[MS-DWSS]:
Document Workspace Web Service Protocol Specification

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's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
07/13/2009	1.02	Major	Revised and edited the technical content
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1   Introduction

This document specifies the Document Workspace Web Service protocol. This 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 (2)** intended for easier creation and maintenance than an entire shared Web site.

1.1   Glossary

The following terms are defined in [MS-GLOS]:

Active Directory
Active Directory Domain Services (AD DS)
authenticated users
globally unique identifier (GUID)
Hypertext Transfer Protocol (HTTP)
XML

The following terms are defined in [MS-OFCGLOS]:

absolute URL
ActiveX Data Objects (ADO)
alert
CAML
check out
current user
document library
Document Workspace site
folder
friendly name
group
item
link
list
list item
member
Members group
page
personal view
role
site
site-relative URL
SOAP action
SOAP body
SOAP fault
SOAP operation
URI (Uniform Resource Identifier)
URL (Uniform Resource Locator)
user identifier
Web Part
WSDL (Web Services Description Language)
XML namespace
XML Schema

The following terms are specific to this document:

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

1.2 References

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. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-ALERTSS] Microsoft Corporation, "Alerts Service Protocol Specification", June 2008.

[MS-OSALER] Microsoft Corporation, "Alerts Interoperability Protocol Specification", June 2008.

[MS-PRSTFR] Microsoft Corporation, "ADO XML Persistence Format Protocol Specification", June 2008.

[MS-WSSCAML] Microsoft Corporation, "Collaborative Application Markup Language (CAML) Structure Specification", June 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.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.ietf.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] World Wide Web Consortium, "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation 8 December 2009, http://www.w3.org/TR/REC-xml-names/

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

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary", June 2008.

1.3 Protocol Overview (Synopsis)

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.
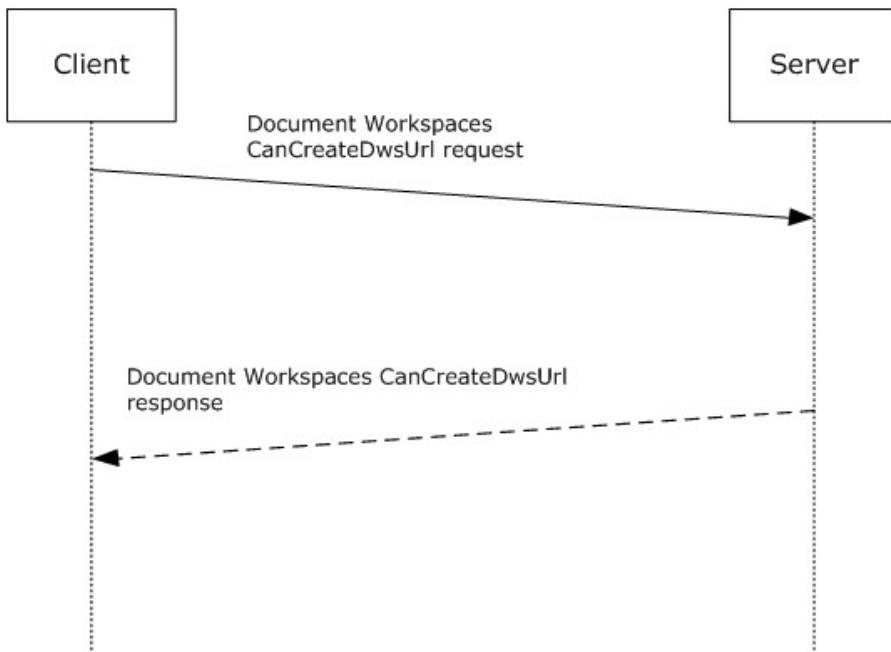
Figure 1: Communication during the CanCreateDwsUrl operation

