[MS-WSSCAP]:
Windows SharePoint Services Collaborative Application Protocol Specification

---

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
04/25/2008	0.2	Editorial	Revised and edited the technical content
06/27/2008	1.0	Major	Revised and edited the technical content
10/06/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
07/13/2009	1.03	Major	Changes made for template compliance
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	Editorial	Revised and edited the technical content
02/19/2010	2.0	Major	Updated and revised the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	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.
03/18/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	3.0	Major	Significantly changed the technical content.
04/11/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	3.0.1	Editorial	Changed language and formatting in the technical content.

Date	Revision History	Revision Class	Comments
09/12/2012	3.0.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	4.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Windows SharePoint Services Collaborative Application Protocol enables a protocol client to retrieve and manipulate various types of content on a protocol server.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

GUID
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
interface
little-endian
XML

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

absolute URL
base view identifier
Boolean
CAML
Collaborative Application Markup Language (CAML)
content type
content type identifier
current user
data source
default form
default list view
default view
discussion bookmark
display form
display name
document
document library
document template
edit form
Entity
field
file
FilterDescriptor
Finder
folder
front-end web server
full-text index catalog
home page
HTTP GET
HTTP POST
HTTP referer
iCalendar

Information Rights Management (IRM)
item
list
list item
list item identifier
list template
list view
List View Web Part
login name
meeting instance
Meeting Workspace site
new form
page
parent site
query
restriction
root folder
server-relative URL
site
site template
site-relative URL
survey list
Uniform Resource Locator (URL)
usage data
user information list
user-agent string
vCard
version
view
web discussion
web discussion comment
Web Part zone
Web Parts Page
Web Services Description Language (WSDL)
workflow
XML namespace
XML schema
XML schema definition (XSD)

The following terms are specific to this document:

basic page: A Web Parts Page that contains only one Web Part zone and, by default, a Content Editor Web Part.

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

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

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.

[MC-FPSEWM] Microsoft Corporation, "FrontPage Server Extensions: Website Management Protocol Specification".

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

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

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

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

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

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

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

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

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

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

[XMLSCHEMA1] Thompson, H.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".

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

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.ietf.org/rfc/rfc2818.txt

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

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

[SOAP1.2/2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part2-20030624

1.3 Overview

This protocol describes the communication between a protocol client and a **front-end Web server** to retrieve and manipulate **lists (1)**, **list views**, and **Web discussions**. By using this protocol, a protocol client can manage schemas and properties of lists, create and modify fields in lists, and render list views. It can also retrieve the schemas of **sites (2)** on the protocol server.

Each method of this protocol is an **HTTP POST** or GET request that accepts a set of parameters and returns a set of values as an HTML response. Any method that updates data on the protocol server needs to be sent by using the HTTP POST method. Calls to the protocol server to retrieve data can be sent by using the **HTTP GET** method. The case-sensitive *Cmd* parameter describes what operation the protocol server is to perform, in addition to the meanings of the other parameters and return values. The protocol server treats the value of the *Cmd* parameter in a case-insensitive manner unless otherwise described. All communication is transported over **HTTP** or secure HTTP (**HTTPS**). The parameters to the method are sent as POST arguments in the body of the POST or as **query** parameters that are part of the **URL**. The *CS* parameter can be used to describe the character set of the form body.

1.3.1 List Management

This protocol enables a protocol client to create lists (1) and **list items**, modify **XML schemas** and properties of lists (1), and delete lists (1) and list items.

1.3.2 List View Management

Data in a list (1) is displayed through a set of list views that are stored on the protocol server. The list view describes the display layout and structure of the data in the list (1). This protocol enables the protocol client to create new list views and manipulate existing views.

1.3.3 Web Discussions

To support discussion and reviews of **document** content, the protocol server enables the protocol client to add comments to documents. The comments are stored on the protocol server and this protocol enables protocol clients to create, enumerate, modify, or delete them.

1.4 Relationship to Other Protocols

This protocol uses HTTP, as described in [RFC2616], or HTTPS, as described in [RFC2818].

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

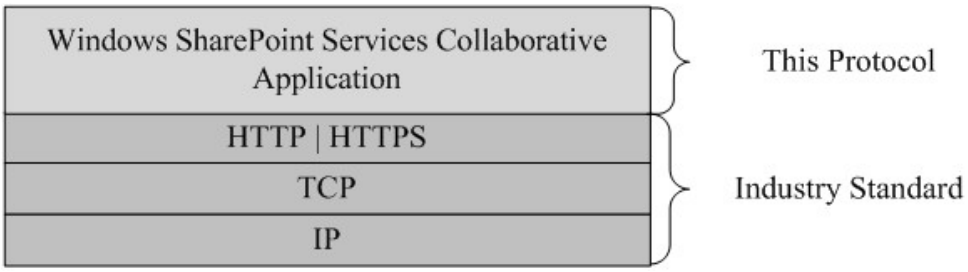


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol assumes that authentication has been performed by the underlying protocols.

1.6 Applicability Statement

This protocol is a precursor to the SOAP protocol, as described in [SOAP1.1], [SOAP1.2/1] and [SOAP1.2/2], and can be used in similar situations.

1.7 Versioning and Capability Negotiation

Protocol Versions: The protocol client and protocol server do not perform version negotiations except where batch **XML** is used, as described in section 2.2.3.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol uses HTTP version 1.1, as specified in [RFC2616], as the transport for the HTTP GET and HTTP POST methods.

2.2 Common Message Syntax

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

2.2.1 Namespaces

This specification defines and references various **XML namespaces** 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.

The following table shows the XML namespaces that are used by this protocol and the prefix for each namespace.

Prefix	Namespace URI	Reference
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

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

Element	Description
Batch	Contains one or more **Method** elements used in a batched mode.
Result	Used to send a response to the protocol client from the **DisplayPost** operation.
WECFileList	Returned by the protocol server as the result of a batch operation.

