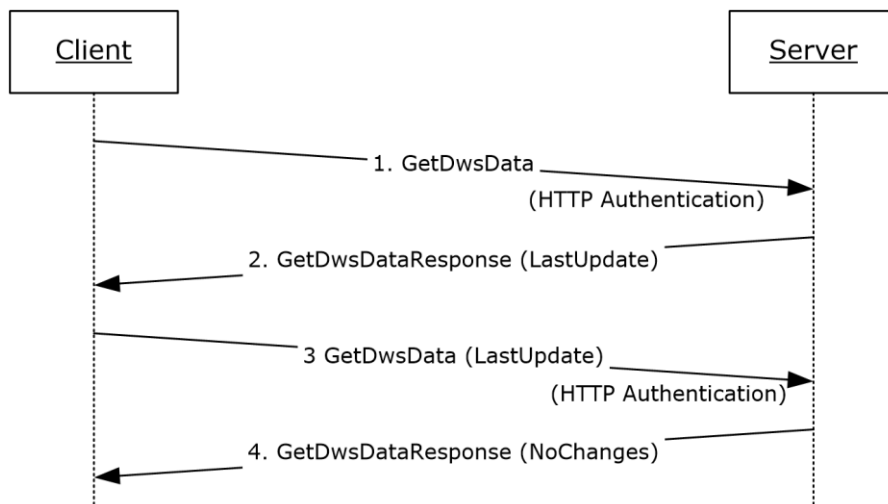


Figure 3: Client-server interaction when LastUpdate is used

1.4 Relationship to Other Protocols

This protocol uses the SOAP message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2/1] and [SOAP1.2/2]. It transmits those messages by using **HTTP**, as described in [RFC2616], or Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS), as described in [RFC2818].

The following diagram shows the underlying messaging and transport stack used by the protocol.

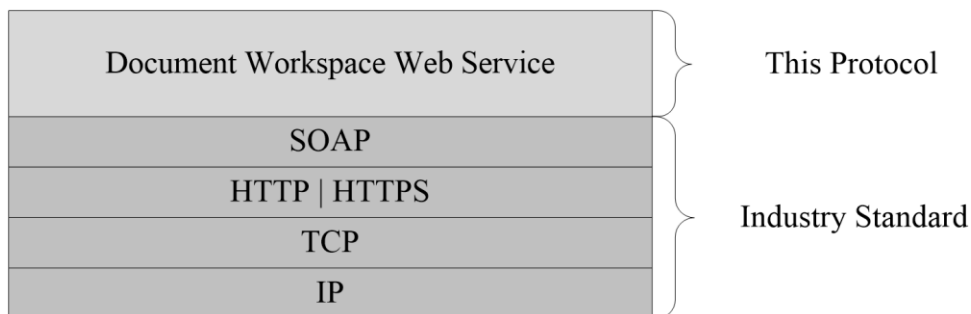


Figure 4: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a site that is identified by a URL that is known by protocol clients. The protocol server endpoint is formed by appending "_vti_bin/Dws.asmx" to the site URL—for example "www.contoso.com/Repository/_vti_bin/Dws.asmx".

This protocol assumes that authentication has been performed by the underlying protocols. The account that accesses this Web service is assigned an appropriate **role** in the Document Workspace **Members group** or for the site that contains, or will contain, the Document Workspace site.

1.6 Applicability Statement

This protocol enables protocol client applications to create and manage Document Workspaces. A Document Workspace is a designated area on a protocol server that is used by a group of users to collaborate on a document or project. The workspace provides a shared area for group documents.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following area:**Supported transports:** This protocol uses multiple transports with SOAP as specified in section 1.7.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

2.1 Transport

Protocol servers **MUST** support SOAP over HTTP. Protocol servers **SHOULD** additionally support SOAP over HTTPS for securing communication with protocol clients.

Protocol messages **MUST** be formatted as specified in [\[SOAP1.1\]](#) SOAP Envelope section 4, or in [\[SOAP1.2/1\]](#) SOAP Message Construct section 5. Protocol server faults **MUST** be returned by using HTTP status codes as specified in [\[RFC2616\]](#) Status Code Definitions section 10 or **SOAP faults** as specified in [\[SOAP1.1\]](#) SOAP Fault section 4.4 or in [\[SOAP1.2/1\]](#) SOAP Fault section 5.4.

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses XML schema, as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and WSDL, as specified in [\[WSDL\]](#).

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific namespace prefix for each namespace that is used, the choice of any particular namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[WSDL]
tns	http://schemas.microsoft.com/sharepoint/soap/dws/	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[WSDL]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xdr	uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882	

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

The following table summarizes the set of common XML schema element definitions defined by this specification. XML schema element definitions that are specific to a particular operation are described with the operation.

Element	Description
Assignees	Returns a list of assigned members for a Document Workspace.
Error	Returns an error code from DWS operations.
ID	Contains a GUID with an optional attribute.
List	Returns data about a list (1).
Member	Returns data about a user or group .
Roles	Contains role information about a list or document.

2.2.3.1 Assignees

This element holds the list of assigned users for a Document Workspace. It is used only in the stand-alone **XML** returned from the **GetDwsData** and the **GetDwsMetaData** operations. This element is defined as follows:

```
<xs:element name="Assignees">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:Member" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

2.2.3.2 Error

This element returns an error indication. It contains a restricted string indication, as well as an integer code that corresponds to the string. Optionally, it contains a URL that the user on the protocol client accesses to authenticate and gain the necessary permissions. This element is defined as follows:

```
<xs:simpleType name="ErrorTypes">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ServerFailure"/>
    <xs:enumeration value="Failed"/>
    <xs:enumeration value="NoAccess"/>
    <xs:enumeration value="Conflict"/>
    <xs:enumeration value="ItemNotFound"/>
    <xs:enumeration value="MemberNotFound"/>
    <xs:enumeration value="ListNotFound"/>
    <xs:enumeration value="TooManyItems"/>
    <xs:enumeration value="DocumentNotFound"/>
    <xs:enumeration value="FolderNotFound"/>
    <xs:enumeration value="WebContainsSubwebs"/>
    <xs:enumeration value="ADMode"/>
    <xs:enumeration value="AlreadyExists"/>
    <xs:enumeration value="QuotaExceeded"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="Error">
```

```

<xs:complexType>
  <xs:simpleContent>
    <xs:extension base="tns:ErrorTypes">
      <xs:attribute name="ID">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:minInclusive value="1"/>
            <xs:maxInclusive value="14"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="AccessUrl" type="xs:string"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:element>

```

AccessUrl: This is a URL that the protocol client SHOULD use to gain the necessary access permissions. This attribute can be present when the **Error** field is set to "NoAccess". This attribute MUST NOT be present in any other case.

Error: This is a string with the error type.

ID: Error type index, converted to a string. The value MUST match the error type as listed in the following table. The returned error codes are also in the following table. See the individual operation specifications for the situations in which these codes are returned. The error types **ADMode** and **MemberNotFound** are reserved and not sent.

Error code	ID
ServerFailure	1
Failed	2
NoAccess	3
Conflict	4
ItemNotFound	5
MemberNotFound	6
ListNotFound	7
TooManyItems	8
DocumentNotFound	9
FolderNotFound	10
WebContainsSubwebs	11
ADMode	12
AlreadyExists	13
QuotaExceeded	14

2.2.3.3 ID

The **ID** element is used in the **List** structure to return a GUID that identifies a document for which the protocol server returns error information. The **DefaultUrl** attribute specifies a **page** on which the user can view the contents of the list. This element is defined as follows:

```
<xs:element name="ID">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="DefaultUrl" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

2.2.3.4 List

This element returns the information about the contents of a list. It is defined as follows:

```
<xs:element name="List">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="tns:Error"/>
      <xs:choice>
        <xs:element name="NoChanges" type="xs:string"/>
        <xs:sequence>
          <xs:sequence>
            <xs:element ref="tns:ID"/>
            <xs:choice>
              <xs:element ref="tns:Error" minOccurs="0"/>
              <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##other"
processContents='skip' />
              </xs:sequence>
            </xs:choice>
          </xs:sequence>
        </xs:choice>
      </xs:choice>
      <xs:attribute name="Name" type="tns:ListType" use="required"/>
    </xs:complexType>
  </xs:element>
```

NoChanges: This element **MUST** be returned when the protocol client has provided a *LastUpdate* parameter and the specified document has not changed since the value in *LastUpdate*. This element **MUST** be empty.

ID: Contains a GUID that corresponds to a document on the protocol server.

Name: Describes the list contents. The list is identified by the **Name** attribute in the **List** element. The following table shows restrictions on the rest of the structure based on the **Name** attribute.

Name	Description
Tasks	The list of tasks associated with this workspace. The list contains a sequence of XML elements with information about the tasks.
Documents	The list of documents associated with this workspace. The list contains a sequence of z:row elements with information about the documents.

Name	Description
Links	The list of external links associated with this workspace. It contains a sequence of XML elements with information about the links contained in the list.

Error: Specifies that an error occurred while accessing the list data. The error codes are described in the following table.

Error code	Description
ServerFailure	The protocol server encountered an unspecified error.
NoAccess	The authenticated user does not have sufficient permissions to access list information for the Document Workspace.
ListNotFound	The specified list does not exist in the Document Workspace.
TooManyItems	The number of lists exceeds an implementation-specific size<1>.

The **List** element can contain an array of **z:row** elements in **any** of its elements. In a **z:row** element, the **z** is equal to #RowsetSchema in the **ActiveX Data Objects (ADO)** XML Persistence format (see [\[MS-PRSTFR\]](#)). Each **z:row** element describes a single list item. The names of the attributes of this element correspond to the names of fields in the list. (For more details, see [\[MS-WSSFO2\]](#).)

2.2.3.5 Member

This element contains information about a member of the workspace. This element is defined as follows:

```
<s:element name="Member">
  <s:complexType>
    <s:all>
      <s:element name="ID" type="s:integer"/>
      <s:element name="Name" type="s:string"/>
      <s:element name="LoginName" type="s:string"/>
      <s:element name="Email" type="s:string" minOccurs="0"/>
      <s:element name="IsDomainGroup" minOccurs="0">
        <s:simpleType>
          <s:restriction base="s:string">
            <s:enumeration value="True" />
            <s:enumeration value="False" />
          </s:restriction>
        </s:simpleType>
      </s:element>
    </s:all>
  </s:complexType>
</s:element>
```

ID: A positive integer that **MUST** uniquely identify a user or group.

Name: A string that contains a **friendly name** for the user or group.

LoginName: Represents the name used to log on to the protocol server. If **IsDomainGroup** is set to **True**, this field **MUST** be empty.

E-mail: A string containing the e-mail address of the user. If **IsDomainGroup** is set to **True**, this field **MUST** be empty.

IsDomainGroup: A Boolean value specifying whether this record is associated with a user or group. When its value is **True**, it is a group. When its value is **False**, it is a user.

2.2.3.6 Roles

The **Roles** element contains information about how to grant access permissions to users. This element is defined as follows:

```
<xs:element name="Roles">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element ref="tns:Error"/>
        <xs:sequence>
          <xs:element name="Role" maxOccurs="unbounded">
            <xs:complexType>
              <xs:attribute name="Name" type="xs:string"
                use="required"/>
              <xs:attribute name="Type" type="tns:RoleType"
                use="required"/>
              <xs:attribute name="Description" type="xs:string"
                use="required"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Error: Specifies that an error has occurred while accessing the **Roles** data. The error codes are described in the following table.

Error code	Description
NoAccess	Authenticated user does not have sufficient permissions to access role information for the Document Workspace.
ServerFailure	Protocol server encountered an unspecified error.

Role: Returns the information about a single role.

Name: Returns the friendly name of a role.

Type: Returns the type of role.

Description: Returns a general description of the role.

2.2.4 Complex Types

The following table summarizes the set of common XML schema complex type definitions defined by this specification. XML schema complex type definitions that are specific to a particular operation are described with the operation.

Complex type	Description
MemberData	Information about a user or group or an error condition that occurred while accessing the user or group data.

2.2.4.1 MemberData

This type returns information about a user or group or an error condition that occurred while accessing the user or group data. Which value is chosen depends on the error that occurred. **Error** MUST be used when there is an access issue or an error other than **TooManyItems**. A sequence containing two **URL** elements and an **Error** element MUST be used for a **TooManyItems** error. **ServerFailure** MUST be used for unspecified protocol server errors. This type is defined as follows:

```
<xs:complexType name="MemberData">
  <xs:choice>
    <xs:element ref="tns:Error"/>
    <xs:sequence>
      <xs:element name="DefaultUrl" type="xs:string"/>
      <xs:element name="AlternateUrl" type="xs:string"/>
      <xs:element ref="tns:Error"/>
    </xs:sequence>
    <xs:sequence>
      <xs:element ref="tns:Member" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:choice>
</xs:complexType>
```

Member: An element describing a single user or group.

DefaultUrl: A URL to a page with an editable listing of the members of the Document Workspace. This URL MUST be returned when the **TooManyItems** error code is returned.

AlternateUrl: A URL to a page with an editable listing of the members of the Document Workspace. The protocol client can use this URL instead of the **DefaultUrl**. This URL MUST be returned when the **TooManyItems** error code is returned.

Error: An **Error** element containing an error that occurred during access. Possible error codes are listed in the following table.

Error code	Description
NoAccess	The current user does not have access to this information.
TooManyItems	The number of members exceeds an implementation-specific size. <2>
ServerFailure	The protocol server encountered an unspecified error.

2.2.4.2 Schema

The **Schema** element is specified in [\[MS-PRSTFR\]](#) section 2.3.1.2. The following XML schema defines the **Schema** element:

```
<xs:element name='Schema'>
  <xs:complexType >
    <xs:choice minOccurs='0' maxOccurs='unbounded'>
      <xs:element ref='xdr:AttributeType'/>
      <xs:element ref='xdr:ElementType'/>
      <xs:element ref='xdr:description'/>
      <xs:any namespace='##other' processContents='skip'/>
    </xs:choice>
    <xs:attribute name='Name' type='xs:string'/>
    <xs:attribute name='id' type='xs:ID'/>
    <xs:anyAttribute namespace='##other'
      processContents='skip'/>
  </xs:complexType >
</xs:element >
```

```
</xs:complexType>
</xs:element>
```

AttributeType: Specified in [MS-PRSTFR] section 2.3.1.2.

ElementType: Specified in [MS-PRSTFR] section 2.3.1.2.

description: A description for the Schema element.

Name: The name of the schema.

id: The identifier of the schema. Specified in [MS-PRSTFR] section 2.3.1.2.

2.2.5 Simple Types

The following table summarizes the set of common XML schema simple type definitions defined by this specification. XML schema simple type definitions that are specific to a particular operation are described with the operation.

Simple type	Description
ListType	A string that indicates the type of list being returned.
RoleType	A string that indicates the user role.

2.2.5.1 ListType

This simple type lists the values for the **Name** attribute of the **List** element as specified in section [2.2.3.4](#). This type is defined as follows:

```
<xs:simpleType name="ListType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Tasks"/>
    <xs:enumeration value="Documents"/>
    <xs:enumeration value="Links"/>
  </xs:restriction>
</xs:simpleType>
```

2.2.5.2 RoleType

This string enumeration specifies the values for user roles in the **GetDwsMetaData** structure. This type is defined as follows:

```
<xs:simpleType name="RoleType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="Reader"/>
    <xs:enumeration value="Contributor"/>
    <xs:enumeration value="WebDesigner"/>
    <xs:enumeration value="Administrator"/>
    <xs:enumeration value="Editor"/>
  </xs:restriction>
</xs:simpleType>
```

The following table describes the rights for each role. In the following descriptions, the word "user" is used for the authenticated identity as described in section 1.5.

Role	Rights
None	This role was created by a user, and is not one of the following four roles.
Reader	Has read-only rights to view items, personalize Web Parts , use alerts , and create a top-level Web site by using Self-Service Site Creation. A reader cannot add content. When a user is granted the role of reader and creates a site by using Self-Service Site Creation, the user becomes the site owner and a member of the administrator class for the new site. This does not affect membership for any other site.
Contributor	Has Reader rights, in addition to rights to add items, edit items, delete items, manage list permissions, manage personal views , personalize Web Part pages, and browse directories. Contributors cannot create new lists or document libraries , but they can add content to existing lists and document libraries.
WebDesigner	Has Contributor rights, in addition to the rights to cancel check out , delete items, manage lists, add or customize pages, define and apply themes and borders, and link style sheets. Web Designers can modify the structure of the site and create new lists or document libraries.
Administrator	Has all rights from all other roles, in addition to the rights to manage roles and view usage-analysis data. This role cannot be customized or deleted and contains at least one member (1). Role members have or can grant themselves access to any item on the site (2).
Editor<3>	Has Contributor rights, plus the rights to manage lists. Editors can create new lists or document libraries.

2.2.6 Attributes

This specification does not define any common XML schema attribute definitions.

2.2.7 Groups

This specification does not define any common XML schema group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML schema attribute group definitions.

3 Protocol Details

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

The client side of this protocol is a pass-through—that is, no additional timers or other states are required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

3.1 Server Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

A Document Workspace is a site that is set up for collaboration by workgroups. The workspace provides a logical organization of documents, tasks, and links to external resources. The Document Workspace provides tools to assist in the collaboration, including these lists, keeping version history for documents, providing a check out service for documents, and supporting alert subscriptions on documents and lists in the workspace. The workspace can have additional organization added with workspace folders.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the list of WSDL operations as defined by this specification.

Operation	Description
CanCreateDwsUrl	Determines whether a protocol client has the permissions necessary to create a Document Workspace at the specified URL. It also returns a URL that is unique for the current site.
CreateDws	Creates a Document Workspace.
CreateFolder	Creates a folder in a Document Workspace or another folder.
DeleteDws	Deletes a Document Workspace.

Operation	Description
DeleteFolder	Deletes a folder in a Document Workspace.
FindDwsDoc	Obtains the URL of a document stored in a Document Workspace.
GetDwsData	Returns information about the specified Document Workspace.
GetDwsMetaData	Returns metadata about the specified Document Workspace.
RemoveDwsUser	Deletes the specified user from the list of authorized Document Workspace users.
RenameDws	Renames the title of a Document Workspace.
UpdateDwsData	Modifies the metadata associated with a Document Workspace.

3.1.4.1 CanCreateDwsUrl

This operation determines whether an authenticated user has permission to create a Document Workspace at the specified URL.

The protocol client sends a **CanCreateDwsUrlSoapIn** request message, and the protocol server responds with the following **CanCreateDwsUrlSoapOut** response message:

```
<wsdl:operation name="CanCreateDwsUrl">
  <wsdl:input message="tns:CanCreateDwsUrlSoapIn" />
  <wsdl:output message="tns:CanCreateDwsUrlSoapOut" />
</wsdl:operation>
```

The protocol server MUST respond with an HTTP 401 error if the user is not authorized to create the specified Document Workspace.

The server MUST respond with an **Error** element, as specified in section [2.2.3.2](#), with the identifier set to 2 and a restricted string of **"Failed"** when the length of the root site (2) URL is greater than an implementation-dependent length. [<4>](#)

3.1.4.1.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
CanCreateDwsUrlSoapIn	Requests the CanCreateDwsUrl WSDL operation, as specified in section 3.1.4.1 .
CanCreateDwsUrlSoapOut	Response to the CanCreateDwsUrl operation.

3.1.4.1.1.1 CanCreateDwsUrlSoapIn

This message requests the **CanCreateDwsUrl** WSDL operation.

The **SOAP action** value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/CanCreateDwsUrl
```

The **SOAP body** contains a **CanCreateDwsUrl** element.

3.1.4.1.1.2 CanCreateDwsUrlSoapOut

This message responds to a **CanCreateDwsUrl** WSDL operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/CanCreateDwsUrl
```

The SOAP body contains a **CanCreateDwsUrlResponse** element.

3.1.4.1.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
CanCreateDwsUrl	Contains the parameter for the CanCreateDwsUrl operation, as specified in section 3.1.4.1 .
CanCreateDwsUrlResponse	Contains the response to the CanCreateDwsUrl operation request.

3.1.4.1.2.1 CanCreateDwsUrl

This element contains the parameter for the **CanCreateDwsUrl** WSDL operation. This element is defined as follows:

```
<s:element name="CanCreateDwsUrl">
  <s:complexType>
    <s:sequence>
      <s:element name="url" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

url: **Site-relative URL** that specifies where to create the Document Workspace. The URL MUST NOT be empty. The URL MUST be unique for the site. The URL SHOULD be based on the URL provided in the *url* input parameter. If the *url* input parameter is empty, the protocol server SHOULD return a created name, such as a GUID.

3.1.4.1.2.2 CanCreateDwsUrlResponse

This structure provides the reply value for the **CanCreateDwsUrl** operation. This structure is defined as follows:

```
<s:element name="CanCreateDwsUrlResponse">
  <s:complexType>
    <s:sequence>
```

```

        <s:element name="CanCreateDwsUrlResult" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
</s:complexType>
</s:element>

```

CanCreateDwsUrlResult: This element contains the return value for the operation. The string **MUST** be a standalone **XML** element specified in the following format:

```

<s:complexType>
  <s:choice>
    <s:element name="Error" type="Error"/>
    <s:element name="Result" type="s:string"/>
  </s:choice>
</s:complexType>

```

Error: See section [2.2.3.2](#) for specification.

Result: A site-relative URL that specifies where the Document Workspace is created.

3.1.4.2 CreateDws

This operation creates a new Document Workspace.

The protocol client sends a **CreateDwsSoapIn** request message, and the protocol server responds with a **CreateDwsSoapOut** response message, as follows:

```

<wsdl:operation name="CreateDws">
  <wsdl:input message="tns:CreateDwsSoapIn" />
  <wsdl:output message="tns:CreateDwsSoapOut" />
</wsdl:operation>

```

The protocol client assembles the necessary parameters in a **CreateDws** element and sends the request in a **CreateDwsSoapIn** request message. The protocol server **MUST** create the specified Document Workspace and creates a **CreateDwsResponse** response message with a **Result** element, except as noted in the following text.

The protocol server **MUST** reply with an HTTP 401 error if the authenticated user is not authorized to create the Document Workspace.

The protocol server **MUST** reply with an **Error** element in the **CreateDwsResponse** response message if it fails to create the specified Document Workspace.

The protocol server **MUST** return a **CreateDwsResponse** response message with either a **Result** element or an **Error** element. The **CreateDwsResponse** response message **MUST NOT** be empty.

3.1.4.2.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
CreateDwsSoapIn	Initiates a CreateDws operation, as specified in section 3.1.4.2 , on a protocol server.
CreateDwsSoapOut	Response to a CreateDwsSoapIn request.

3.1.4.2.1.1 CreateDwsSoapIn

This message initiates a **CreateDws** operation on a protocol server. The **CreateDws** operation creates a Document Workspace.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/CreateDws
```

The SOAP body contains a **CreateDws** element.

3.1.4.2.1.2 CreateDwsSoapOut

The **CreateDwsSoapOut** response message is the response to a **CreateDwsSoapIn** request message.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/CreateDws
```

The SOAP body contains a **CreateDwsResponse** element.

3.1.4.2.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
CreateDws	Contains the parameters for the CreateDws operation, as specified in section 3.1.4.2 .
CreateDwsResponse	Contains the response to the CreateDws operation request.

3.1.4.2.2.1 CreateDws

This element contains the parameters for the **CreateDws** operation. This element is defined as follows:

```
<s:element name="CreateDws">
  <s:complexType>
    <s:sequence>
      <s:element name="name" type="s:string"/>
      <s:element name="users" type="s:string"/>
      <s:element name="title" type="s:string"/>
      <s:element name="documents" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

name: Specifies the name of the Document Workspace site. This parameter MAY be empty. If this is non-empty and another site with the same name already exists on the site on which the workspace is being created, the protocol server MUST return a **ServerFailure** error code (see section [2.2.3.2](#)). If empty, the protocol server MUST use the *title* parameter as the name of the new workspace.

users: String in XML that specifies the users to be added as contributors in the Document Workspace site. This string MAY be empty.

When the string is not empty, the XML MUST conform to the following **XML schema definition (XSD)**:

```
<s:complexType>
  <s:element name="items">
    <s:complexType>
      <s:sequence>
        <s:element name="item" maxOccurs="unbounded">
          <xs:complexType>
            <s:attribute name="Name" type="s:string" use="required"/>
            <s:attribute name="Email" type="s:string" use="required"/>
          </xs:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
  </s:element>
</s:complexType>
```

Name: Friendly name for a user.

E-mail: E-mail address of the user.

title: The title for the workspace. This string MAY be empty. If the *name* parameter is empty, the protocol server MUST use the *title* parameter as the name of the new workspace when there is no site associated with that name, or generate a unique name from the *title* as the name of the new workspace. If the *name* and *title* parameters are empty, the protocol server MUST generate a new unique GUID to use as the name of the new workspace.

Documents: String in XML format that contains information to be stored as a key-value pair in the site metadata. This string MAY be empty. If it is not empty, it MUST conform to the following XSD:

```
<s:complexType>
  <s:element name="items">
    <s:complexType>
      <s:sequence>
        <s:element name="item" maxOccurs="unbounded">
          <xs:complexType>
            <s:attribute name="Name" type="s:string" />
            <s:attribute name="ID" type="s:string" />
          </xs:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
  </s:element>
</s:complexType>
```

Name: A string containing a folder-relative URL that identifies a document.

ID: A unique string that represents a document key.

3.1.4.2.2 CreateDwsResponse

This element returns the protocol server response to the protocol client. This element is defined as follows:

```
<s:element name="CreateDwsResponse">
  <s:complexType>
    <s:sequence>
```

```

    <s:element name="CreateDwsResult" type="s:string" minOccurs="0" maxOccurs="1" />
  </s:sequence>
</s:complexType>
</s:element>

```

CreateDwsResult: String that contains stand-alone XML. The string MUST conform to the following XSD:

```

<s:complexType>
  <s:choice>
    <s:element ref="tns:Error"/>
    <s:element name="Results">
      <s:complexType>
        <s:sequence>
          <s:element name="Url" type="s:string" />
          <s:element name="DoclibUrl" type="s:string" />
          <s:element name="ParentWeb" type="s:string"/>
          <s:element name="FailedUsers" type="tns:UserType" minOccurs="0"
maxOccurs="unbounded"/>
          <s:element name="AddUsersUrl" type="s:string"/>
          <s:element name="AddUsersRole" type="s:string"/>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:choice>
</s:complexType>

```

Error: See section [2.2.3.2](#) for specification. Possible values for this operation are listed in the following table.

Error code	Identifier	Description
QuotaExceeded	14	User creating the workspace has exceeded the allocated resource quota.
ServerFailure	1	Protocol server encountered an error during the attempt to create the workspace.
AlreadyExists <5>	13	A site with the specified name already exists on this site.

Url: URL for the new workspace. This MUST be an **absolute URL**.

DoclibUrl: Site-relative URL for the shared documents list associated with the workspace.

ParentWeb: Title of the site (2) on which the workspace is created. If the workspace is the root site of the site collection, this MUST be the title of the new workspace.

FailedUsers: A list of users from the **users** field of **CreateDws** that could not be added to the list of authorized users in the new workspace. It is standalone XML that MUST use the following definition:

```

<s:complexType name="UserType">
  <s:sequence>
    <s:element name="User" minOccurs="0">
      <s:complexType>
        <s:attribute name="Email" type="s:string"/>
      </s:complexType>
    </s:element>
  </s:sequence>

```

```
</s:complexType>
```

AddUsersUrl: An absolute URL to a Web page that provides the ability to add users to the workspace.

AddUsersRole: Role for adding users to the workspace. If the *users* parameter to **CreateDws** (section [3.1.4.2.2.1](#)) was not empty, the value MUST be the string "Microsoft.SharePoint.SPRoleDefinition".

3.1.4.3 CreateFolder

This operation creates a **folder** in the document library of the current Document Workspace site.

The protocol client sends a **CreateFolderSoapIn** request message, and the protocol server responds with a **CreateFolderSoapOut** response message, as follows:

```
<wsdl:operation name="CreateFolder">
  <wsdl:input message="tns:CreateFolderSoapIn" />
  <wsdl:output message="tns:CreateFolderSoapOut" />
</wsdl:operation>
```

If the parent folder for the specified URL does not exist, the protocol server MUST return an **Error** element with a **FolderNotFound** error code.

If the specified URL already exists, the protocol server MUST return an **Error** element with an **AlreadyExists** error code.

If the user does not have sufficient access permissions to create the folder, the protocol server MUST return an **Error** element with a **NoAccess** error code.

The protocol server MUST return an **Error** element with a **Failed** or a **ServerFailure** error code if an unspecified error prevents creating the specified folder.

If none of the prior conditions apply, the protocol server MUST create the folder specified in the **CreateFolder** element.

3.1.4.3.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
CreateFolderSoapIn	Requests a CreateFolder operation, as specified in section 3.1.4.3 , on the protocol server.
CreateFolderSoapOut	Response to a CreateFolderSoapIn request.

3.1.4.3.1.1 CreateFolderSoapIn

This message requests a **CreateFolder** operation on the protocol server.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/CreateFolder
```

The SOAP body contains a **CreateFolder** element.

3.1.4.3.1.2 CreateFolderSoapOut

This message returns the response.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/CreateFolder
```

The SOAP body contains a **CreateFolderResponse** element.

3.1.4.3.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
CreateFolder	Contains the parameter for the CreateFolder operation, as specified in section 3.1.4.3 .
CreateFolderResponse	Contains the response to a CreateFolder operation request.

3.1.4.3.2.1 CreateFolder

The **CreateFolder** element provides a holder for the parameter for the **CreateFolder** operation. This element is defined as follows:

```
<s:element name="CreateFolder">  
  <s:complexType>  
    <s:sequence>  
      <s:element name="url" type="s:string" minOccurs="0" maxOccurs="1"/>  
    </s:sequence>  
  </s:complexType>  
</s:element>
```

url: This is a site-relative URL with the full path for the new folder [6](#).

3.1.4.3.2.2 CreateFolderResponse

This element contains the response from the protocol server. This element is defined as follows:

```
<s:element name="CreateFolderResponse">  
  <s:complexType>  
    <s:sequence>  
      <s:element name="CreateFolderResult" type="s:string" minOccurs="0" maxOccurs="1"/>  
    </s:sequence>  
  </s:complexType>  
</s:element>
```

CreateFolderResult: This element contains a string that is interpreted as standalone XML. The element **MUST** contain either an **Error** element or a **Result** element as follows:

```
<s:complexType>  
  <s:choice>  
    <s:element name="Result"/>
```

```

        <s:element ref="tns:Error"/>
    </s:choice>
</s:complexType>

```

Error: An **Error** element as specified in section [2.2.3.2](#).

Result: An empty **Result** element ("`<Result/>`") if the call is successful.

3.1.4.4 DeleteDws

This operation deletes a Document Workspace from the protocol server.

The protocol client sends a **DeleteDwsSoapIn** request message, and the protocol server responds with a **DeleteDwsSoapOut** response message, as follows:

```

<wsdl:operation name="DeleteDws">
    <wsdl:input message="tns>DeleteDwsSoapIn" />
    <wsdl:output message="tns>DeleteDwsSoapOut" />
</wsdl:operation>

```

The protocol server MUST return an **Error** element with a **NoAccess** code if the authenticated user is not authorized to delete the Document Workspace. If the specified Document Workspace has subsites, the protocol server MUST return an **Error** element with the **WebContainsSubwebs** error code.

If the specified Document Workspace does not exist, the protocol server SHOULD [<7>](#) return HTTP status code 404 with response body which contains text "404 FILE NOT FOUND".

If the specified Document Workspace is the root site of the site collection, the protocol server MUST return an **Error** element with the **ServerFailure** error code.

If none of the prior conditions apply, the protocol server MUST delete the specified Document Workspace and return a **Result** element.

3.1.4.4.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
DeleteDwsSoapIn	Request for the DeleteDws operation, as specified in section 3.1.4.4 .
DeleteDwsSoapOut	Response to a DeleteDwsSoapIn request that returns the operation status.

3.1.4.4.1.1 DeleteDwsSoapIn

This message requests the **DeleteDws** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/DeleteDws
```

The SOAP body contains a **DeleteDws** element.

3.1.4.4.1.2 DeleteDwsSoapOut

This message returns the operation status.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/DeleteDws
```

The SOAP body contains a **DeleteDwsResponse** element.

3.1.4.4.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
DeleteDws	Contains the parameters for a DeleteDws operation, as specified in section 3.1.4.4 .
DeleteDwsResponse	Contains the response to a DeleteDws operation request.

3.1.4.4.2.1 DeleteDws

This element requests a **DeleteDws** operation. This element is defined as follows:

```
<s:element name="DeleteDws">  
  <s:complexType/>  
</s:element>
```

3.1.4.4.2.2 DeleteDwsResponse

This element contains the response from the protocol server. It is defined as follows:

```
<s:element name="DeleteDwsResponse">  
  <s:complexType>  
    <s:sequence>  
      <s:element name="DeleteDwsResult" type="s:string" minOccurs="0" maxOccurs="1"/>  
    </s:sequence>  
  </s:complexType>  
</s:element>
```

DeleteDwsResult: This element contains the protocol server response to the **DeleteDws** operation. This element contains a string that is stand-alone XML. The string MUST conform to either the **Error** element or a **Result** element as follows:

```
<s:complexType>  
  <s:choice>  
    <s:element ref="tns:Error"/>  
    <s:element name="Result"/>  
  </s:choice>  
</s:complexType>
```

Error: An **Error** element as specified in section [2.2.3.2](#).

Result: An empty **Result** element ("`<Result/>`") if the call is successful.

3.1.4.5 DeleteFolder

This operation deletes a folder from a document library on the site.

The protocol client sends a **DeleteFolderSoapIn** request message, and the protocol server responds with a **DeleteFolderSoapOut** response message, as follows:

```
<wsdl:operation name="DeleteFolder">
  <wsdl:input message="tns:DeleteFolderSoapIn" />
  <wsdl:output message="tns:DeleteFolderSoapOut" />
</wsdl:operation>
```

If the parent of the specified URL does not exist, the protocol server MUST return an **Error** element with a **FolderNotFound** error code.

If the specified URL does not exist, the protocol server MUST return a **Result** element as specified in **DeleteFolderResponse** (section [3.1.4.5.2.2](#)).

If an unspecified error prevents the deletion of the specified folder, the protocol server MUST return an **Error** element with a **Failed** or a **ServerFailure** error code.

If none of the prior conditions apply, the protocol server MUST delete the folder specified in the **CreateFolder** element and return a **Result** element as specified in **DeleteFolderResponse** (section [3.1.4.5.2.2](#)).

3.1.4.5.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
DeleteFolderSoapIn	Request for a DeleteFolder operation, as specified in section 3.1.4.5 .
DeleteFolderSoapOut	Response to a DeleteFolderSoapIn request.

3.1.4.5.1.1 DeleteFolderSoapIn

This message requests the **DeleteFolder** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/DeleteFolder
```

The SOAP body contains a **DeleteFolder** element.

3.1.4.5.1.2 DeleteFolderSoapOut

This message responds to a protocol client.

The SOAP action value of the message is defined as follows:

The SOAP body contains a **DeleteFolderResponse** element.

3.1.4.5.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
DeleteFolder	Contains the parameter for the DeleteFolder operation, as specified in section 3.1.4.5 .
DeleteFolderResponse	Contains the response to the DeleteFolder operation request.

3.1.4.5.2.1 DeleteFolder

The **DeleteFolder** element contains the parameter for the **DeleteFolder** operation. This element is defined as follows:

```
<s:element name="DeleteFolder">
  <s:complexType>
    <s:sequence>
      <s:element name="url" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

url: This is a site-relative URL specifying the folder to delete.

3.1.4.5.2.2 DeleteFolderResponse

The **DeleteFolderResponse** element contains the response from the protocol server. This element is defined as follows:

```
<s:element name="DeleteFolderResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="DeleteFolderResult" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

DeleteFolderResult: This element is a string that contains stand-alone XML. The XML MUST conform to either the **Error** element or a **Result** element as follows:

```
<s:complexType>
  <s:choice>
    <s:element ref="tns:Error"/>
    <s:element name="Result"/>
  </s:choice>
</s:complexType>
```

Error: An **Error** element as specified in section [2.2.3.2](#).

Result: An empty **Result** element ("`<Result/>`") if the call is successful.

3.1.4.6 FindDwsDoc

This operation obtains a URL for a named document in a Document Workspace.

The protocol client sends a **FindDwsDocSoapIn** request message, and the protocol server responds with a **FindDwsDocSoapOut** response message, as follows:

```
<wsdl:operation name="FindDwsDoc">
  <wsdl:input message="tns:FindDwsDocSoapIn" />
  <wsdl:output message="tns:FindDwsDocSoapOut" />
</wsdl:operation>
```

If the protocol server cannot locate a document with the specified identifier, it MUST return an **Error** element with a code of **ItemNotFound**.

If the protocol server encounters another error that prevents it from providing a response with the correct URL, it MUST return an **Error** element with a code of **ServerFailure**.

If none of the prior conditions apply, the protocol server SHOULD reply with a **Result** element as specified in **FindDwsDocResponse** containing an absolute URL for the specified document.

3.1.4.6.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
FindDwsDocSoapIn	Request for a FindDwsDoc operation, as specified in section 3.1.4.6 .
FindDwsDocSoapOut	Response to a FindDwsDocSoapIn request.

3.1.4.6.1.1 FindDwsDocSoapIn

This message initiates a **FindDwsDoc** WSDL operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/FindDwsDoc
```

The SOAP body contains a **FindDwsDoc** element.

3.1.4.6.1.2 FindDwsDocSoapOut

This message provides a response.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/FindDwsDoc
```

The SOAP body contains a **FindDwsDocResponse** element.

3.1.4.6.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
FindDwsDoc	Contains the parameter for the FindDwsDoc operation, as specified in section 3.1.4.6 .
FindDwsDocResponse	Contains the response to a FindDwsDoc operation request.