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.
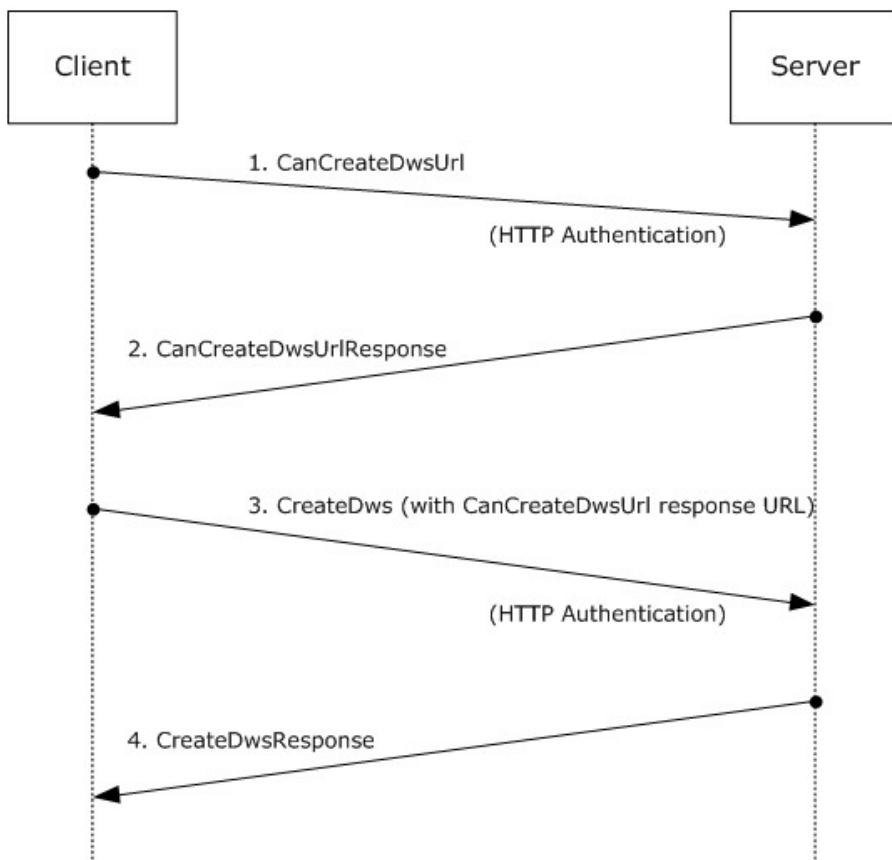
Figure 2: Creating a Document Workspace

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 (2) 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.
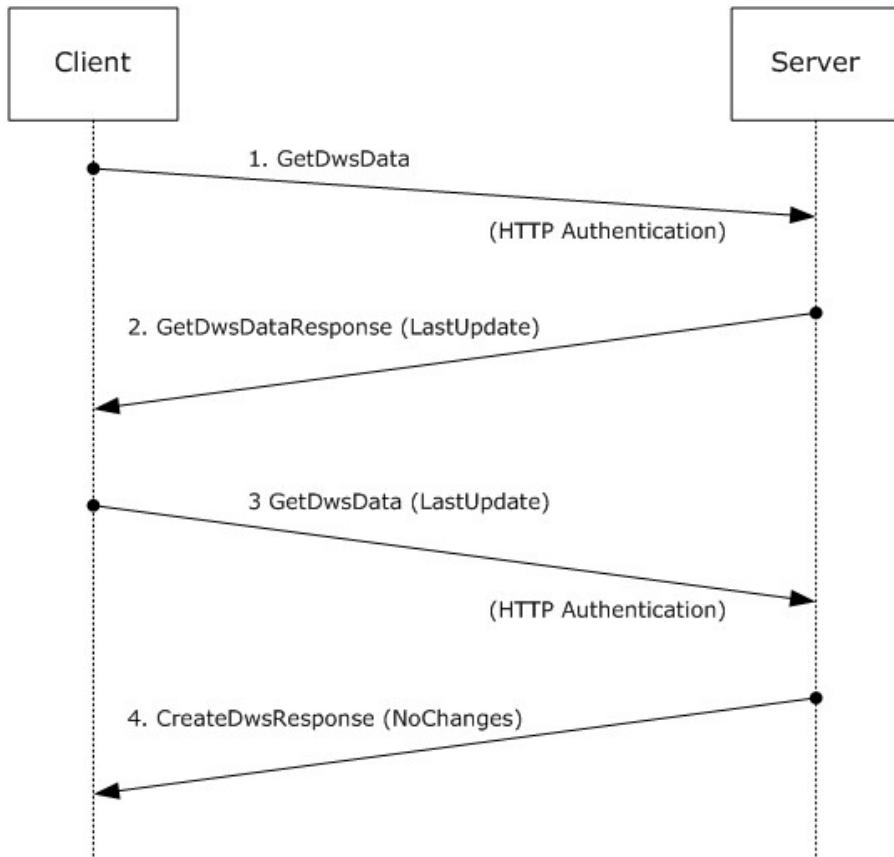
Figure 3: Client-server interaction when LastUpdate is used