2.2.3.1 Batch

The **Batch** element contains one or more **Method** elements and is used in a batched mode operation to perform multiple operations in a single client-server transaction.

```
<xs:element name="Batch">
  <xs:complexType>
    <xs:attribute name="OnError" type="OnErrorEnum" use="optional" />
    <xs:attribute name="Version" type="xs:string" use="required" >
```

```
          <xs:restriction base="xs:string">
             <xs:pattern value="[0-9]\.[0-9]\.[0-9]\.[0-9][0-9][0-9][0-9]"/>
          </xs:restriction>
        </xs:attribute>

        <xs:sequence>
  <xs:element name="Method">
      <xs:complexType>
        <xs:attribute name="ID"
                      type="xs:string"
                      use="required" />
        <xs:sequence>
          <xs:element minOccurs="0" name="SetList">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="Scope"
                                type="xs:string"
                                use="required" />
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>

          <xs:element maxOccurs="unbounded" name="SetVar">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="Name"
                                type="xs:string"
                                use="required" />
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

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

OnError: Optional. Specifies how to handle errors that occur during a batched mode operation. If the attribute is not present, then the server MUST default to the "Return" value of the **OnErrorEnum** type (section 2.2.5.1).

Version: A string that specifies the version number of the service on the protocol server. A version number consists of four integers in the format N.N.N.NNNN, which represent the major, minor, phase, and incremental versions of the service. The protocol server SHOULD<1> fails the request if the version is older than a certain value.

Method: A string that specifies the operation to be performed on the server. It contains elements to handle all of the parameters to be passed to the method. For more information about the parameters that are used with a specific method, refer to the section of this document that specifies that method.

ID: A string that specifies a unique identifier that the server MUST use as the **ID** attribute of the **Result** element (section 2.2.3.2) for the XML response that is sent to the protocol client.

SetList: A string that specifies the GUID for the list (1). Any method that runs on a specific list (1) MUST specify the list GUID within the **SetList** element. When using **DisplayPost** to display information about a list (1) or **document library** on the site (2), the **SetList** element MUST be set to the string "Lists". For more information, see section 3.1.4.8.

Scope: A string that MUST be set to "Request".

SetVar: The other parameters for the method MUST be passed with **SetVar** elements. The **Name** attribute of the **SetVar** element MUST be set to the name of the parameter. The string value within the **SetVar** element MUST be set to the parameter value. **SetList** is the only way to specify the GUID for the list (1). GUID for the list (1) MUST be ignored by **SetVar.**

2.2.3.2 Result

The **Result** element is used to send a response to the protocol client from **DisplayPost** (section 3.1.4.8). A **Result** element MUST be present for every method that is requested by the protocol client.

```
<xs:element name="Result">
  <xs:complexType mixed="true">
    <xs:sequence minOccurs="0">
      <xs:element name="ErrorText" type="xs:string" />
    </xs:sequence>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:any minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:string" use="required" />
    <xs:attribute name="Code" type="xs:int" use="required" />
</xs:complexType>
  </xs:element>
```

If an error occurs, the contents of the **Result** element MUST be an **ErrorText** element that contains the error message. For information about the contents of the **Result** element for a successful operation, see the section for that method.

ID: A string that MUST be equal to the **ID** attribute of each **Method** element that is sent to the protocol server.

Code: An integer that specifies the return value for the method call.

2.2.3.3 WECFileList

The **WECFileList** element is returned by the protocol server as the result of a batch operation. One **WECFileList** element MUST be returned for the batch operation if documents or **folders** are modified on the protocol server by the operation. It contains the list (1) of documents and folders that were modified on the protocol server by the operation. The information returned includes the **server-relative URLs** and metadata of the modified documents and folders. The names and metadata of the documents are returned as a **document_list**, as specified in [MC-FPSEWM] section 3.1.5.3.1, or in **DOCUMENT-LIST-RETURN-TYPE** format, as specified in [MC-FPSEWM] section 2.2.2.2.13. The data about the folders are returned as a **VECTOR-URL-DIRECTORY**, as specified [MC-FPSEWM] section 2.2.2.2.1 and [MC-FPSEWM] section 2.2.2.2.6.

```
<xs:element name="WECFileList">
   </xs:element>
```

2.2.4 Complex Types

This specification does not define any common XML schema complex type definitions.

2.2.5 Simple Types

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

Simple type	Description
OnErrorEnum	Specifies how to handle errors that occur during a batch operation.
Method XML Fragment	

2.2.5.1 OnErrorEnum

The **OnErrorEnum** simple type specifies how to handle errors that occur during a batch operation.

```
<xs:simpleType name="OnErrorEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Return" />
    <xs:enumeration value="Continue" />
  </xs:restriction>
</xs:simpleType>
```

Return: Stops processing any more methods after the first error occurs.

Continue: Continues processing subsequent methods after an error occurs.

2.2.5.2 Method Xml Fragment

The following **XML schema definition (XSD)** specifies the **Method Xml** fragment:

```
<xs:element name="Method" type="ViewMethodDefinition" minOccurs="0" maxOccurs="1" />
<xs:complexType name="ViewMethodDefinition">
    <xs:sequence>
      <xs:element name="Filter" type="ViewFilterDefinition" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="Name" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="ViewFilterDefinition">
    <xs:attribute name="Name" type="xs:string" />
    <xs:attribute name="Value" type="xs:string" />
</xs:complexType>
```

Filter: Specifies operations applied to filter results from the **data source (1)** for the list (1).

Method: Part of a view query to filter results from the data source (1) for the list (1).

Name: Name of the method or filter.

Value: Value of the filter operation.

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.

2.2.9 Common Data Structures

None.

2.2.9.1 Binary Structure

None.

2.2.9.1.1 Usage Data Binary Field Structure

A structure that contains usage data for a site (2). The structure starts with a header that describes the data contained by the field, followed by 5 types of usage data blocks, as shown by the following table:

Usage Data Header (100 bytes)
Page Data (Variable)
User Data (Variable)
Operating System Data (Variable)
Browser Data (Variable)
Referrer Data (Variable)
Reserved (290 bytes)

Usage Data Header (100 bytes): Defined in section 2.2.9.1.1.1.

Page Data (Variable): A series of Usage Records that specify the pages that have been requested from a site (2). Each Usage Record contains the **site-relative URL** of the **page** that was requested followed by the number of times that it has been requested in each of the last 31 days (for daily **usage data**), or 31 months (for monthly usage data). There MUST NOT be any Usage Records for pages that have not been requested.

User Data (Variable): A series of Usage Records that specify the users that have requested content from a site. Each Usage Records contains the **login name** of a user that requested content followed by the number of requests for each of the last 31 days (for daily usage data), or 31 months (for monthly usage data). There MUST NOT be any Usage Records for users that have not requested content.

Operating System Data (Variable): A series of Usage Records that specify the operating systems that have requested content from a site, as provided in the **user-agent string**. Each Usage Record contains the name of the operating system followed by the number of requests for each of the last 31 days (for daily usage data), or 31 months (for monthly usage data). There MUST NOT be any Usage Records for operating systems that have not requested content.

Browser Data (Variable): A series of Usage Records that specify the browsers that have requested content from a site, as provided in the user-agent string. Each Usage Record contains the name of the browser followed by the number of requests for each of the last 31 days (for daily usage data), or 31 months (for monthly usage data). There MUST NOT be any Usage Records for browsers that have not requested content.

Referrer Data (Variable): A series of Usage Records that specify the **HTTP referer** in requests to content from a site. Each Usage Record contains the address of the HTTP referer, followed by the number of times that the address has been present in requests for each of the last 31 days (for daily usage data), or 31 months (for monthly usage data). There MUST NOT be any Usage Records for referrers that did not link to content in the site (2).

Reserved (190 bytes): Reserved. MUST be ignored by reader.

2.2.9.1.1.1 Usage Data Header Structure

The Usage Data Header describes the information contained by the Usage Data Binary Field structure.

byte1	byte2	byte3	byte4
Size			
Update Counter			
Page Data Offset			
User Data Offset			
Operating System Data Offset			
Browser Data Offset			
Referrer Data Offset			
Reserved 1			
…			
Page Data Count			
User Data Count			
Operating System Data Count			
Browser Data Count			

byte1	byte2	byte3	byte4
Referrer Data Count			
Reserved 2			
…			
Last Accessed Day		Rollover Day	Reserved 3
…			
…			
…			

Size (4 bytes): An unsigned integer that specifies the number of bytes contained by the structure.

Update Counter (4 bytes): An unsigned integer describing the number of times that the structure has been stored.

Page Data Offset (4 bytes): An unsigned integer that counts the number of bytes from the start of the structure to the start of the Page Data.

User Data Offset (4 bytes): An unsigned integer that counts the number of bytes from the start of the structure to the start of the User Data.

Operating System Data Offset (4 bytes): An unsigned integer that counts the number of bytes from the start of the structure to the start of the Operating System Data.

Browser Data Offset (4 bytes): An unsigned integer that counts the number of bytes from the start of the structure to the start of the Browser Data.

Referrer Data Offset (4 bytes): An unsigned integer that counts the number of bytes from the start of the structure to the start of the Browser Data.

Reserved 1 (8 bytes): MUST be ignored by reader.

Page Data Count (4 bytes): An unsigned integer that counts the number of entries of Page Data.

User Data Count (4 bytes): An unsigned integer that counts the number of entries of User Data.

Operating System Data Count (4 bytes): An unsigned integer that counts the number of entries of Operating System Data.

Browser Data Count (4 bytes): An unsigned integer that counts the number of entries of Browser Data.

Referrer Data Count (4 bytes): An unsigned integer that counts the number of entries of Referrer Data.

Reserved 2 (8 bytes): MUST be ignored by reader.

Last Accessed Day (2 bytes): An unsigned integer that contains the number of days since 1/1/1899 to the day that the structure was last stored.

Rollover Day (1 byte): An unsigned integer that specifies the rollover day of usage data from daily data into monthly data. The value MUST be between 1 and 27 (inclusive).

Reserved 3 (33 bytes): MUST be ignored by reader.

2.2.9.1.1.2 Usage Record Structure

Each of the usage data blocks consists of a series of Usage Records. The first Usage Record in each usage data block contains summary information for the usage data block. Individual usage entries then follow, each in its own Usage Record.

The Usage Record Structure consists of a Description field and a Data field.

Record Description (variable)	Record Data (variable)

Record Description (variable): A NULL terminated UTF8 encoded string. It MUST be NULL if this is the first Usage Record. For any other records, it contains the string representation of the usage data being recorded.

Record Data (variable): The usage data for the record is organized as shown in the following table:

byte1	byte2								byte3	byte4
Bytes	R	R	R	R	R	R	B	A	Last Accessed	
Hit Vector										
Total										