3.1.4.6.2.1 FindDwsDoc

This element contains the *id* parameter for the **FindDwsDoc** operation. This element is defined as follows:

```
<s:element name="FindDwsDoc">
  <s:complexType>
    <s:sequence>
      <s:element name="id" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

id: See the description of the **ID** attribute of the **Documents** element of **CreateDws**, specified in section [3.1.4.2.2.1](#).

3.1.4.6.2.2 FindDwsDocResponse

This element contains the response from the protocol server. This element is defined as follows:

```
<s:element name="FindDwsDocResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="FindDwsDocResult" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

FindDwsDocResult: This element contains a string that is stand-alone XML encoded either as an **Error** element as specified in [2.2.3.2](#) or a **Result** element defined as follows:

```
<s:complexType>
  <s:choice>
    <s:element ref="tns:Error"/>
    <s:element name="Result" type="s:string"/>
  </s:choice>
</s:complexType>
```

Error: An **Error** element as specified in section 2.2.3.2.

Result: A **Result** element for which the content MUST be an absolute URL that refers to the requested document.

3.1.4.7 GetDwsData

This operation returns general information about the Document Workspace site, as well as its members, documents, links, and tasks.

The protocol client sends a **GetDwsDataSoapIn** request message, and the protocol server responds with a **GetDwsDataSoapOut** response message, as follows:

```
<wsdl:operation name="GetDwsData">
  <wsdl:input message="tns:GetDwsDataSoapIn" />
  <wsdl:output message="tns:GetDwsDataSoapOut" />
</wsdl:operation>
```

If the protocol client provides a non-empty *document* parameter and the protocol server cannot locate the document, the protocol server SHOULD return an **Error** element with the **ListNotFound** code.

If the protocol server detects an access restriction during processing, it MUST return an **Error** with the **NoAccess** code and a URL for an authentication page.

If the protocol server detects some other problem during processing, it MUST return an **Error** with the **ServerFailure** code.

If no **Error** elements are returned as previously described, the protocol server MUST return a **Result** element with the information for the Document Workspace and document.

3.1.4.7.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetDwsDataSoapIn	Request for a GetDwsData operation, as specified in section 3.1.4.7 .
GetDwsDataSoapOut	Response to a GetDwsDataSoapIn request.

3.1.4.7.1.1 GetDwsDataSoapIn

This message requests a **GetDwsData** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/GetDwsData
```

The SOAP body contains a **GetDwsData** element.

3.1.4.7.1.2 GetDwsDataSoapOut

This message returns the results of a **GetDwsData** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/GetDwsData
```

The SOAP body contains a **GetDwsDataResponse** element.

3.1.4.7.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetDwsData	Contains the parameters for the GetDwsData operation, as specified in section 3.1.4.7 .
GetDwsDataResponse	Contains the response to a GetDwsData operation request.

3.1.4.7.2.1 GetDwsData

This element provides the parameters for the **GetDwsData** operation. This element is defined as follows:

```
<s:element name="GetDwsData">
  <s:complexType>
    <s:sequence>
      <s:element name="document" type="s:string" minOccurs="0"/>
      <s:element name="lastUpdate" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

document: A site-based URL of a document in the document library in the Document Workspace. This argument provides the context for the documents list in the reply.

lastUpdate: Contains the **lastUpdate** value returned in the result of a previous **GetDwsData** or **GetDwsMetaData** operation, or an empty string. If the list in the Document Workspace has not changed since **lastUpdate**, **GetDwsData** MUST return a **NoChanges** child element of the **List** element as specified in **GetDwsDataResponse**. If the protocol client provides an empty string, the protocol server MUST provide all data for the specified context.

3.1.4.7.2.2 GetDwsDataResponse

This element returns the response to the **GetDwsData** operation. This element is defined as follows:

```
<s:element name="GetDwsDataResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetDwsDataResult" type="tns:GetDwsDataResultType" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetDwsDataResult: This structure is a string that contains stand-alone XML. The XML MUST conform to the following schema:

```
<s:complexType name="GetDwsDataResultType">
  <s:choice>
    <s:element ref="tns:Error"/>
    <s:element ref="tns:Results"/>
  </s:choice>
</s:complexType>
```

Error: An **Error** element as specified in section [2.2.3.2](#).

Results: The **Results** element contains the requested data as a string. It MUST conform to the following schema:

```
<s:element name="Results">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Title" type="xs:string"/>
      <xs:element name="LastUpdate" type="xs:integer"/>
      <xs:element name="User">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ID" type="xs:string"/>
            <xs:element name="Name" type="xs:string"/>
            <xs:element name="LoginName" type="xs:string"/>
            <xs:element name="Email" type="xs:string"/>
            <xs:element name="IsDomainGroup">
              <s:simpleType>
                <s:restriction base="s:string">
                  <s:enumeration value="True" />
                  <s:enumeration value="False" />
                </s:restriction>
              </s:simpleType>
            </xs:element>
            <xs:element name="IsSiteAdmin">
              <s:simpleType>
                <s:restriction base="s:string">
                  <s:enumeration value="True" />
                  <s:enumeration value="False" />
                </s:restriction>
              </s:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Members" type="tns:MemberData"/>
      <xs:sequence minOccurs="0">
        <xs:element ref="tns:Assignees"/>
        <xs:element ref="tns:List"/>
        <xs:element ref="tns:List"/>
        <xs:element ref="tns:List"/>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</s:element>
```

Title: The title of the workspace.

LastUpdate: An integer indicating the last time that the workspace was updated. This value MAY be used by a protocol client during **GetDwsData** or **GetDwsMetaData** operations.

User: The member information for the user who requested the **GetDwsData** operation. The content of this element MUST be a **Member** element as specified in section [2.2.3.5](#), with the exception that both **IsDomainGroup** and **Email** MUST be present. In addition, the **IsSiteAdmin** element MUST be present and MUST contain a Boolean value.

Members: This element MUST conform to the schema for the complex data type **MemberData**.

The following elements MUST be present for the **GetDwsData** operation. The elements MUST NOT be present if this data is being returned by **GetDwsMetaData** and the *minimal* parameter is set to **true**.

Assignees: This element specifies the users assigned to the workspace. This element MUST conform to the **Assignees** element schema specified in section [2.2.3.1](#).

List: This element contains information about the Tasks list. The **Name** attribute MUST be set to "Tasks". The type of the **List** element MUST be List (section [2.2.3.4](#)).

List: This element contains information about the Documents list. The **Name** attribute MUST be set to "Documents". The type of the **List** element MUST be List (specified in section 2.2.3.4). If the URL specified in the **GetDwsData** request cannot be found, the protocol server MUST return an **Error** element with a code of **ListNotFound**.

List: This element contains information about the Links list. The **Name** attribute MUST be set to "Links". The type of the **List** element MUST be List (specified in section 2.2.3.4).

3.1.4.8 GetDwsMetaData

This operation returns information about a Document Workspace site and the lists that it contains.

The protocol client sends a **GetDwsMetaDataSoapIn** request message, and the protocol server responds with the following **GetDwsMetaDataSoapOut** response message:

```
<wsdl:operation name="GetDwsMetaData">
  <wsdl:input message="tns:GetDwsMetaDataSoapIn" />
  <wsdl:output message="tns:GetDwsMetaDataSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetDwsMetaDataSoapIn** request message containing the **GetDwsMetaData** element with *operation* parameters. The protocol server processes the operation and returns its results in a **GetDwsMetaDataResponse** element. The element contains a string that is stand-alone XML. The schema of the XML is specified in the following sections. Because of the complexity of the return structure, the details are discussed in the corresponding schema definition section.

3.1.4.8.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetDwsMetaDataSoapIn	Request for a GetDwsMetaData operation, as specified in section 3.1.4.8 .
GetDwsMetaDataSoapOut	Response to a GetDwsMetaDataSoapIn request.

3.1.4.8.1.1 GetDwsMetaDataSoapIn

This message requests the **GetDwsMetaData** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/GetDwsMetaData
```

The SOAP body contains a **GetDwsMetaData** element.

3.1.4.8.1.2 GetDwsMetaDataSoapOut

This message returns the results of a **GetDwsMetaData** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/GetDwsMetaData
```

The SOAP body contains a **GetDwsMetaDataResponse** element.

3.1.4.8.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetDwsMetaData	Contains the parameters for the GetDwsMetaData operation, as specified in section 3.1.4.8 .
GetDwsMetaDataResponse	Contains the response to a GetDwsMetaData operation request.
ListInfo	Contains metadata about a list in a Document Workspace.

3.1.4.8.2.1 GetDwsMetaData

This element contains the parameters to the **GetDwsMetaData** operation. This element is defined as follows:

```
<s:element name="GetDwsMetaData">
  <s:complexType>
    <s:sequence>
      <s:element name="document" type="s:string" minOccurs="0"/>
      <s:element name="id" type="s:string" minOccurs="0"/>
      <s:element name="minimal" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

document: This is a site-relative URL that specifies the list or document to describe in the response.

id: See the **ID** attribute of the **Documents** element of **CreateDws**, specified in section [3.1.4.2.2.1](#). If this parameter is not NULL or empty, the server will ignore the *document* parameter, and get a value according to the key **id** from the **Documents** key-value pair, then update the document parameter with the value.

minimal: A Boolean value that specifies whether to return information. If **false**, the protocol server MUST return the following elements from the **Results** element of **GetDwsMetaData**, and MUST NOT return them otherwise:

- **SubscribeUrl**
- **ListInfo (Tasks)**
- **ListInfo (Documents)**
- **ListInfo (Links)**

If **false** and the workspace that **document** belongs to contains a **Tasks** list, the protocol server will return the **Schema (Tasks)** element, and otherwise not.

If **false** and the workspace that **document** belongs to contains a **DocumentLibrary** list, the protocol server will return the **Schema (Documents)** element, and otherwise not.

If **false** and the workspace that **document** belongs to contains a **Links** list, the protocol server will return the **Schema (Links)** element, and otherwise not.

3.1.4.8.2.2 GetDwsMetaDataResponse

This element contains the protocol server response to a **GetDwsMetaData** operation. The element contains a string that is stand-alone XML, defined as follows:

```
<s:element name="GetDwsMetaDataResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="GetDwsMetaDataResult" type="tns:GetDwsMetaDataResultType"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetDwsMetaDataResult: This element contains a string that is stand-alone XML, encoded either as an **Error** element as specified in section [2.2.3.2](#) or a **Results** element defined as follows:

```
<s:complexType name="GetDwsMetaDataResultType">
  <s:choice>
    <s:element ref="tns:Error"/>
    <s:element name="Results">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="SubscribeUrl" type="xs:string" minOccurs="0"/>
          <xs:element name="MtgInstance" type="xs:string"/>
          <xs:element name="SettingUrl" type="xs:string"/>
          <xs:element name="PermsUrl" type="xs:string"/>
          <xs:element name="UserInfoUrl" type="xs:string"/>
          <xs:element ref="tns:Roles"/>
          <xs:element ref="tns:Schema" minOccurs="0" maxOccurs="3"/>
          <xs:element ref="tns:ListInfo" minOccurs="0" maxOccurs="3"/>
          <xs:element name="Permissions">
            <xs:complexType>
              <s:choice>
                <s:element ref="tns:Error"/>
                <s:sequence>
                  <xs:element name="ManageSubwebs" minOccurs="0"/>
                  <xs:element name="ManageWeb" minOccurs="0"/>
                  <xs:element name="ManageRoles" minOccurs="0"/>
                  <xs:element name="ManageLists" minOccurs="0"/>
                  <xs:element name="InsertListItems" minOccurs="0"/>
                  <xs:element name="EditListItems" minOccurs="0"/>
                  <xs:element name="DeleteListItems" minOccurs="0"/>
                </s:sequence>
              </s:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="HasUniquePerm"/>
          <xs:element name="WorkspaceType"/>
          <xs:element name="IsADMode"/>
          <xs:element name="DocUrl"/>
          <xs:element name="Minimal"/>
          <s:element ref="tns:Results"/>
        </xs:sequence>
      </xs:complexType>
    </s:element>
  </s:choice>
</s:complexType>
```