1.4 Relationship to Other Protocols

This protocol uses Simple Object Access Protocol (SOAP) over **HTTP** or HTTPS as shown in the following layering diagram:
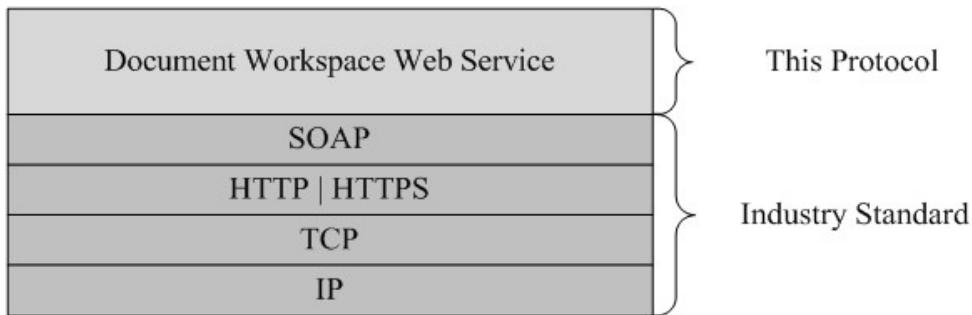
Figure 4: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a site (2) 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 (2) 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 (2) 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 areas:

Supported Transports: This protocol uses multiple transports with SOAP as specified in **Transport** (section 2.1).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2   Messages

In the following sections, the schema definition might be less restrictive than 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 specifies additional restrictions 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, present, and not null.

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

2.2   Common Message Syntax

This section contains an **XML Schema** as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and WSDL (Web Services Description Language) as defined 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 XML namespace prefix for each XML namespace that is used, the choice of any particular XML 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]

2.2.2   Messages

None.

2.2.3   Elements

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

Element	Description
Assignees	Returns a **list (1)** of assigned **members (1)** 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.
Member	Returns data about a user or **group (2)**.
Roles	Contains role information about a list or document.

2.2.3.1 Assignees

This element holds the list (1) 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"/>
    </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 appropriate 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="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 appropriate 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. Error codes that can be returned are also in the 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	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 (1) structure to return a GUID that identifies a document for which the protocol server returns error information. The **ID** element can have a **DefaultUrl** that contains a URL specifying 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 (1). 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"/>
              <xs:sequence>
                <xs:element ref="z:row"/>
              </xs:sequence>
            </xs:choice>
          </xs:sequence>
        </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 (1) 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 (1) 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 **row** elements with information about the documents.
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 that can be returned are described in the following table.

Error code	Description
ServerFailure	Protocol server encountered an unspecified error.
NoAccess	**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 lists (1) exceeds an implementation-specific size<1>

z:row: 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.

2.2.3.5 Member

This element contains information about a member (1) 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 (2).

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 (2). When its value is "True", it's a group. When its value is "False", it's a user.

2.2.3.6 Row

This element returns metadata about list items. See [MS-PRSTFR].