Hit Values (variable)										

Bytes (1 byte): An unsigned integer specifying the number of bytes contained by the Record Data structure.

A (1 bit): MUST be set to 1 if the size of values in the Hit Values field is 2 bytes per value. MUST be set to 0 for all other cases.

B (1 bit): MUST be set to 1 if the size of values in the Hit Values field is 4 bytes per value. MUST be set to 0 for all other cases.

R (1 bit): MUST be set to 0.

Last Accessed (2 bytes): An unsigned integer that contains the number of days since 1/1/1899 to the day that the record was last stored.

Hit Vector (4 bytes): A 32-bit value that specifies the number of values in the Hit Values field. The high bit corresponds to the value for the Last Accessed field. The following bits to the previous 31 days (for daily usage data) or previous 31 months (for monthly usage data).

Total (4 bytes): An unsigned integer that contains the sum of the values in the Hit Values field.

Hit Values (variable): A series of 32 unsigned integers that specify a count per day (for daily usage data) or per month (for monthly usage data) for the usage data being described by this record. The size of each value MUST be 1 byte if the A and B flags are set to 0. The size MUST be 2 bytes if the A flag is set to 1 and the B flag is set to 0. The size MUST be 4 bytes if the A flag is set to 0 and the B flag is set to 1. The number of values in this field MUST be equal to the number of bits set to 1 in the Hit Vector field.

3 Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is 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 WSSCAP 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 specification does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this specification.

3.1.1.1 List

A list (1) is a container for list items. A list (1) MUST specify a base type. The set of base types is specified in [MS-WSSFO2] section 2.2.3.11.

A list (1) with a base type of "1" MUST have one **file** associated with every list item that is stored in the list (1). This type of list (1) is referred to as a document library.

A list (1) MUST be created from a **ListDefinition**, as specified in [MS-WSSTS] section 2.1.3.2. The schema for the **ListDefinition** type is specified in [MS-WSSCAML] section 2.3.2.6. In addition, each list (1) has a template associated with it. The template specifies the schema for the list (1). For information about the XML schema of a **list template**, see **ListTemplateDefinition** type in [MS-WSSCAML] section 2.3.2.13.

A list (1) MUST have one or more **content types**. The content type specifies the schema for the list items.

A list (1) MUST have zero or more **views**.

A list (1) has various settings that are determined by list flags. For a complete set of common flags, see [MS-WSSFO2] section 2.2.2.5.

3.1.1.2 View

A view is a visualization of a list (1), and is also referred to as a list view. A view MUST have zero or more settings flags specified. These flags are specified in [MS-WSSFO2] section 2.2.2.12. The view is specified using **Collaborative Application Markup Language (CAML)**, as specified in the **ViewDefinition** type of [MS-WSSCAML] section 2.3.2.17.

3.1.1.3 List Item

A list item is a collection of related properties that represents a logical object. A list item MUST have only one parent list (1).

A list item MUST have an item index property, which specifies the unique index of the **item** in the list. A list item can be represented only one time in its parent list (1).

A list item MUST always have one specified content type. A list item MUST have one property that specifies the **content type identifier** of the list item.

If a list item is stored in a document library, the list item MUST have one corresponding file. This type of list item is known as a document. The list item has a version property that indicates how many times the item has been modified. The protocol server updates the item only if the protocol client has the same version of the item as the protocol server.

3.1.1.4 Web Discussions

This protocol enables a user to comment on and discuss documents that are stored on the protocol server. Each comment has an identifier and is associated with a document. The comments are stored separately from the document on the protocol server. Other information that is stored on the protocol server includes the author of the comment, the date when the comment was last modified, and a client-specified **discussion bookmark** that enables the protocol client to place the comment in the right position in the document. In the case of threaded discussions, comments have a parent comment associated with them through the parent comment identifier, which is stored in the comment. The protocol client can use this information to construct the thread hierarchy of comments and replies to comments.

3.1.2 Timers

None.

3.1.3 Initialization

This protocol operates against a Web site that is identified by a URL that is known by protocol clients. The protocol server endpoint is formed by appending "/_vti_bin/owssvr.dll" to the URL of the Web site, for example http://www.example.com/Repository/_vti_bin/owssvr.dll.

3.1.4 Message Processing Events and Sequencing Rules

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

Message	Description
CItreq	MAY<2> add, edit, or delete Web discussions associated with a page or document.
Delete	Deletes a list item from a list (1) or document library.
DeleteField	Deletes a **field (2)** from a list (1).
DeleteList	Deletes a list.
DeleteView	Deletes a view of a list.
DialogView	Obtains and renders an HTML view of the document libraries within a site, a specific document library, or a folder within a document library.
Display	Obtains a set of list items from a list (1).
DisplayPost	Runs methods, renders Collaborative Application Markup Language (CAML), and displays information about lists or list views.
ExportList	Exports the XML schema of a list in **CAML** format.
GetProjSchema	Obtains XML schemas from a site.

Message	Description
GetUsageBlob	Obtains the usage data information for a site.
HitCounter	Increments the hit count on a page that contains a hit counter, and obtains an image of the hit counter.
ModListSettings	Modifies the settings of a list (1).
MtgKeep	Clears the orphaned state of a **meeting instance** that is not scheduled anymore in the calendar where it was created.
MtgMove	Deletes an orphaned meeting instance, or moves the **Meeting Workspace site** content associated with an orphaned meeting instance to another meeting instance.
NewField	Adds a field (2) to a list (1).
NewList	Creates a list (1).
NewView	Creates a view of a list (1) with a more complex set of parameters than *NewViewPage*
NewViewPage	Creates a view of a list (1).
NewWebPage	Creates a **Web Parts Page** or a **basic page** in a document library.