```

    </s:element>
  </s:choice>
</s:complexType>

```

If the element is an **Error**, it MUST contain an error code as specified in section 2.2.3.2. It MUST contain one of the error codes from the following table.

Error code	Description
DocumentNotFound	The site or document could not be found.
NoAccess	The authenticated user is not permitted to access this information. The Error element MUST contain the URL of a page where the user can request access permission. <8>
ServerFailure	The protocol server encountered an unspecified error during operation processing.

SubscribeUrl: This is the **URI** of a page that enables users to subscribe to changes in the document specified in the parameters. See [\[MS-OSALER\]](#) and [\[MS-ALERTSS\]](#) for more information about alert subscriptions.

MtgInstance: If the **document** element specifies a meeting item, this element MUST represent a string that contains the meeting information. The meeting information contains the instance identifier of the meeting. Otherwise, the element SHOULD be empty.

SettingUrl: URI of a page that enables workspace settings to be modified.

PermsUrl: URI of a page that enables the workspace permissions settings to be modified.

UserInfoUrl: URI of a page that enables the list of users to be modified.

Roles: Specifies the roles that apply to the workspace. This element MUST conform to the element specification in section [2.2.3.6](#).

Schema: Contains the schema information for the following workspace lists appearing in a sequence as below. The element MUST conform to the **Schema** element in the **xml** element specified in section [2.2.4.2](#).

- **Tasks** [<9>](#)
- **Documents**
- **Links**

ListInfo: This element contains list information for the following workspace lists appearing in a sequence as below. This element MUST conform to the **ListInfo** element specified in section [3.1.4.8.2.3](#).

- **Tasks**
- **Documents**
- **Links**

Permissions: This element contains the permissions of the authenticated user for this Document Workspace. This element contains elements that indicate permissible operations. The list of elements that MAY be present are described in the WSDL, specified in section 6. Each element represents the ability to perform the action for which it is named.

HasUniquePerm: Set to **True** if, and only if, the workspace has custom role assignments; otherwise, if role assignments are inherited from the site in which the workspace is created, set to **False**.

WorkspaceType: This value MUST be "DWS", "MWS", or an empty string. "DWS" identifies the workspace as a Document Workspace; "MWS" identifies it as a Meeting Workspace. A Meeting Workspace is a convenient and centralized place for project collaboration and meeting proceedings. If the site is not one of those types, an empty string MUST be returned.

IsADMode: Set to **True** if, and only if, the workspace is set to **Active Directory** mode, a mode in which, when new users are created in the site, new site member accounts are created automatically in **Active Directory Domain Services (AD DS)**; otherwise set to **False**.

DocUrl: This SHOULD be set to the value of **document** from the **GetDwsMetaData** request or be empty if the value of **document** is not specified in the **GetDwsMetaData** request.

Minimal: This element contains the **minimal** flag from the **GetDwsMetaData** request. This value MUST match the value in the request.

Results: This element is identical to **Results** element of the **GetDwsDataResult** element specified in section [3.1.4.7.2.2](#). Note the effect of the **minimal** flag on the contained elements.

3.1.4.8.2.3 ListInfo

The **ListInfo** element contains metadata about a list in a Document Workspace. This element is defined as follows:

```
<xs:element name="ListInfo">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="tns:Error" minOccurs="0"/>
      <xs:sequence>
        <xs:element name="Moderated"/>
        <s:simpleType>
          <s:restriction base="s:string">
            <s:enumeration value="True" />
            <s:enumeration value="False" />
          </s:restriction>
        </s:simpleType>
      </xs:element>
      <xs:element name="ListPermissions">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="InsertListItems" minOccurs="0"/>
            <xs:element name="EditListItems" minOccurs="0"/>
            <xs:element name="DeleteListItems" minOccurs="0"/>
            <xs:element name="ManageLists" minOccurs="0"/>
            <xs:element ref="tns:Error" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
    <xs:attribute name="Name" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
```

Error: The **Error** element is returned when there is an error accessing the list information. The error code values for this element are shown in the following table.

Error code	Description
NoAccess	The current user does not have sufficient permissions to access list permission information. The AccessUrl attribute MUST NOT be present in this element. <10>
ServerFailure	The current user does not have sufficient permissions to view the list.

Moderated: Boolean value that specifies whether adding new entries to the list requires the approval of the list owner.

ListPermissions: This element displays the current user permissions that are associated with the list. The following elements MUST be present if, and only if, the user has that permission:

InsertListItems, **EditListItems**, **DeleteListItems**, and **ManageLists**.

InsertListItems: Specifies that the current user can add new **items** to the list.

EditListItems: Specifies that the current user can edit **list items**.

DeleteListItems: Specifies that the current user can delete list items.

ManageLists: Specifies that the current user can manage the list.

Error: This element is present if an error is encountered when attempting to get permission information.

Name: Contains the name of the list.

3.1.4.9 RemoveDwsUser

This operation deletes a user from a Document Workspace.

The protocol client sends a **RemoveDwsUserSoapIn** request message, and the protocol server responds with a **RemoveDwsUserSoapOut** response message, as follows:

```
<wsdl:operation name="RemoveDwsUser">
  <wsdl:input message="tns:RemoveDwsUserSoapIn" />
  <wsdl:output message="tns:RemoveDwsUserSoapOut" />
</wsdl:operation>
```

If the protocol server successfully deletes the specified user from the workspace members list, the protocol server MUST return a string containing an empty **Result** element as follows:

```
<s:element name="Results">
  <s:complexType/>
</s:element>
```

If an error of any type occurs during the processing, the protocol server MUST return an **Error** element as specified in section [2.2.3.2](#) with an error code of **ServerFailure**.

3.1.4.9.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
RemoveDwsUserSoapIn	Request for a RemoveDwsUser operation, as specified in section 3.1.4.9 .
RemoveDwsUserSoapOut	Response to a RemoveDwsUserSoapIn request.

3.1.4.9.1.1 RemoveDwsUserSoapIn

This message requests to delete a user from a Document Workspace.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/RemoveDwsUser
```

The SOAP body contains a **RemoveDwsUser** element.

3.1.4.9.1.2 RemoveDwsUserSoapOut

This message returns the status of a requested **RemoveDwsUser** operation from the protocol server to the protocol client.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/RemoveDwsUser
```

The SOAP body contains a **RemoveDwsUserResponse** element.

3.1.4.9.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
RemoveDwsUser	Contains the parameter for the RemoveDwsUser operation, as specified in section 3.1.4.9 .
RemoveDwsUserResponse	Contains the response to a RemoveDwsUser operation request.

3.1.4.9.2.1 RemoveDwsUser

This element contains the *input* parameter for the **RemoveDwsUser** operation. This element is defined as follows:

```
<s:element name="RemoveDwsUser">
  <s:complexType>
    <s:sequence>
      <s:element name="id" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

ID: The **user identifier** of the user to remove from the workspace. This positive integer MUST be in the range from zero through 2,147,483,647, inclusive.

3.1.4.9.2.2 RemoveDwsUserResponse

This element contains the result for a **RemoveDwsUser** operation. This element is defined as follows:

```
<s:element name="RemoveDwsUserResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="RemoveDwsUserResult" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

RemoveDwsUserResult: This element contains a string that is stand-alone XML as follows:

```
<s:complexType>
  <xs:choice>
    <xs:element ref="tns:Error"/>
    <xs:element name="Results"/>
  </xs:choice>
</s:complexType>
```

Error: An **Error** element as specified in section [2.2.3.2](#).

Results: This is an empty element that indicates success.

3.1.4.10 RenameDws

This operation changes the title of a Document Workspace.

The protocol client sends a **RenameDwsSoapIn** request message, and the protocol server responds with a **RenameDwsSoapOut** response message, as follows:

```
<wsdl:operation name="RenameDws">
  <wsdl:input message="tns:RenameDwsSoapIn" />
  <wsdl:output message="tns:RenameDwsSoapOut" />
</wsdl:operation>
```

If a processing failure prevents the protocol server from recording the new title, the protocol server MUST return an **Error** element with a **Failed** code.

If the user submitting the request is not authorized to change the title, the protocol server MUST return an **Error** element with a **NoAccess** code. The **Error** element MUST NOT contain an **AccessUri** attribute.

If another error occurs during processing, the protocol server MUST return an **Error** element with a code of **ServerFailure**.

If the protocol server successfully changes the title of the workspace, it MUST return an empty **Result** element.

3.1.4.10.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
RenameDwsSoapIn	Request for a RenameDws operation, as specified in section 3.1.4.10 .
RenameDwsSoapOut	Response to a RenameDwsSoapIn request.

3.1.4.10.1.1 RenameDwsSoapIn

This message requests a **RenameDws** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/RenameDws
```

The SOAP body contains a **RenameDws** element.

3.1.4.10.1.2 RenameDwsSoapOut

This message returns a response to the **RenameDws** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/RenameDws
```

The SOAP body contains a **RenameDwsResponse** element.

3.1.4.10.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
RenameDws	Contains the parameter for the RenameDws operation, as specified in section 3.1.4.10 .
RenameDwsResponse	Contains the response to a RenameDws operation request.

3.1.4.10.2.1 RenameDws

This element contains the parameter to the **RenameDws** operation. This element is defined as follows:

```
<s:element name="RenameDws">
  <s:complexType>
    <s:sequence>
      <s:element name="title" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

title: This is a string that contains the new title of the workspace.

3.1.4.10.2.2 RenameDwsResponse

This element contains the protocol server reply to the **RenameDws** request. This element is defined as follows:

```
<s:element name="RenameDwsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="RenameDwsResult" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

RenameDwsResult: This element contains a string that is standalone XML as follows:

```
<xs:complexType>
  <xs:choice>
    <xs:element ref="tns:Error"/>
    <xs:element name="Result"/>
  </xs:choice>
</xs:complexType>
```

Error: An **Error** element as specified in section [2.2.3.2](#).

Result: This is an empty element that indicates success.

3.1.4.11 UpdateDwsData

This operation modifies the metadata of a Document Workspace. This method is deprecated and SHOULD NOT be called by the protocol client. [<11>](#) This operation is defined as follows:

```
<wsdl:operation name="UpdateDwsData">
  <wsdl:input message="tns:UpdateDwsDataSoapIn" />
  <wsdl:output message="tns:UpdateDwsDataSoapOut" />
</wsdl:operation>
```

The protocol client sends an **UpdateDwsDataSoapIn** request message, and the protocol server responds with an **UpdateDwsDataSoapOut** response message.

If there is a failure during processing of this operation, the protocol server MUST return a **ServerFailure** error, as specified in section [2.2.3.2](#).

If the protocol server does not return an **Error** element, it MUST return a **Result** element as specified in section [3.1.4.11.2.2](#).

3.1.4.11.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
UpdateDwsDataSoapIn	Request for an UpdateDwsData operation, as specified in section 3.1.4.11 .

Message	Description
UpdateDwsDataSoapOut	Response to an UpdateDwsDataSoapIn request.

3.1.4.11.1.1 UpdateDwsDataSoapIn

This message requests an **UpdateDwsData** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/UpdateDwsData
```

The SOAP body contains an **UpdateDwsData** element.

3.1.4.11.1.2 UpdateDwsDataSoapOut

This message returns a response to the **UpdateDwsData** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/dws/UpdateDwsData
```

The SOAP body contains an **UpdateDwsDataResponse** element.

3.1.4.11.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
UpdateDwsData	Contains the parameters for an UpdateDwsData operation, as specified in section 3.1.4.11.1 .
UpdateDwsDataResponse	Contains the response to an UpdateDwsData operation request.

3.1.4.11.2.1 UpdateDwsData

This element contains the parameters for an **UpdateDwsData** operation. This element is defined as follows:

```
<s:element name="UpdateDwsData">
  <s:complexType>
    <s:sequence>
      <s:element name="updates" type="s:string" minOccurs="0"/>
      <s:element name="meetingInstance" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

updates: This is a string that contains **CAML** instructions specifying how to update the workspace information. The format of the string is specified as the **Batch** element in [\[MS-WSSCAP\]](#) section

2.2.3.1. `<?xml version="1.0" encoding="UTF-8"?>` needs to be added in front of the **Batch** element, and the **Version** attribute of the **Batch** element is optional.

meetingInstance: If **UpdateDwsData** is used to update a meeting item, this element MUST represent a string that contains the meeting information. The meeting information contains the instance identifier of the meeting. Otherwise, this element SHOULD be empty.