2.2.3.7 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">
              <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 that can be returned 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 **MemberData** XML schema complex type definitions are defined by this specification. XML schema complex type definitions that are specific to a particular operation are described with the operation.

2.2.4.1 MemberData

This type returns information about a user or group (2) or an error condition that occurred while accessing the user or group data. Which possible 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.

ErrorCode	Description
NoAccess	The current user does not have access to this information.
TooManyItems	The number of members (1) exceeds an implementation-specific size.<2>
ServerFailure	The protocol server encountered an unspecified error.

2.2.5 Simple Types

The following table summarizes the set of common XML Schema simple type definitions that are 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 (1) being returned.
RoleType	A string that indicates the user role.

2.2.5.1 ListType

This simple type lists the possible 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 possible 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: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 (2) by using Self-Service Site Creation, the user becomes the site owner and a member (1) of the administrator class for the new site (2). This does not affect membership for any other site (2).

Role	Rights
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 always contains at least one member (1). Role members always have or can grant themselves access to any item on the site (2).

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 be less restrictive than 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 specifies additional restrictions 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, present, and not null.

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   Document Workspace Web Service Protocol 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 (2) 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 (1) 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

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 (2).
CreateDws	Creates a Document Workspace.
CreateFolder	Creates a folder in a Document Workspace or another folder.
DeleteDws	Deletes a Document Workspace.
DeleteFolder	Deletes a folder in a Document Workspace.

Operation	Description
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 (1) 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 described in section 2.2.3.2, with the identifier set to 2 and a restricted string of **"Failed"** when the length of the site (2) URL is greater than an implementation-dependent length.<3>

3.1.4.1.1 Messages

The following WSDL message definitions are specific to this 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 XML Schema element definitions are specific to this operation.

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 (2). The URL SHOULD be based on the URL provided in the *url* input parameter. If the *url* input parameter is empty, the protocol server MAY 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:

```
<Result>(site relative unique url)</Result>
```

The XML Schema for the **XML** element is defined as follows:

```
<s:element name="Result" type="s:string"/>
```

Error: See section 2.2.3.2 for specification.

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="CreateDwsSoapIn" />
    <wsdl:output message="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

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

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 can be empty. If this is non-empty and another site (2) with the same name already exists on the site (2) 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 can be empty.

When the string is not empty, the XML MUST conform to the following XSD:

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

Name: Friendly name for a user.

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

title: The title for the workspace. This string can 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 (2) metadata. This string can be empty. If it is not empty, it MUST conform to the following XSD:

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

```
          <s:attribute name="Name" type="s:string" />
          <s:attribute name="ID" type="s:string" />
        </s:element>
      </s:sequence>
    </s:complexType>
```

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

ID: A string containing an integer identifier 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:sequence>
    <s:element name="User" minOccurs="0">
      <s:attribute name="Email" type="s:string" minOccurs="1" />
    </s:element>
  </s:sequence>
</s:complexType>
<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="User" minOccurs="0" maxOccurs="unbounded"/>
      <s:element name="AddUsersUrl" type="s:string"/>
      <s:element name="AddUsersRole" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

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

DoclibUrl: Site-relative URL for the shared documents list (1) 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 **CreateDws Users** field 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>
    <s:sequence>
      <s:element name="User" minOccurs="0">
        <s:attribute name="Email" type="s:string" minOccurs="1" />
      </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. The value MUST be the string "Microsoft.SharePoint.SPRoleDefinition"**.**

Error: See section 2.2.3.2 for specification. Possible values for this operation are listed in the following table.

Error	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	13	A site (2) with the specified name already exists on this site (2).

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="CreateFolderSoapIn" />
      <wsdl:output message="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

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

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.