RenderView	Obtains the HTML that renders a view and view definition for a **List View Web Part**, as specified in [MS-WSSCAML] section 2.3.2.17.
ReorderFields	Changes the order in which fields appear in a list (1).
SiteProvision	Applies a predefined **site template** or custom site template to an existing site and optionally adds the default set of lists specified in the site template or custom site template.
UpdateProject	Updates the **display name** or description of a site.
UpdateView	Updates the properties of a view.

The protocol client MUST send an **X-Vermeer-Content-Type** HTTP header, as specified in [RFC2616], with the same value as the standard HTTP **Content-Type** header to safeguard against one-click attacks (section 5.1). The protocol server MUST use this header to determine the content type of the request. If this HTTP header is not present, the protocol server MUST fail the request.

The protocol client MUST also include the case-sensitive string "FrontPage" in its **User-Agent** HTTP header, as specified in [RFC2616].

Except as indicated in specific methods, protocol server responses MUST have the HTTP **Content-Type** "application/x-vermeer-rpc".

3.1.4.1 Cltreq

The **Cltreq** method performs all Web discussion operations such as adding, editing, or deleting discussions associated with a page or document that is stored on the protocol server. Protocol servers SHOULD<3> support the **Cltreq** method.

The protocol server responses for the **Cltreq** method are binary in nature. The binary blob MUST be in **little-endian** format. The protocol server MUST specify the HTTP **Content-Type** header as "application/binary". The HTTP **Content-Length** header MUST specify the length of the **Cltreq** response in bytes.

The **CItreq** method MUST NOT be used in conjunction with the **DisplayPost** (section 3.1.4.8) method.

Cmd: A string that specifies the method name. If the **CItreq** method is used, the *Cmd* parameter MUST be present and have the value "CItreq".

UL: This parameter MUST be present and MUST have the value "1".

STRMVER: Specifies the version number of the binary data stream that is returned to the protocol client by the protocol server. This parameter MUST have the value "4".

ACT: Specifies the Web discussion action to perform. This parameter MUST be present. The following table contains the values that can be specified.

Value	Action	Description
1	ENUMTHREADSFROMURL	Return all **Web discussion comments** that are associated with the specified URL.
2	ADDCOMMENT	Add a Web discussion comment.
3	EDITCOMMENT	Edit a Web discussion comment.
4	REMOVECOMMENT	Delete a Web discussion comment.
5	GETCAPABILITY	Determine the Web discussion capabilities of the protocol server.
6	CLOSECOMMENT	Close a Web discussion comment.
7	CLOSETHREAD	Close a Web discussion comment and its child Web discussion comments.
8	ACTIVATECOMMENT	Activate the Web discussion comment.

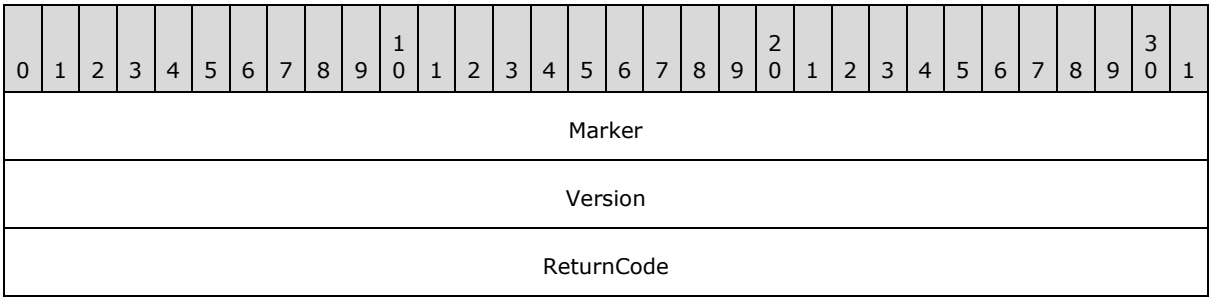
URL: Specifies the URL of the document with which the Web discussion comment is associated.

ID (optional): Specifies the identifier of the Web discussion comment for which the Web discussion action is to be performed. This parameter MUST be present for the **REMOVECOMMENT** Web discussion action. This parameter MUST be an integer. For all other Web discussion actions, the *ID* parameter MUST be ignored.

The *UL*, *STRMVER*, *ACT* and *URL* parameters MAY<4> be specified in the HTTP query string for all Web discussion actions.

3.1.4.1.1 Common Response Header

All **CItreq** responses are prefixed with a 12-byte header whose structure is as follows:



Marker (4 bytes): Reserved. The value MUST be 0xebb0ebb0.

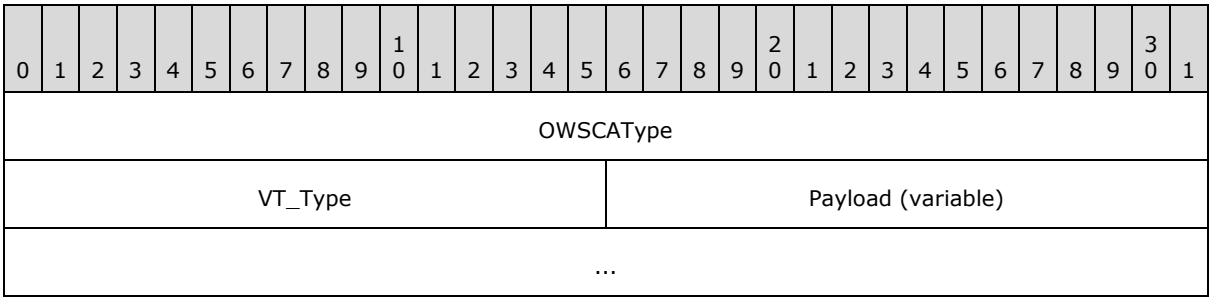
Version (4 bytes): Application-specific. MAY<5> contain the version of the protocol server.

ReturnCode (4 bytes): The return code of the request. Zero indicates success; a negative value indicates failure.

In failure conditions, the **Common Response** header SHOULD<6> be the only data that is returned by the protocol server.

3.1.4.1.2 OWSCA Structure

The **OWSCA** structure is a variable-length structure that is used to pass information between the protocol client and the protocol server. It has the following structure**:**



OWSCAType (4 bytes): An unsigned integer value specifying the type of the **OWSCA** structure. The OWSCAType value MUST be one of the following values:

Name	Value	Associated VT_Type	Meaning
OWSCA_ID	0	3	The identifier of the Web discussion comment.
OWSCA_PARENTID	1	3	The identifier of the parent Web discussion comment of the Web discussion comment.
OWSCA_STATUS	6	2	The visibility status of the Web discussion comment.
OWSCA_COMMENTID	9	8	An application-specific string, or tag, that is associated with the Web discussion comment.
OWSCA_BOOKMARK	10	8	A discussion bookmark for the Web discussion comment in the document.
OWSCA_AUTHOR	11	8	The author of the Web discussion comment.
OWSCA_LOGINNAME	12	8	The login name of the author of the Web discussion comment.
OWSCA_SUBJECT	13	8	The subject of the Web discussion comment.
OWSCA_TIMESTAMP	14	7	The creation date and time of the Web discussion comment.
OWSCA_COMMENT	15	8	The comment for the Web discussion comment.

VT_Type (2 bytes): An unsigned integer value specifying the type of data in the **OWSCA** structure. The value MUST be the "VT_Type" value that is associated with the **OWSCAType** in the preceding table, and MUST be one of the following values:

VT_Type	Payload
2	A 16-bit integer serialized in little-endian format.
3	A 32-bit integer serialized in little-endian format.
7	A date and time value, serialized as a double.
8	A NULL terminated UTF-16 string, prefixed by its length in bytes. The first 4 bytes of the string stores an integer that is the length, in bytes, of the following string. If the string is not NULL terminated, the last byte of the string MUST be ignored by the protocol server.

Payload[0…n] (variable): The value of the property, whose type is specified by **VT_TYPE** and whose format is specified in the preceding table.

The **OWSCA_STATUS OWSCAType** enables the protocol server to specify whether a protocol client shows or hides a Web discussion comment. The **Payload** values for **OWSCA_STATUS** MUST be one of the following values:

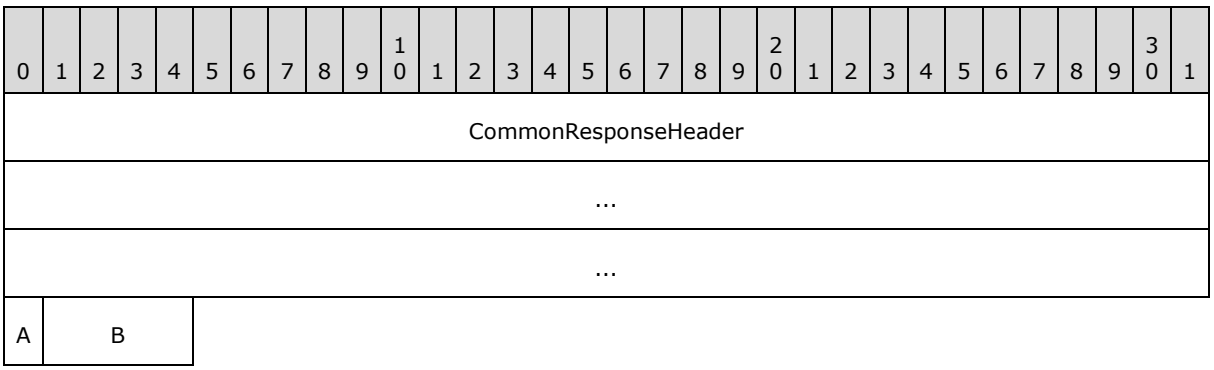
Value	Meaning
0	Default value. The Web discussion comment and its replies are suggested to be shown.
2	The Web discussion comment is suggested to be hidden.
4	The Web discussion comment and its replies are suggested to be hidden.
8	The Web discussion comment was hidden previously, but is now suggested to be shown.

3.1.4.1.3 Actions

3.1.4.1.3.1 ENUMTHREADSFROMURL

The **ENUMTHREADSFROMURL** Web discussion action enables a protocol client to enumerate the Web discussion comments for a list (1) or document.

The response varies in length depending on the number of Web discussion comments that are associated with the URL. The structure of the response is as follows:



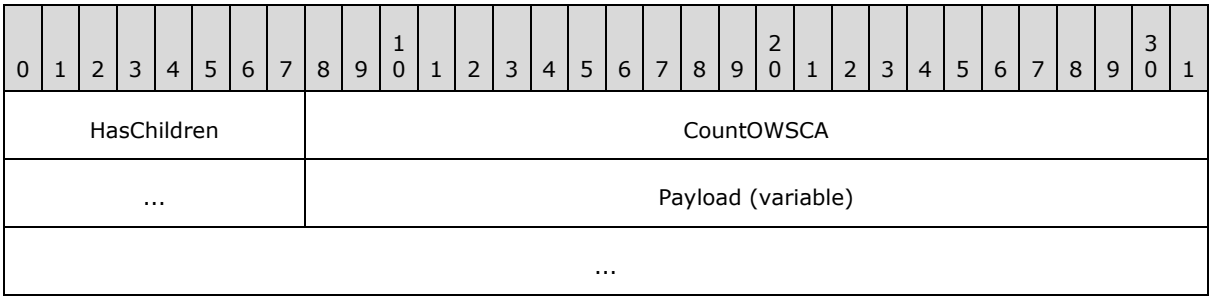
Common Response Header (12 bytes): This is the **Common Response** header, as specified in section 3.1.4.1.1.

A - Info (1 bit): This value MUST be zero or 2. If the value is zero, there are no Web discussion comments for the document and the response MUST contain no further data. If the value is2, the document has one or more Web discussion comments and the response contains more data.

B - TLComments (4 bits): Indicates the number of top-level Web discussion comments for the document.

3.1.4.1.3.1.1 Web Discussion Comment Info

The Web discussion comments that are associated with the document occur sequentially after the TLComments . Each Web discussion comment has the following form.



HasChildren (1 byte): This value MUST be 1 or 3. If the value is 3, the Web discussion comment has child comments. Otherwise, the value MUST be 1.

CountOWSCA (4 bytes): The number of OWSCA data structures that follow for the Web discussion comment. This value MUST be 10.

Payload (variable): Following this are ten OWSCA data structures that correspond to the Web discussion comment and MUST occur in the following order, by type:

1. OWSCA_ID
2. OWSCA_PARENTID
3. OWSCA_STATUS
4. OWSCA_COMMENTID
5. OWSCA_BOOKMARK
6. OWSCA_AUTHOR
7. OWSCA_LOGINNAME
8. OWSCA_SUBJECT
9. OWSCA_TIMESTAMP
10. OWSCA_COMMENT

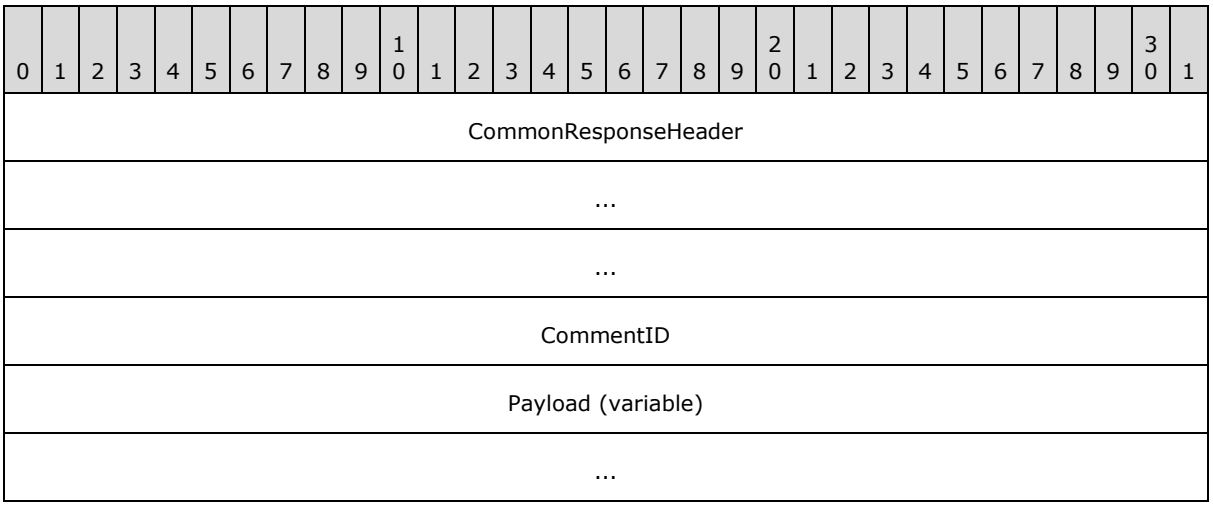
If the Web discussion comment has replies, then the next 32-bit integer MUST contain the number of replies for the Web discussion comment. The response then MUST contain the Web discussion comment information for the first reply, second reply, and so forth, in sequence, recursively.

3.1.4.1.3.2 ADDCOMMENT

The **ADDCOMMENT** Web discussion action enables the protocol client to add a Web discussion comment to a document. When **ADDCOMMENT** is used, the body of the HTTP POST MUST have the following format:

1. A 32-bit integer indicating the number of **OWSCA** data structures.
2. A list (1) of serialized **OWSCA** data structures of the indicated length. This list (1) MAY<7> contain any of the following **OWSCA** data structures, not in any particular order:
   - An **OWSCA_PARENTID** whose value is the identifier of the parent Web discussion comment. If the value is "0" (zero), the new Web discussion comment is a top-level Web discussion comment with no parent.
   - An **OWSCA_AUTHOR** whose value is the author of the new Web discussion comment.
   - An **OWSCA_SUBJECT** whose value is the subject of the new Web discussion comment.
   - An **OWSCA_COMMENT** whose value is the comment in the new Web discussion comment.
   - An **OWSCA_COMMENTID** whose value is an application-specific tag for the Web discussion comment.
   - An **OWSCA_STATUS** of the Web discussion comment as specified in section 3.1.4.1.2.
   - An **OWSCA_BOOKMARK** whose value is a discussion bookmark for the Web discussion comment in the document. If the **OWSCA_BOOKMARK** is not specified, the comment is not associated with a specific part of the document.

The response contains the standard 12-byte **Common Response** header, as specified in section 3.1.4.1.1. If no errors occurred when adding the Web discussion comment, the response has the following format.



CommonResponseHeader (12 bytes): This is the **Common Response** header, as specified in section 3.1.4.1.1.

CommentID (4 bytes): The identifier of the new Web discussion comment.

Payload (variable): Following **CommentID** is a UTF-16 string, prefixed by its 4-byte length in bytes, that stores the name of the author of the new Web discussion comment. Following that is an 8-byte, double-precision, floating point number that stores the creation date and time for the Web discussion comment.

3.1.4.1.3.3 EDITCOMMENT

The **EDITCOMMENT** Web discussion action enables a protocol client to change a Web discussion comment for a document. When **EDITCOMMENT** is used, the HTTP POST MUST have the following format:

1. A 32-bit integer indicating the count of **OWSCA** data structures to follow.
2. A list (1) of **OWSCA** data structures of the indicated length. This list (1) MUST contain an **OWSCA_ID** with the identifier of the Web discussion comment. In addition, the list (1) of **OWSCA** data structures can be any of the following **OWSCA** data structures, not in any particular order:
   - An **OWSCA_SUBJECT** whose value is the subject of the Web discussion comment.
   - An **OWSCA_STATUS** of the Web discussion comment as specified in section 3.1.4.1.2.
   - An **OWSCA_COMMENT** whose value is the comment in the Web discussion comment.

The protocol server response contains the 12-byte **Common Response** header, as specified in section 3.1.4.1.1.

3.1.4.1.3.4 REMOVECOMMENT

The **REMOVECOMMENT** Web discussion action enables a protocol client to delete a Web discussion comment for a document. When **REMOVECOMMENT** is used, the identifier of the Web discussion comment to be removed MUST be the *ID* parameter of the common set in the URL of the HTTP POST.

The protocol server response contains the **Common Response** header, as specified in section 3.1.4.1.1.

3.1.4.1.3.5 GETCAPABILITY

The **GETCAPABILITY** Web discussion action enables a protocol client to determine the Web discussion capabilities of the protocol server. **GETCAPABILITY** is the only Web discussion action that does not require the protocol client to authenticate; anonymous HTTP POST operations that use **GETCAPABILITY** requests are valid.

The protocol server capabilities are a 32-bit, unsigned integer bitmask of the following values:

Value	Supports Action
0x00000001	ENUMTHREADSFROMURL
0x00000002	ADDCOMMENT
0x00000004	EDITCOMMENT
0x00000008	DELETECOMMENT
0x00000040	Reserved. Bit value MUST be "1".

Value	Supports Action
0x00000080	CLOSECOMMENT

The protocol server MUST return "0x000000CF", indicating support for the Web discussion actions **EDITCOMMENT**, **DELETECOMMENT**, **ADDCOMMENT**, **CLOSECOMMENT**, and **ENUMTHREADSFROMURL**.

3.1.4.1.3.6 CLOSECOMMENT

The **CLOSECOMMENT** Web discussion action enables a protocol client to close a Web discussion comment for a document. When **CLOSECOMMENT** is used, the HTTP POST MUST have the following format:

1. A 32-bit integer indicating the count of OWSCA data structures to follow.
2. A list (1) of serialized **OWSCA** data structures of the indicated length. This list (1) MUST contain an **OWSCA_ID** with the identifier of the Web discussion comment.

The protocol server response contains the **Common Response** header, as specified in section 3.1.4.1.1.

3.1.4.1.3.7 CLOSETHREAD

The **CLOSETHREAD** Web discussion action enables a protocol client to close a Web discussion comment, and all of the replies to that comment, for a document. When **CLOSETHREAD** is used, the body of the HTTP POST MUST have the following format:

1. A 32-bit integer indicating the count of OWSCA data structures to follow.
2. A list (1) of serialized **OWSCA** data structures of the indicated length. This list (1) MUST contain an **OWSCA_ID** with the identifier of the Web discussion comment.

The protocol server response contains the **Common Response** header, as specified in section 3.1.4.1.1.

3.1.4.1.3.8 ACTIVATECOMMENT

If a Web discussion comment is closed, the **ACTIVATECOMMENT** Web discussion action enables a protocol client to reopen the Web discussion comment. When **ACTIVATETHREAD** is used, the HTTP POST MUST have the common set of query parameters. In addition, the body of the HTTP POST MUST have the following format:

1. A 32-bit integer indicating the count of OWSCA data structures to follow.
2. A list (1) of serialized **OWSCA** data structures of the indicated length. This list (1) MUST contain an **OWSCA_ID** with the identifier of the Web discussion comment.

3.1.4.1.4 Return Values

The **CItreq** method MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
200	Success. The protocol server response MUST contain a **Common Response** header, as specified in section 3.1.4.1.1. If the request was successful, the protocol server response MUST contain the appropriate Web discussion action response body. If it failed, the **Common Response** header MUST contain a negative integer and no response body.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Page not found.

3.1.4.2 Delete

The **Delete** method is called to delete a list item from a list (1) or document library. The following are the parameters used with the **Delete** method:

Cmd: A string that specifies the method name. If the **Delete** method is used, the *Cmd* parameter MUST be present and have the value "Delete".

List: The identifier of the list (1) that contains the list item to be deleted. This MUST be a **GUID** and it MUST be present.

ID: The identifier of the list item to be deleted. If the list (1) is a document library, the parameter is ignored. The *ID* parameter MUST be present.

NextUsing: A string that specifies the URL to which the protocol client is to be redirected after the method is finished. The URL MUST be either a server-relative URL or an **absolute URL** on the same server. If this method is called through the DisplayPost method, this parameter MUST be ignored.

owsfileref: This specifies the server-relative URL of the file or folder to be deleted. If the list (1) is a document library, this parameter MUST be present. The parameter MUST be ignored if the containing list (1) is not a document library.

owshiddenversion: Specifies the version number of the list item to be deleted. If this parameter does not match the version number of the list item on the protocol server, the method MUST return an error. If the list (1) is a document library, this parameter MUST be ignored.

See section 2.2.3.1 for the list of parameters used to call the **Delete** method from the DisplayPost method (section 3.1.4.8).

3.1.4.2.1 Return Values

If the **Delete** method is called by the DisplayPost method (section 3.1.4.8), it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **Delete** method is not called by the DisplayPost method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the value of the *NextUsing* parameter if that parameter was specified. If not specified, the protocol client is redirected to the **default view** of the list (1).
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.3 DeleteField

The **DeleteField** method is called to delete a field from a list (1). If called through the DisplayPost method, see section 2.2.3.1 for the **Method** element parameters that are used to call **DeleteField**.

Cmd: A string that specifies the method name. If the **DeleteField** method is used, the *Cmd* parameter MUST be present and have the value "DeleteField".

List: The identifier of the list (1) from which the field is to be deleted. This parameter MUST be a GUID and it MUST be present.

Field: A string that specifies the name of the field to be deleted. This parameter MUST be present.

NextUsing: A string that specifies the server-relative URL to which the protocol client is to be redirected after the method is finished.

owshiddenversion: An integer that specifies the version of the list (1) on which this operation is to be performed. This represents the version of the list (1).

3.1.4.3.1 Return Values

If the **DeleteField** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or the field does not exist, or a negative value if an error occurs.

If the **DeleteField** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the value of the *NextUsing* parameter if that parameter was specified. If the *NextUsing* parameter is not specified, the redirection URL MUST be for the settings page for the list (1) from which the field (2) was deleted.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.4 DeleteList

The **DeleteList** method is called to delete a list (1). For calling **DeleteList** through the **DisplayPost** method, see section 2.2.3.1 for the **Method** elements parameters used to call **DeleteList**:

Cmd: A string that specifies the method name. If the **DeleteList** method is used, the *Cmd* parameter MUST be present and have the value "DeleteList".

List: The GUID of the list (1) to be deleted. This parameter MUST be present.

3.1.4.4.1 Return Values

Protocol server responses MUST have the HTTP **Content-Type** "text/html".

If the **DeleteList** method is called through the DisplayPost method, it MUST return "0" (zero) it is successful or a negative value if an error occurs.

If the **DeleteList** method is not called through the DisplayPost method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be a view of all of the content on the site.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.5 DeleteView

The **DeleteView** method is called to delete a view of a list (1). For calling **DeleteView** through the DisplayPost method, see section 2.2.3.1 for the **Method** element parameters used to call **DeleteView**:

Cmd: A string that specifies the method name. If the **DeleteView** method is used, the *Cmd* parameter MUST be present and have the value "DeleteView."

List: A GUID that identifies the list (1) from which to delete the view. This parameter MUST be present.

View: A GUID that identifies the view to delete. This parameter MUST be present.

NextUsing: A string that specifies the URL to which the client is to be redirected after the method is finished. The URL MUST be a server-relative URL or an absolute URL on the same server.

3.1.4.5.1 Return Values