3.1.4.11.2.2 UpdateDwsDataResponse

This element contains the response to the **UpdateDwsData** operation. This element is defined as follows:

```
<s:element name="UpdateDwsDataResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="UpdateDwsDataResult" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

UpdateDwsDataResult: This is a string formatted as stand-alone XML. If there is a failure during processing of the operation, it MUST contain a **ServerFailure** error, as specified in section [2.2.3.2](#).

If this string does not contain an **Error** element, it MUST contain a string formatted as stand-alone XML that is the result of the CAML instructions, as specified by the **updates** element in section [3.1.4.11.2.1](#). This result is the **Xml** property of an **SPListItemCollection** (that is, **SPListItemCollection.Xml**) that is returned by the **SPList.GetItems(SPQuery)** call, where **SPQuery** contains the CAML instructions as specified by the **updates** element.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

This protocol creates and manages Document Workspaces. A Document Workspace is required before most **WSDL** operations can be performed. Exceptions are the **CanCreateDwsUrl** and **CreateDws** operations.

4.1 CanCreateDwsUrl

This operation tests whether the current user has the permissions necessary to create a Document Workspace at the URL provided by that user. The sequence of messages would resemble the following code.

4.1.1 Request

HTTP: Request, POST: /_vti_bin/DWS.asmx:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CanCreateDwsUrl
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <url>coho</url>
    </CanCreateDwsUrl>
  </soap:Body>
</soap:Envelope>
```

4.1.2 Response

The protocol server would return the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CanCreateDwsUrlResponse
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <CanCreateDwsUrlResult>
        &lt;Result&gt;coho&lt;/Result&gt;
      </CanCreateDwsUrlResult>
    </CanCreateDwsUrlResponse>
  </soap:Body>
</soap:Envelope>
```

The contents of the **CanCreateDwsUrlResult** element would be as follows:

```
<Result>coho</Result>
```

4.2 CreateDws

Http: Request, POST /_vti_bin/DWS.asmx:

4.2.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateDws xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <name></name>
      <users></users>
      <title>contoso</title>
      <documents></documents>
    </CreateDws>
  </soap:Body>
</soap:Envelope>
```

4.2.2 Response

The protocol server would return the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateDwsResponse
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <CreateDwsResult>
        &lt;Results&gt;
          &lt;Url&gt;http://office/contoso&lt;/Url&gt;
          &lt;DoclibUrl&gt;Shared Documents&lt;/DoclibUrl&gt;
          &lt;ParentWeb&gt;Home&lt;/ParentWeb&gt;
          &lt;FailedUsers&gt;
            &lt;/FailedUsers&gt;
          &lt;AddUsersUrl&gt;
            http://office/contoso/_layouts/aclinv.aspx
          &lt;/AddUsersUrl&gt;
          &lt;AddUsersRole&gt;
            &lt;/AddUsersRole&gt;
          &lt;/Results&gt;
        </CreateDwsResult>
      </CreateDwsResponse>
    </soap:Body>
  </soap:Envelope>
```

The contents of the **CreateDwsResult** element would be as follows:

```
<Results>
  <Url>http://office/contoso</Url>
  <DoclibUrl>Shared Documents</DoclibUrl>
  <ParentWeb>Home</ParentWeb>
  <FailedUsers>
  </FailedUsers>
  <AddUsersUrl>
    http://office/contoso/_layouts/aclinv.aspx
  </AddUsersUrl>
  <AddUsersRole>
  </AddUsersRole>
</Results>
```

4.3 CreateFolder

Http: Request, POST /coho/contoso/_vti_bin/DWS.asmx:

4.3.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateFolder
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <url>
        Documents/coho-recipes
      </url>
    </CreateFolder>
  </soap:Body>
</soap:Envelope>
```

4.3.2 Response

The protocol server would return the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateFolderResponse
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <CreateFolderResult>
        &lt;Result/&gt;
      </CreateFolderResult>
    </CreateFolderResponse>
  </soap:Body>
</soap:Envelope>
```

The contents of the **CreateFolderResult** element would be as follows:

```
<Result/>
```

4.4 DeleteDws

In this example, the request is being posted to the "coho" site (2). This causes an error because a subsite "contoso" has been created but has not yet been deleted.

HTTP: Request, POST /coho/_vti_bin/DWS.asmx:

4.4.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteDws xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/" />
```

```
</soap:Body>
</soap:Envelope>
```

4.4.2 Response

The protocol server would return the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteDwsResponse
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <DeleteDwsResult>
        &lt;Error ID="11"&gt;WebContainsSubwebs&lt;/Error&gt;
      </DeleteDwsResult>
    </DeleteDwsResponse>
  </soap:Body>
</soap:Envelope>
```

The contents of the **DeleteDwsResult** element would be as follows:

```
<Error ID="11">WebContainsSubwebs</Error>
```

4.5 DeleteFolder

HTTP: Request, POST /coho/contoso/_vti_bin/DWS.asmx:

4.5.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteFolder
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <url>
        Documents/coho-recipes
      </url>
    </DeleteFolder>
  </soap:Body>
</soap:Envelope>
```

4.5.2 Response

The protocol server would return the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteFolderResponse
```



```

    xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
    <DeleteFolderResult>
      &lt;Result/&gt;
    </DeleteFolderResult>
  </DeleteFolderResponse>
</soap:Body>
</soap:Envelope>

```

The contents of the **DeleteFolderResult** element would be as follows:

```
<Result/>
```

4.6 FindDwsDoc

This example shows how to use **FindDwsDoc**.

HTTP: Request, POST: /_vti_bin/DWS.asmx:

4.6.1 Request

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <FindDwsDoc
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <id>
        1
      </id>
    </FindDwsDoc>
  </soap:Body>
</soap:Envelope>

```

4.6.2 Response

The protocol server would return the following response message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <FindDwsResponse
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <FindDwsResult>
        &lt;Result&gt;http://office/coho/contoso/example.doc&lt;/Result/&gt;
      </FindDwsResult>
    </FindDwsResponse>
  </soap:Body>
</soap:Envelope>

```

The contents of the **FindDwsDocResult** element would be as follows:

```
<Result>http://office/coho/contoso/example.doc</Result>
```

4.7 RemoveDwsUser

HTTP: Request, POST /coho/contoso/_vti_bin/DWS.asmx:

4.7.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <RemoveDwsUser
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <id>
        15
      </id>
    </RemoveDwsUser>
  </soap:Body>
</soap:Envelope>
```

4.7.2 Response

The protocol server would return the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <RemoveDwsUserResponse
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <RemoveDwsUserResult>
        &lt;Result/&gt;
      </RemoveDwsUserResult>
    </RemoveDwsUserResponse>
  </soap:Body>
</soap:Envelope>
```

The contents of the **RemoveDwsUserResult** element would be as follows:

```
<Result/>
```

4.8 RenameDws

HTTP, POST: /coho/contoso/_vti_bin/DWS.asmx:

4.8.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <RenameDws xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <title>
        contoso
      </title>
    </RenameDws>
  </soap:Body>
</soap:Envelope>
```

```
    </RenameDws>
  </soap:Body>
</soap:Envelope>
```

4.8.2 Response

The protocol server would return the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <RenameDwsResponse
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <RenameDwsResult>
        &lt;Result/&gt;
      </RenameDwsResult>
    </RenameDwsResponse>
  </soap:Body>
</soap:Envelope>
```

The contents of the **RenameDwsResult** element would be as follows:

```
<Result/>
```

4.9 GetDwsData

This is an example of the SOAP payload portions of a **GetDwsData** operation request and response.

HTTP: Request, POST: /_vti_bin/DWS.asmx:

4.9.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetDwsData
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/" />
  </soap:Body>
</soap:Envelope>
```

4.9.2 Response

Some repetitive content has been removed for brevity, but the protocol server would return the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
```

```

<GetDwsDataResponse
  xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
  <GetDwsDataResult>
    &lt;Results&gt;
      &lt;Title&gt;Home&lt;/Title&gt;
      &lt;LastUpdate&gt;633397396056064066&lt;/LastUpdate&gt;
      &lt;User&gt;
        &lt;ID&gt;1&lt;/ID&gt;
        &lt;Name&gt;Ekaterina&lt;/Name&gt;
        &lt;LoginName&gt;WORKGROUP\loginname&lt;/LoginName&gt;
        &lt;Email&gt;ekat@cohowinery.com&lt;/Email&gt;
        &lt;IsDomainGroup&gt;False&lt;/IsDomainGroup&gt;
        &lt;IsSiteAdmin&gt;True&lt;/IsSiteAdmin&gt;
      &lt;/User&gt;
      &lt;Members&gt;
        &lt;Member&gt;
          &lt;ID&gt;1&lt;/ID&gt;
          &lt;Name&gt;Ekaterina&lt;/Name&gt;
          &lt;LoginName&gt;WORKGROUP\loginname&lt;/LoginName&gt;
          &lt;Email&gt;ekat@cohowinery.com&lt;/Email&gt;
          &lt;IsDomainGroup&gt;False&lt;/IsDomainGroup&gt;
        &lt;/Member&gt;
        &lt;Member&gt;
          &lt;ID&gt;7&lt;/ID&gt;
          &lt;Name&gt;Designers&lt;/Name&gt;
          &lt;Email /&gt;
          &lt;LoginName /&gt;
          &lt;IsDomainGroup&gt;True&lt;/IsDomainGroup&gt;
        &lt;/Member&gt;
        &lt;Member&gt;
          &lt;ID&gt;14&lt;/ID&gt;
          &lt;Name&gt;Viewers&lt;/Name&gt;
          &lt;Email /&gt;
          &lt;LoginName /&gt;
          &lt;IsDomainGroup&gt;True&lt;/IsDomainGroup&gt;
        &lt;/Member&gt;
      &lt;/Members&gt;
      &lt;Assignees&gt;
        &lt;Member&gt;
          &lt;ID&gt;1&lt;/ID&gt;
          &lt;Name&gt;Ekaterina&lt;/Name&gt;
          &lt;LoginName&gt;WORKGROUP\loginname&lt;/LoginName&gt;
        &lt;/Member&gt;
      &lt;/Assignees&gt;
      &lt;List Name=&quot;Tasks&quot;&gt;
        &lt;ID&gt;{A35606E6-635F-4D65-82BF-B4119999E517}&lt;/ID&gt;
      &lt;/List&gt;
      &lt;List Name=&quot;Documents&quot;&gt;
        &lt;Error ID=&quot;1&quot;&gt;ServerFailure&lt;/Error&gt;
      &lt;/List&gt;
      &lt;List Name=&quot;Links&quot;&gt;
        &lt;ID&gt;{E387A317-980C-480C-BFFC-BF595142D1D4}&lt;/ID&gt;
      &lt;/List&gt;
    &lt;/Results&gt;
  </GetDwsDataResult>
</GetDwsDataResponse>
</soap:Body>
</soap:Envelope>

```

The contents of the **GetDwsDataResult** element would be as follows:

```

<Results>
  <Title>Home</Title>
  <LastUpdate>633397396056064066</LastUpdate>
  <User>
    <ID>1</ID>
    <Name>Ekaterina</Name>

```

```

    <LoginName>WORKGROUP\loginname</LoginName>
    <Email>ekat@cohowinery.com</Email>
    <IsDomainGroup>False</IsDomainGroup>
    <IsSiteAdmin>True</IsSiteAdmin>
  </User>
  <Members>
    <Member>
      <ID>1</ID>
      <Name>Ekaterina</Name>
      <LoginName>WORKGROUP\loginname</LoginName>
      <Email>ekat@cohowinery.com</Email>
      <IsDomainGroup>False</IsDomainGroup>
    </Member>
    <Member>
      <ID>7</ID>
      <Name>Designers</Name>
      <Email />
      <LoginName />
      <IsDomainGroup>True</IsDomainGroup>
    </Member>
    <Member>
      <ID>14</ID>
      <Name>Viewers</Name>
      <Email />
      <LoginName />
      <IsDomainGroup>True</IsDomainGroup>
    </Member>
  </Members>
  <Assignees>
    <Member>
      <ID>1</ID>
      <Name>Ekaterina</Name>
      <LoginName>WORKGROUP\loginname</LoginName>
    </Member>
  </Assignees>
  <List Name="Tasks">
    <ID>{A35606E6-635F-4D65-82BF-B4119999E517}</ID>
  </List>
  <List Name="Documents">
    <Error ID="1">ServerFailure</Error>
  </List>
  <List Name="Links">
    <ID>{E387A317-980C-480C-BFFC-BF595142D1D4}</ID>
  </List>
</Results>

```

4.10 GetDwsMetaData

This is an example of the SOAP payload portions of a **GetDwsMetaData** operation request and response.

HTTP: Request, POST: /_vti_bin/DWS.asmx:

4.10.1 Request

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetDwsMetaData
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws">
      <minimal>
        false
      </minimal>
    </GetDwsMetaData>
  </soap:Body>
</soap:Envelope>