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 name="Error" type="Error"/>
  </s:choice>
</s:complexType>
```

If the element is an **Error** element, it MUST contain an error code as specified in section 2.2.3.2.

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="DeleteDwsSoapIn" />
    <wsdl:output message="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 MUST return a **Result** element as specified under 3.1.4.4.2.2.

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

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

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.

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

3.1.4.5 DeleteFolder

This operation deletes a folder from a document library on the site (2).

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="DeleteFolderSoapIn" />
    <wsdl:output message="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 **DeleteDwsResponse**.

3.1.4.5.1 Messages

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:

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

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

3.1.4.5.2 Elements

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

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="FindDwsDocSoapIn" />
    <wsdl:output message="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

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

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: A GUID encoded in a string that specifies the appropriate document URL.

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

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

If the element is an **Error**, it MUST contain an error code as specified in section 2.2.3.2.

If the element is a **Result**, the content MUST be an absolute URL referring to the requested document.

3.1.4.7 GetDwsData

This operation returns general information about the **Document Workspace site**, as well as its members (1), 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="GetDwsDataSoapIn" />
    <wsdl:output message="GetDwsDataSoapOut" />
</wsdl:operation>
```

If the Document Workspace at the specified URL does not have a valid parent site (2), the protocol server MUST return an **Error** element with the **DocumentNotFound** code.

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 appropriate information for the Document Workspace and document.

3.1.4.7.1 Messages

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

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

document: A site (2)-based URL of a document in the document library in the Document Workspace. This argument provides the context for the documents list (1) in the reply.

ID: A GUID encoded in a string that specifies the appropriate document URL.

Minimal: A Boolean value indicating whether to return list information in the **GetDwsDataResponse** element. If true, the protocol server MUST NOT include the **Assignees** element or its child elements in the **GetDwsDataResponse**. If true, the protocol server MUST NOT include the **Lists** element or its child elements in the **GetDwsDataResponse**.

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="s:string" 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>
    <s:choice>
      <s:element type="Error"/>
      <s:element name="Results">
        <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" type="xs:boolean"/>
                <xs:element name="IsSiteAdmin" type="xs:boolean"/>
              </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>
      </s:element>
    </s:choice>
  </s:complexType>
```

Title: The title of the workspace.

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

User: The member (1) 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 the appropriate 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.

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. 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.

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="GetDwsMetaDataSoapIn" />
    <wsdl:output message="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 appropriate schema definition section.

3.1.4.8.1 Messages

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

There are two formal elements for the **GetDwsMetaData** operation: **GetDwsMetaData** for the input parameters and **GetDwsMetaDataResponse** for the results. The results are encoded in stand–alone XML.

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 (1) or document to describe in the response.

ID: A GUID that a protocol client can use instead of the document URL.

Minimal: A Boolean value that specifies whether to return information. If false, the protocol server MUST return the following elements, and MUST NOT return them otherwise:

From the GetDwsMetaData **Results** element:

- SubscribeUrl

- Schema (Tasks)

- Schema (Documents)

- Schema (Links)

- ListInfo (Tasks)

- ListInfo (Documents)

- ListInfo (Links)

From the GetDwsData **Results** element:

- Assignees

- List (Tasks)

- List (Documents)

- List (Links)

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="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
```

GetDwsMetaDataResult: This element contains a string that is stand–alone XML encoded either as an **Error** element specified in 2.2.3.2 or a **Result** element defined as follows:

```
  <s:complexType>
    <s:choice>
      <s:element name="Error" type="Error"/>
      <s:element name="Result" type="s:string"/>
    </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	Site (2) or document could not be found.
NoAccess	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.
ServerFailure	Protocol server encountered an unspecified error during operation processing.

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

```
  <xs: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:sequence>
      <xs:element name="PermsUrl" type="xs:string"/>
      <xs:element name="UserInfoUrl"/>
      <xs:element ref="tns:Roles"/>
      <xs:sequence minOccurs="0">
        <xs:element ref="tns:Schema"/>
        <xs:element ref="tns:Schema"/>
        <xs:element ref="tns:Schema"/>
        <xs:element ref="tns:ListInfo"/>
        <xs:element ref="tns:ListInfo"/>
        <xs:element ref="tns:ListInfo"/>
      </xs:sequence>
      <xs:element name="Permissions">
         <xs:complexType>
            <xs:choice>
             <xs:element ref="tns:Error"/>
               <xs: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"/>
               </xs:sequence>
             </xs: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="GetDwsDataResult"/>
   </xs:complexType>
  </xs:element>
```

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 (1) subscriptions.

MtgInstance: If **document** element specifies a meeting item, this element MUST represent a string that contains the meeting information. 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 (1) of users to be modified.

Roles: Specifies the roles that apply to the workspace. This element MUST conform to the element specification in 2.2.3.7.

Schema: Contains the schema information for the workspace *Ta*sks list. The element MUST conform to the **Schema** element in the **xml** element specified in [MS-PRSTFR].

Schema: Contains the schema information for the workspace list. The element MUST conform to the **Schema** element in the **xml** element specified in [MS-PRSTFR].

Schema: This element contains the schema information for the workspace Links list. This element MUST conform to the **Schema** element in the **xml** element specified in [MS-PRSTFR].

ListInfo: This element contains list information for the workspace Tasks list. This element MUST conform to the **ListInfo** element specified in 3.1.4.8.2.3.

ListInfo: This element contains list information for the workspace Documents list. This element MUST conform to the **ListInfo** element specified in 3.1.4.8.2.3.

ListInfo: This element contains list information for the workspace Links list. This element MUST conform to the **ListInfo** element specified in 3.1.4.8.2.3.

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 can be present are described in the WSDL specified in **Appendix A** (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. If the workspace 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, that is, a mode in which new site (2) members are created in **Active Directory Domain Services (AD DS)**; otherwise set to FALSE.

DocUrl: This MUST 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.

GetDwsDataResult: This element is identical to the **GetDwsDataResult** element specified in section 3.1.4.7.2.2, with the exception that it MUST contain a **Result** element. 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 (2) in a Document Workspace. This element is defined as follows:

```
<xs:element name="ListInfo">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="tns:Error"/>
      <xs:sequence>
        <xs:element name="Moderated" type="xs:boolean"/>
        <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"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </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	Current user does not have sufficient permissions to access list permission information.

Error code	Description
	The **AccessUrl** attribute MUST NOT be present in this element.
ServerFailure	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 (1). 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 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="RemoveDwsUserSoapIn" />
    <wsdl:output message="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

3.1.4.9.1.1   RemoveDwsUserSoapIn

This message requests that a user be deleted 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

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 2147483647.

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: This element is returned when any error occurs in processing.

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="RenameDwsSoapIn" />
    <wsdl:output message="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 **AccessUrl** 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

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

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: This element is returned when an error occurs in processing.

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.<4> This operation is defined as follows:

```
<wsdl:operation name="UpdateDwsData">
    <wsdl:input message="UpdateDwsDataSoapIn" />
    <wsdl:output message="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 [MS-WSSCAML].

3.1.4.11.1 Messages

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

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 as specified in [MS-WSSCAML] that specifies how to update the workspace information.

meetingInstance: If **document** element specifies a meeting item, this element MUST represent a string that contains the meeting information. Otherwise, the 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 MAY contain a **ServerFailure, Conflict or ItemNotFound** error, as specified in section 2.2.3.2.

If it does not contain an **Error** element, it MUST contain a **Result** element as specified in [MS-WSSCAML].

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Document Workspace Web Service Protocol Client Details

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:

```
<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=&quot;11&quot;&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>
```

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>
```