```

```

    </GetDwsMetaData>
  </soap:Body>
</soap:Envelope>

```

4.10.2 Response

Some repetitive content has been removed for brevity, but the protocol server would return the following response message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetDwsMetaDataResponse
      xmlns="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <GetDwsMetaDataResult>
        <Results>
          <SubscribeUrl>http://office/_layouts/SubNew.aspx</SubscribeUrl>
          <MtgInstance />
          <SettingUrl>http://office/_layouts/settings.aspx</SettingUrl>
          <PermsUrl>http://office/_layouts/EditPrms.aspx</PermsUrl>
          <UserInfoUrl>http://office/_layouts/UserEdit.aspx</UserInfoUrl>
          <Roles>
            <Role Name="Full Control" Type="Administrator"
      Description="Has full control." />
            <Role Name="Design" Type="WebDesigner"
      Description="Can view, add, update, delete, approve, and customize." />
            <Role Name="Contribute" Type="Contributor"
      Description="Can view, add, update, and delete." />
            <Role Name="Read" Type="Reader" Description="Can view
      only." />
          </Roles>
          <Schema Name="Tasks" />
            <Field Name="Title" Type="Text"
      Required="True" />
            <Choices />
            </Field>
            <Field Name="Priority" Type="Choice"
      Required="False" />
            <Choices>
              <Choice>(1) High</Choice>
              <Choice>(2) Normal</Choice>
              <Choice>(3) Low</Choice>
            </Choices>
            </Field>
            <Field Name="Status" Type="Choice"
      Required="False" />
            <Choices>
              <Choice>Not Started</Choice>
              <Choice>In Progress</Choice>
              <Choice>Completed</Choice>
              <Choice>Deferred</Choice>
              <Choice>Waiting on someone else</Choice>
            </Choices>
            </Field>
            </Schema>
            <Schema Name="Documents" Url="Style Library" />
            <Field Name="FileLeafRef" Type="File"
      Required="True" />
            <Choices />
            </Field>
            <Field Name="Title" Type="Text"
      Required="False" />
            <Choices />

```

```

    </Field>;
  </Schema>;
  <Schema Name="Links">;
    <Field Name="URL" Type="URL" Required="True">;
      <Choices />;
    </Field>;
    <Field Name="Comments" Type="Note"
Required="False">;
      <Choices />;
    </Field>;
  </Schema>;
  <ListInfo Name="Tasks">;
    <Moderated>False</Moderated>;
    <ListPermissions>;
      <InsertListItems />;
      <EditListItems />;
      <DeleteListItems />;
      <ManageLists />;
    </ListPermissions>;
  </ListInfo>;
  <ListInfo Name="Documents">;
    <Moderated>False</Moderated>;
    <ListPermissions>;
      <InsertListItems />;
      <EditListItems />;
      <DeleteListItems />;
      <ManageLists />;
    </ListPermissions>;
  </ListInfo>;
  <ListInfo Name="Links">;
    <Moderated>False</Moderated>;
    <ListPermissions>;
      <InsertListItems />;
      <EditListItems />;
      <DeleteListItems />;
      <ManageLists />;
    </ListPermissions>;
  </ListInfo>;
  <Permissions>;
    <ManageSubwebs />;
    <ManageWeb />;
    <ManageRoles />;
    <ManageLists />;
    <InsertListItems />;
    <EditListItems />;
    <DeleteListItems />;
  </Permissions>;
  <HasUniquePerm>True</HasUniquePerm>;
  <WorkspaceType />;
  <IsADMode>False</IsADMode>;
  <DocUrl />;
  <Minimal>False</Minimal>;
  <Results>;
    <Title>Home</Title>;
    <LastUpdate>633397370817168378</LastUpdate>;
    <User>;
      <ID>1</ID>;
      <Name>Ekaterina</Name>;
      <LoginName>loginname</LoginName>;
      <Email>ekat@cohowinery.com</Email>;
      <IsDomainGroup>False</IsDomainGroup>;
      <IsSiteAdmin>True</IsSiteAdmin>;
    </User>;
  <Members>;
    <Member>;
      <ID>1</ID>;
      <Name>Ekaterina</Name>;
      <LoginName>loginname</LoginName>;
      <Email>ekat@cohowinery.com</Email>;

```

```

        <IsDomainGroup>False</IsDomainGroup>
    </Member>
    <Member>
        <ID>7</ID>
        <Name>Designers</Name>
        <Email />
        <LoginName />
        <IsDomainGroup>True</IsDomainGroup>
    </Member>
    <Assignees>
        <Member>
            <ID>1</ID>
            <Name>Ekaterina</Name>
            <LoginName>loginname</LoginName>
        </Member>
    </Assignees>
    <List Name="Tasks">
        <ID>{A35606E6-635F-4D65-82BF-B4119999E517}</ID>
    </List>
    <List Name="Documents">
        <ID>{74AC02DF-C060-4AB3-A309-CD456623B60C}</ID>
        <z:row
ows FileRef="Style Library/en-us/Core Styles/zz2 EditMode.css"
ows_FSObjType="0" ows_Created="2008-02-26T20:25:53Z"
ows_Author="1;#Ekaterina" ows_Modified="2008-02-26T20:25:53Z"
ows_Editor="1;#Ekaterina" ows_ID="64" ows_ProgID="
xmlns:z="RowsetSchema" />
        <z:row
ows FileRef="Style Library/en-us/Core Styles/PageLayouts.css"
ows_FSObjType="0" ows_Created="2008-02-26T20:25:53Z"
ows_Author="1;#Ekaterina" ows_Modified="2008-02-26T20:25:53Z"
ows_Editor="1;#Ekaterina" ows_ID="67" ows_ProgID="
xmlns:z="RowsetSchema" />
        </List>
    <List Name="Links">
        <ID>{E387A317-980C-480C-BFFC-BF595142D1D4}</ID>
    </List>
    </Results>
</Results>
</GetDwsMetaDataResult>
</GetDwsMetaDataResponse>
</soap:Body>
</soap:Envelope>

```

The contents of the **GetDwsMetaDataResult** element would be as follows:

```

<Results>
  <SubscribeUrl>http://office/_layouts/SubNew.aspx</SubscribeUrl>
  <MtgInstance />
  <SettingUrl>http://office/_layouts/settings.aspx</SettingUrl>
  <PermsUrl>http://office/_layouts/EditPrms.aspx</PermsUrl>
  <UserInfoUrl>http://office/_layouts/UserEdit.aspx</UserInfoUrl>
  <Roles>
    <Role Name="Full Control" Type="Administrator" Description="Has full control." />
    <Role Name="Design" Type="WebDesigner" Description="Can view, add, update, delete,
approve, and customize." />
    <Role Name="Contribute" Type="Contributor" Description="Can view, add, update, and
delete." />
    <Role Name="Read" Type="Reader" Description="Can view only." />
  </Roles>
  <Schema Name="Tasks">
    <Field Name="Title" Type="Text" Required="True">
      <Choices />
    </Field>
    <Field Name="Priority" Type="Choice" Required="False">
      <Choices>
        <Choice>(1) High</Choice>
      </Choices>
    </Field>
  </Schema>
</Results>

```



```

        <Choice>(2) Normal</Choice>
        <Choice>(3) Low</Choice>
    </Choices>
</Field>
<Field Name="Status" Type="Choice" Required="False">
    <Choices>
        <Choice>Not Started</Choice>
        <Choice>In Progress</Choice>
        <Choice>Completed</Choice>
        <Choice>Deferred</Choice>
        <Choice>Waiting on someone else</Choice>
    </Choices>
</Field>
</Schema>
<Schema Name="Documents" Url="Style Library">
    <Field Name="FileLeafRef" Type="File" Required="True">
        <Choices />
    </Field>
    <Field Name="Title" Type="Text" Required="False">
        <Choices />
    </Field>
</Schema>
<Schema Name="Links">
    <Field Name="URL" Type="URL" Required="True">
        <Choices />
    </Field>
    <Field Name="Comments" Type="Note" Required="False">
        <Choices />
    </Field>
</Schema>
<ListInfo Name="Tasks">
    <Moderated>False</Moderated>
    <ListPermissions>
        <InsertListItems />
        <EditListItems />
        <DeleteListItems />
        <ManageLists />
    </ListPermissions>
</ListInfo>
<ListInfo Name="Documents">
    <Moderated>False</Moderated>
    <ListPermissions>
        <InsertListItems />
        <EditListItems />
        <DeleteListItems />
        <ManageLists />
    </ListPermissions>
</ListInfo>
<ListInfo Name="Links">
    <Moderated>False</Moderated>
    <ListPermissions>
        <InsertListItems />
        <EditListItems />
        <DeleteListItems />
        <ManageLists />
    </ListPermissions>
</ListInfo>
<Permissions>
    <ManageSubwebs />
    <ManageWeb />
    <ManageRoles />
    <ManageLists />
    <InsertListItems />
    <EditListItems />
    <DeleteListItems />
</Permissions>
<HasUniquePerm>True</HasUniquePerm>
<WorkspaceType />
<IsADMode>False</IsADMode>

```

```

<DocUrl />
<Minimal>False</Minimal>
<Results>
  <Title>Home</Title>
  <LastUpdate>633397370817168378</LastUpdate>
  <User>
    <ID>1</ID>
    <Name>Ekaterina</Name>
    <LoginName>loginname</LoginName>
    <Email>ekat@cohowinery.com</Email>
    <IsDomainGroup>False</IsDomainGroup>
    <IsSiteAdmin>True</IsSiteAdmin>
  </User>
  <Members>
    <Member>
      <ID>1</ID>
      <Name>Ekaterina</Name>
      <LoginName>loginname</LoginName>
      <Email>ekat@cohowinery.com</Email>
      <IsDomainGroup>False</IsDomainGroup>
    </Member>
    <Member>
      <ID>7</ID>
      <Name>Designers</Name>
      <Email />
      <LoginName />
      <IsDomainGroup>True</IsDomainGroup>
    </Member>
  <Assignees>
    <Member>
      <ID>1</ID>
      <Name>Ekaterina</Name>
      <LoginName>loginname</LoginName>
    </Member>
  </Assignees>
  <List Name="Tasks">
    <ID>{A35606E6-635F-4D65-82BF-B4119999E517}</ID>
  </List>
  <List Name="Documents">
    <ID>{74AC02DF-C060-4AB3-A309-CD456623B60C}</ID>
    <z:row
ows FileRef="Style Library/en-us/Core Styles/zz2 EditMode.css"
ows_FSObjType="0" ows_Created="2008-02-26T20:25:53Z"
ows_Author="1;#Ekaterina" ows_Modified="2008-02-26T20:25:53Z"
ows_Editor="1;#Ekaterina" ows_ID="64" ows_ProgID="
xmlns:z="#RowsetSchema" />
    <z:row
ows FileRef="Style Library/en-us/Core Styles/PageLayouts.css"
ows_FSObjType="0" ows_Created="2008-02-26T20:25:53Z"
ows_Author="1;#Ekaterina" ows_Modified="2008-02-26T20:25:53Z"
ows_Editor="1;#Ekaterina" ows_ID="67" ows_ProgID="
xmlns:z="#RowsetSchema" />
    </List>
  <List Name="Links">
    <ID>{E387A317-980C-480C-BFFC-BF595142D1D4}</ID>
  </List>
</Results>
</Results>

```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL and schema are provided in this appendix.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/dws/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/dws/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/dws/">
      <s:element name="CreateDws">
        <s:complexType>
          <s:sequence>
            <s:element name="name" type="s:string" />
            <s:element name="users" type="s:string" />
            <s:element name="title" type="s:string" />
            <s:element name="documents" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="CreateDwsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="CreateDwsResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="DeleteDws">
        <s:complexType />
      </s:element>
      <s:element name="DeleteDwsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="DeleteDwsResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetDwsMetaData">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="document" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1"
name="id" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1"
name="minimal" type="s:boolean" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetDwsMetaDataResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="GetDwsMetaDataResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>

```

```

<s:element name="GetDwsData">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="document" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1"
        name="lastUpdate" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetDwsDataResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="GetDwsDataResult" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="UpdateDwsData">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="updates" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1"
        name="meetingInstance" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="UpdateDwsDataResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="UpdateDwsDataResult" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="RemoveDwsUser">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="id" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="RemoveDwsUserResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="RemoveDwsUserResult" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="RenameDws">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="title" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="RenameDwsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="RenameDwsResult" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="FindDwsDoc">

```

```

    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
          name="id" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="FindDwsDocResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
          name="FindDwsDocResult" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="CanCreateDwsUrl">
    <s:complexType>
      <s:sequence>
        <s:element name="url" type="s:string"
          minOccurs="0" maxOccurs="1" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="CanCreateDwsUrlResponse">
    <s:complexType>
      <s:sequence>
        <s:element name="CanCreateDwsUrlResult" type="s:string"
          minOccurs="0" maxOccurs="1" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="CreateFolder">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
          name="url" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="CreateFolderResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
          name="CreateFolderResult" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="DeleteFolder">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
          name="url" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="DeleteFolderResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
          name="DeleteFolderResult" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
</wsdl:types>
<wsdl:message name="CreateDwsSoapIn">
  <wsdl:part name="parameters" element="tns:CreateDws" />
</wsdl:message>
<wsdl:message name="CreateDwsSoapOut">