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>
        &lt;Results&gt;
          &lt;SubscribeUrl&gt;http://office/_layouts/SubNew.aspx&lt;/SubscribeUrl&gt;
          &lt;MtgInstance /&gt;
          &lt;SettingUrl&gt;http://office/_layouts/settings.aspx&lt;/SettingUrl&gt;
          &lt;PermsUrl&gt;http://office/_layouts/EditPrms.aspx&lt;/PermsUrl&gt;
          &lt;UserInfoUrl&gt;http://office/_layouts/UserEdit.aspx&lt;/UserInfoUrl&gt;
          &lt;Roles&gt;
            &lt;Role Name=&quot;Full Control&quot; Type=&quot;Administrator&quot;
Description=&quot;Has full control.&quot; /&gt;
            &lt;Role Name=&quot;Design&quot; Type=&quot;WebDesigner&quot;
Description=&quot;Can view, add, update, delete, approve, and customize.&quot; /&gt;
            &lt;Role Name=&quot;Contribute&quot; Type=&quot;Contributor&quot;
Description=&quot;Can view, add, update, and delete.&quot; /&gt;
            &lt;Role Name=&quot;Read&quot; Type=&quot;Reader&quot; Description=&quot;Can view
only.&quot; /&gt;
          &lt;/Roles&gt;
          &lt;Schema Name=&quot;Tasks&quot;&gt;
            &lt;Field Name=&quot;Title&quot; Type=&quot;Text&quot;
Required=&quot;True&quot;&gt;
              &lt;Choices /&gt;
            &lt;/Field&gt;
            &lt;Field Name=&quot;Priority&quot; Type=&quot;Choice&quot;
Required=&quot;False&quot;&gt;
              &lt;Choices&gt;
                &lt;Choice&gt;(1) High&lt;/Choice&gt;
                &lt;Choice&gt;(2) Normal&lt;/Choice&gt;
                &lt;Choice&gt;(3) Low&lt;/Choice&gt;
              &lt;/Choices&gt;
            &lt;/Field&gt;
            &lt;Field Name=&quot;Status&quot; Type=&quot;Choice&quot;
Required=&quot;False&quot;&gt;
              &lt;Choices&gt;
                &lt;Choice&gt;Not Started&lt;/Choice&gt;
                &lt;Choice&gt;In Progress&lt;/Choice&gt;
                &lt;Choice&gt;Completed&lt;/Choice&gt;
                &lt;Choice&gt;Deferred&lt;/Choice&gt;
                &lt;Choice&gt;Waiting on someone else&lt;/Choice&gt;
              &lt;/Choices&gt;
            &lt;/Field&gt;
          &lt;/Schema&gt;
          &lt;Schema Name=&quot;Documents&quot; Url=&quot;Style Library&quot;&gt;
            &lt;Field Name=&quot;FileLeafRef&quot; Type=&quot;File&quot;
Required=&quot;True&quot;&gt;
              &lt;Choices /&gt;
            &lt;/Field&gt;
```

```
                &lt;Field Name=&quot;Title&quot; Type=&quot;Text&quot;
Required=&quot;False&quot;&gt;
                    &lt;Choices /&gt;
                &lt;/Field&gt;
            &lt;/Schema&gt;
            &lt;Schema Name=&quot;Links&quot;&gt;
                &lt;Field Name=&quot;URL&quot; Type=&quot;URL&quot; Required=&quot;True&quot;&gt;
                    &lt;Choices /&gt;
                &lt;/Field&gt;
                &lt;Field Name=&quot;Comments&quot; Type=&quot;Note&quot;
Required=&quot;False&quot;&gt;
                    &lt;Choices /&gt;
                &lt;/Field&gt;
            &lt;/Schema&gt;
            &lt;ListInfo Name=&quot;Tasks&quot;&gt;
                &lt;Moderated&gt;False&lt;/Moderated&gt;
                &lt;ListPermissions&gt;
                    &lt;InsertListItems /&gt;
                    &lt;EditListItems /&gt;
                    &lt;DeleteListItems /&gt;
                    &lt;ManageLists /&gt;
                &lt;/ListPermissions&gt;
            &lt;/ListInfo&gt;
            &lt;ListInfo Name=&quot;Documents&quot;&gt;
                &lt;Moderated&gt;False&lt;/Moderated&gt;
                &lt;ListPermissions&gt;
                    &lt;InsertListItems /&gt;
                    &lt;EditListItems /&gt;
                    &lt;DeleteListItems /&gt;
                    &lt;ManageLists /&gt;
                &lt;/ListPermissions&gt;
            &lt;/ListInfo&gt;
            &lt;ListInfo Name=&quot;Links&quot;&gt;
                &lt;Moderated&gt;False&lt;/Moderated&gt;
                &lt;ListPermissions&gt;
                    &lt;InsertListItems /&gt;
                    &lt;EditListItems /&gt;
                    &lt;DeleteListItems /&gt;
                    &lt;ManageLists /&gt;
                &lt;/ListPermissions&gt;
            &lt;/ListInfo&gt;
            &lt;Permissions&gt;
                &lt;ManageSubwebs /&gt;
                &lt;ManageWeb /&gt;
                &lt;ManageRoles /&gt;
                &lt;ManageLists /&gt;
                &lt;InsertListItems /&gt;
                &lt;EditListItems /&gt;
                &lt;DeleteListItems /&gt;
            &lt;/Permissions&gt;
            &lt;HasUniquePerm&gt;True&lt;/HasUniquePerm&gt;
            &lt;WorkspaceType /&gt;
            &lt;IsADMode&gt;False&lt;/IsADMode&gt;
            &lt;DocUrl /&gt;
            &lt;Minimal&gt;False&lt;/Minimal&gt;
            &lt;Results&gt;
                &lt;Title&gt;Home&lt;/Title&gt;
                &lt;LastUpdate&gt;633397370817168378&lt;/LastUpdate&gt;
                &lt;User&gt;
```

```
              &lt;ID&gt;1&lt;/ID&gt;
              &lt;Name&gt;Ekaterina&lt;/Name&gt;
              &lt;LoginName&gt;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;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;Assignees&gt;
              &lt;Member&gt;
                &lt;ID&gt;1&lt;/ID&gt;
                &lt;Name&gt;Ekaterina&lt;/Name&gt;
                &lt;LoginName&gt;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;ID&gt;{74AC02DF-C060-4AB3-A309-CD456623B60C}&lt;/ID&gt;
              &lt;z:row
          ows_FileRef=&quot;Style Library/en-us/Core Styles/zz2_EditMode.css&quot;
          ows_FSObjType=&quot;0&quot; ows_Created=&quot;2008-02-26T20:25:53Z&quot;
          ows_Author=&quot;1;#Ekaterina&quot; ows_Modified=&quot;2008-02-26T20:25:53Z&quot;
          ows_Editor=&quot;1;#Ekaterina&quot; ows_ID=&quot;64&quot; ows_ProgID=&quot;
          xmlns:z=&quot;#RowsetSchema&quot; /&gt;
              &lt;z:row
          ows_FileRef=&quot;Style Library/en-us/Core Styles/PageLayouts.css&quot;
          ows_FSObjType=&quot;0&quot; ows_Created=&quot;2008-02-26T20:25:53Z&quot;
          ows_Author=&quot;1;#Ekaterina&quot; ows_Modified=&quot;2008-02-26T20:25:53Z&quot;
          ows_Editor=&quot;1;#Ekaterina&quot; ows_ID=&quot;67&quot; ows_ProgID=&quot;
          xmlns:z=&quot;#RowsetSchema&quot; /&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;
        &lt;/Results&gt;
      </GetDwsMetaDataResult>
    </GetDwsMetaDataResponse>
  </soap:Body>
</soap:Envelope>
```

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

```xml
<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>
```

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 operation is provided in this section:

```
<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: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: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

- Windows® SharePoint® Services 2.0

- Windows® SharePoint® Services 3.0

- Microsoft® SharePoint® Foundation 2010

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 3.1.4.1: The maximum length for a URL in Windows SharePoint Services is 441 in this case.

<4> Section 3.1.4.11: Office Word 2007 uses **UpdateDwsData**.

8 Change Tracking

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

9 Index

Release: Sunday, December 19, 2010