```

```

    <wsdl:part name="parameters" element="tns:CreateDwsResponse" />
</wsdl:message>
<wsdl:message name="DeleteDwsSoapIn">
  <wsdl:part name="parameters" element="tns:DeleteDws" />
</wsdl:message>
<wsdl:message name="DeleteDwsSoapOut">
  <wsdl:part name="parameters" element="tns:DeleteDwsResponse" />
</wsdl:message>
<wsdl:message name="GetDwsMetaDataSoapIn">
  <wsdl:part name="parameters" element="tns:GetDwsMetaData" />
</wsdl:message>
<wsdl:message name="GetDwsMetaDataSoapOut">
  <wsdl:part name="parameters" element="tns:GetDwsMetaDataResponse" />
</wsdl:message>
<wsdl:message name="GetDwsDataSoapIn">
  <wsdl:part name="parameters" element="tns:GetDwsData" />
</wsdl:message>
<wsdl:message name="GetDwsDataSoapOut">
  <wsdl:part name="parameters" element="tns:GetDwsDataResponse" />
</wsdl:message>
<wsdl:message name="UpdateDwsDataSoapIn">
  <wsdl:part name="parameters" element="tns:UpdateDwsData" />
</wsdl:message>
<wsdl:message name="UpdateDwsDataSoapOut">
  <wsdl:part name="parameters" element="tns:UpdateDwsDataResponse" />
</wsdl:message>
<wsdl:message name="RemoveDwsUserSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveDwsUser" />
</wsdl:message>
<wsdl:message name="RemoveDwsUserSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveDwsUserResponse" />
</wsdl:message>
<wsdl:message name="RenameDwsSoapIn">
  <wsdl:part name="parameters" element="tns:RenameDws" />
</wsdl:message>
<wsdl:message name="RenameDwsSoapOut">
  <wsdl:part name="parameters" element="tns:RenameDwsResponse" />
</wsdl:message>
<wsdl:message name="FindDwsDocSoapIn">
  <wsdl:part name="parameters" element="tns:FindDwsDoc" />
</wsdl:message>
<wsdl:message name="FindDwsDocSoapOut">
  <wsdl:part name="parameters" element="tns:FindDwsDocResponse" />
</wsdl:message>
<wsdl:message name="CanCreateDwsUrlSoapIn">
  <wsdl:part name="parameters" element="tns:CanCreateDwsUrl" />
</wsdl:message>
<wsdl:message name="CanCreateDwsUrlSoapOut">
  <wsdl:part name="parameters" element="tns:CanCreateDwsUrlResponse" />
</wsdl:message>
<wsdl:message name="CreateFolderSoapIn">
  <wsdl:part name="parameters" element="tns:CreateFolder" />
</wsdl:message>
<wsdl:message name="CreateFolderSoapOut">
  <wsdl:part name="parameters" element="tns:CreateFolderResponse" />
</wsdl:message>
<wsdl:message name="DeleteFolderSoapIn">
  <wsdl:part name="parameters" element="tns:DeleteFolder" />
</wsdl:message>
<wsdl:message name="DeleteFolderSoapOut">
  <wsdl:part name="parameters" element="tns:DeleteFolderResponse" />
</wsdl:message>
<wsdl:portType name="DwsSoap">
  <wsdl:operation name="CreateDws">
    <wsdl:input message="tns:CreateDwsSoapIn" />
    <wsdl:output message="tns:CreateDwsSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="DeleteDws">
    <wsdl:input message="tns:DeleteDwsSoapIn" />

```

```

        <wsdl:output message="tns:DeleteDwsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetDwsMetaData">
        <wsdl:input message="tns:GetDwsMetaDataSoapIn" />
        <wsdl:output message="tns:GetDwsMetaDataSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetDwsData">
        <wsdl:input message="tns:GetDwsDataSoapIn" />
        <wsdl:output message="tns:GetDwsDataSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="UpdateDwsData">
        <wsdl:input message="tns:UpdateDwsDataSoapIn" />
        <wsdl:output message="tns:UpdateDwsDataSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="RemoveDwsUser">
        <wsdl:input message="tns:RemoveDwsUserSoapIn" />
        <wsdl:output message="tns:RemoveDwsUserSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="RenameDws">
        <wsdl:input message="tns:RenameDwsSoapIn" />
        <wsdl:output message="tns:RenameDwsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="FindDwsDoc">
        <wsdl:input message="tns:FindDwsDocSoapIn" />
        <wsdl:output message="tns:FindDwsDocSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="CanCreateDwsUrl">
        <wsdl:input message="tns:CanCreateDwsUrlSoapIn" />
        <wsdl:output message="tns:CanCreateDwsUrlSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="CreateFolder">
        <wsdl:input message="tns:CreateFolderSoapIn" />
        <wsdl:output message="tns:CreateFolderSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="DeleteFolder">
        <wsdl:input message="tns:DeleteFolderSoapIn" />
        <wsdl:output message="tns:DeleteFolderSoapOut" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="DwsSoap" type="tns:DwsSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="CreateDws">
        <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/CreateDws"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteDws">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/DeleteDws"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetDwsMetaData">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/GetDwsMetaData" style="document"
/>
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>

```



```

        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetDwsData">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/GetDwsData"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateDwsData">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/UpdateDwsData" style="document"
/>
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveDwsUser">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/RemoveDwsUser" style="document"
/>
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RenameDws">
        <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/RenameDws"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="FindDwsDoc">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/FindDwsDoc" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CanCreateDwsUrl">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/CanCreateDwsUrl"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateFolder">

```

```

        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/CreateFolder" style="document"
/>
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteFolder">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/DeleteFolder" style="document"
/>
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="DwsSoap12" type="tns:DwsSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="CreateDws">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/CreateDws" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteDws">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/DeleteDws" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetDwsMetaData">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/GetDwsMetaData" style="document"
/>
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetDwsData">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/GetDwsData" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateDwsData">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/UpdateDwsData" style="document"
/>

```

```

    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RemoveDwsUser">
    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/RemoveDwsUser" style="document"
/>
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RenameDws">
    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/RenameDws" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="FindDwsDoc">
    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/FindDwsDoc" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CanCreateDwsUrl">
    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/CanCreateDwsUrl"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CreateFolder">
    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/CreateFolder" style="document"
/>
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="DeleteFolder">
    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/dws/DeleteFolder" style="document"
/>
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>

```

```
</wsdl:output>  
</wsdl:operation>  
</wsdl:binding>  
</wsdl:definitions>
```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

- Microsoft Office 2003
- The 2007 Microsoft Office system
- Microsoft Office 2010 suites
- Microsoft Office 2013
- Windows SharePoint Services 2.0
- Windows SharePoint Services 3.0
- Microsoft SharePoint Foundation 2010
- Microsoft SharePoint Foundation 2013
- Windows 8.1 Update
- Microsoft Office 2016
- Windows 10 operating system
- Microsoft SharePoint Server 2016

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.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

[<1> Section 2.2.3.4](#): The Document Workspace limit for lists in Windows SharePoint Services is 99.

[<2> Section 2.2.4.1](#): The Document Workspace limit for a members list in Windows SharePoint Services is 99.

[<3> Section 2.2.5.2](#): This role is available only in SharePoint Foundation 2013.

[<4> Section 3.1.4.1](#): The maximum length for the root site (2) URL in Windows SharePoint Services is 441 in this case.

[<5> Section 3.1.4.2.2](#): When a site with the specified name already exists on this site, Windows SharePoint Services 3.0, SharePoint Foundation 2010 and SharePoint Foundation 2013 will return error **ServerFailure** instead of **AlreadyExists**.

[<6> Section 3.1.4.3.2.1](#): For Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if **url** contains multiple folders and the first folder in the **url** doesn't exist in the site, the server will ignore the first folder in the **url** and replace it with "**Shared Documents**".

[<7> Section 3.1.4.4](#): For SharePoint Foundation 2010, the text is "404 NOT FOUND". For Windows SharePoint Services 3.0, the text is empty.

[<8> Section 3.1.4.8.2.2](#): Windows SharePoint Services 3.0, SharePoint Foundation 2010 and SharePoint Foundation 2013 will never return the "NoAccess" error code; instead, these product versions will return HTTP status code 401 with a response body containing the text "401 Unauthorized".

<9> [Section 3.1.4.8.2.2](#): SharePoint Foundation 2013 does not return this **Schema** element.

<10> [Section 3.1.4.8.2.3](#): Windows SharePoint Services 3.0, SharePoint Foundation 2010 and SharePoint Foundation 2013 will never return "NoAccess" error code and will return HTTP status code 401 with response body which contains text "401 Unauthorized".

<11> [Section 3.1.4.11](#): Microsoft Office Word 2007 uses **UpdateDwsData**.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

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

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
Z Appendix B: Product Behavior	Updated product behavior section to accurately reflect supported products.	Y	Product behavior note updated.

9 Index

A

Abstract data model
[server](#) 23
[Applicability](#) 13
[Assignees element](#) 15
[Attribute groups](#) 22
[Attributes](#) 22

C

[CanCreateDwsUrl example](#) 53
[Capability negotiation](#) 13
[Change tracking](#) 79
[Complex types](#) 19
 [MemberData](#) 20
 [Schema](#) 20
[CreateDws example](#) 53
[CreateFolder example](#) 55

D

Data model - abstract
[server](#) 23
[DeleteDws example](#) 55
[DeleteFolder example](#) 56

E

Elements
 [Assignees](#) 15
 [Error](#) 15
 [ID](#) 17
 [List](#) 17
 [Member](#) 18
 [Roles](#) 19
[Error element](#) 15
Events
 [local - server](#) 52
 [timer - server](#) 52

Examples

[CanCreateDwsUrl](#) 53
[CreateDws](#) 53
[CreateFolder](#) 55
[DeleteDws](#) 55
[DeleteFolder](#) 56
[FindDwsDoc](#) 57
[GetDwsData](#) 59
[GetDwsMetaData](#) 61
[overview](#) 53
[RemoveDwsUser](#) 58
[RenameDws](#) 58

F

[Fields - vendor-extensible](#) 13
[FindDwsDoc example](#) 57
[Full WSDL](#) 68

G

[GetDwsData example](#) 59
[GetDwsMetaData example](#) 61
[Glossary](#) 7
[Groups](#) 22

I

[ID element](#) 17
[Implementer - security considerations](#) 67
[Index of security parameters](#) 67
[Informative references](#) 10
Initialization
 [server](#) 23
[Introduction](#) 7

L

[List element](#) 17
[ListType simple type](#) 21
Local events
 [server](#) 52

M

[Member element](#) 18
[MemberData complex type](#) 20
Message processing
 [server](#) 23
Messages
 [Assignees element](#) 15
 [attribute groups](#) 22
 [attributes](#) 22
 [complex types](#) 19
 [elements](#) 15
 [enumerated](#) 14
 [Error element](#) 15
 [groups](#) 22
 [ID element](#) 17
 [List element](#) 17
 [ListType simple type](#) 21
 [Member element](#) 18
 [MemberData complex type](#) 20
 [namespaces](#) 14
 [Roles element](#) 19
 [RoleType simple type](#) 21
 [Schema complex type](#) 20
 [simple types](#) 21
 [syntax](#) 14
 [transport](#) 14

N

[Namespaces](#) 14
[Normative references](#) 9

O

Operations
 [CanCreateDwsUrl](#) 24
 [CreateDws](#) 26
 [CreateFolder](#) 30

[DeleteDws](#) 32
[DeleteFolder](#) 34
[FindDwsDoc](#) 36
[GetDwsData](#) 38
[GetDwsMetaData](#) 41
[RemoveDwsUser](#) 46
[RenameDws](#) 48
[UpdateDwsData](#) 50
[Overview \(synopsis\)](#) 10

P

[Parameters - security index](#) 67
[Preconditions](#) 12
[Prerequisites](#) 12
[Product behavior](#) 77
Protocol Details
[overview](#) 23

R

[References](#) 9
[informative](#) 10
[normative](#) 9
[Relationship to other protocols](#) 12
[RemoveDwsUser example](#) 58
[RenameDws example](#) 58
[Roles element](#) 19
[RoleType simple type](#) 21

S

[Schema complex type](#) 20
Security
[implementer considerations](#) 67
[parameter index](#) 67
Sequencing rules
[server](#) 23
Server
[abstract data model](#) 23
[CanCreateDwsUrl operation](#) 24
[CreateDws operation](#) 26
[CreateFolder operation](#) 30
[DeleteDws operation](#) 32
[DeleteFolder operation](#) 34
[FindDwsDoc operation](#) 36
[GetDwsData operation](#) 38
[GetDwsMetaData operation](#) 41
[initialization](#) 23
[local events](#) 52
[message processing](#) 23
[RemoveDwsUser operation](#) 46
[RenameDws operation](#) 48
[sequencing rules](#) 23
[timer events](#) 52
[timers](#) 23
[UpdateDwsData operation](#) 50
[Simple types](#) 21
[ListType](#) 21
[RoleType](#) 21
[Standards assignments](#) 13
Syntax
[messages - overview](#) 14

T

Timer events
[server](#) 52
Timers
[server](#) 23
[Tracking changes](#) 79
[Transport](#) 14
Types
[complex](#) 19
[simple](#) 21

V

[Vendor-extensible fields](#) 13
[Versioning](#) 13

W

[WSDL](#) 68