If the **DeleteView** method is called through the DisplayPost method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **DeleteView** method is not called through the DisplayPost method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the value of the specified *NextUsing* parameter and appended with the value "?List=*GUID*", where *GUID* is the value of the *List* parameter. If the *NextUsing* parameter is set to a URL that is not on the server, the protocol server MUST redirect the protocol client to the site's **home page**. If the *NextUsing* parameter is not specified, the protocol client is redirected to the default view of the list (1).
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.6 DialogView

The **DialogView** method is called to obtain a view of all of the document libraries on a site (2), a specific document library, or a folder within a document library, rendered in HTML. This method is used by a protocol client to display files and folders in a file dialog box. **DialogView** takes the following parameters:

dialogview: A string that specifies the view to return. This parameter MUST be present. This string MUST be one of the following values:

Value	Description
FileOpen	The protocol server MUST return HTML content.
FileSave	The protocol server MUST return HTML content.
SaveForm	The protocol server MUST return HTML content representing the properties of the file specified by the *location* parameter.
All other values	The protocol server MUST ignore the parameter value.

location: Specifies the site-relative URL of the document library or a folder or file within a document library. If the *location* parameter is passed without specifying a value, this method MUST return a view of all the document libraries on the site. If the *dialogview* parameter passes the value "SaveForm", the URL for the *location* parameter MUST specify the file being saved.

FileDialogFilterValue: Specifies the file name extension to filter the view that is returned. If multiple file name extensions are specified, the client MUST separate them with a semicolon (;). For example: "*.htm;*.html;*.aspx".

3.1.4.6.1 Return Values

The **DialogView** method MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
200	Success. Returns the view of a document library, a folder in a document library, or the form that is used to specify document properties when saving a file. If the result is not an error, the protocol server response MUST be an HTML document that contains a **table** element with an **id** attribute that is set to the value "FileDialogView". Each file in the selected location MUST be represented as a **tr** element in the **table** element with an **id** attribute set to the server-relative URL of the file and a **fileattribute** attribute with the value "file". Each folder in the selected location MUST be represented as a **tr** element in the **table** with an **id** attribute set to the server-relative URL of the folder and a **fileattribute** attribute with the value "folder".
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.
410	Error. If the result is an error because of an incorrect **location** parameter, the HTTP return value MUST be "410".

3.1.4.7 Display

The **Display** method is called to obtain a set of list items from a list (1). If called through the **DisplayPost** method, see section 2.2.3.1 for the **Method** element parameters that are used to call the **Display** method:

Cmd: A string that specifies the method name. If the **Display** method is used, the *Cmd* parameter MUST be present and have the value "Display".

List: The GUID of the list (1) that contains the list item data. This parameter MUST be present.

XMLDATA: A string that specifies that the return results MUST be in XML format. When specified, the output of the method is XML. The value of *XMLDATA* MUST be ignored. The parameter MUST be present.

View: The GUID of the view that is used to determine the set of list items that are returned. If this parameter is not present, then the default view MUST be used. The *View* parameter is used to filter and sort the returned set of list items. The format of this parameter includes braces and dashes separating the GUID components, and any alphabetic characters. Alphabetic characters that are passed in this parameter MUST be uppercase. For example, {B79FF069-A491-4827-97E4-9E92CC979619}, not b79ff069a491482797e49e92CC979619.

Query: A string whose format is a sequence of field names separated by spaces, or else the first character of this parameter MUST be an asterisk (*). If the first character is an asterisk, then the set of fields MUST be the full set of all fields in the list (1). If the *Query* parameter is not specified, the query for the specified view is used. If the list (1) is the **user information list**, and the *Query* parameter is not specified, then the *Query* parameter defaults to the value "ID Name Title EMail IsSiteAdmin".

Using: A string that specifies a URL for the file that is used to export a list item or list (1). The URL MUST end with a file name that is one of the values in the following table. The *List* parameter MUST also be used. When the file name is "event.ics" or "vcard.vcf", the *ID* parameter MUST also be used. The appropriate file format is returned in the body of the HTTP response, and the **Content-Type** HTTP header is set appropriately according to the following table:

File name	Returns	Content-Type
event.ics	An **iCalendar** file that represents the calendar list item.	text/calendar
vcard.vcf	A **vCard** file that represents the contact list item.	text/x-vcard
query.iqy	A query file that represents the list item.	text/x-ms-iqy

If the *XMLDATA* parameter is present, the *Using* parameter MUST be ignored. For more information about the *Using* parameter, see section 3.1.4.7.1.

CacheControl: A string that determines the emission of a **Cache-Control** HTTP header. If *CacheControl* is specified, or the *XMLDATA* parameter is specified, the protocol server MUST NOT emit the **Cache-Control** HTTP header. Otherwise, the protocol server MUST emit the **Cache-Control** header with the value "no-cache". The value of the *CacheControl* parameter MUST be ignored.

ID: The identifier of the list item to return. If the *XMLDATA* parameter is present, The *ID* parameter MUST be ignored.

3.1.4.7.1 IQY sample output

If the *Using* parameter specifies a URL that ends with the file name "query.iqy", the format of the **DISPLAY** method return value has the following form:

```
WEB
1
http://<server>/<site>/_vti_bin/owssvr.dll?XMLDATA=1&List=<ListID>&View=<ViewID>&RowLimit=0&R
ootFolder=<RootFolder>

Selection=<ListID>-<ViewID>
EditWebPage=
```

```
Formatting=None
PreFormattedTextToColumns=True
ConsecutiveDelimitersAsOne=True
SingleBlockTextImport=False
DisableDateRecognition=False
DisableRedirections=False
SharePointApplication=http://<server>/<site>/_vti_bin
SharePointListView=<ViewID>
SharePointListName=<ListID>
RootFolder=<RootFolder>
```

Each line ends with a line feed character.

Server: The name of the protocol server.

Site: The server-relative URL of the site.

ListID: The identifier of the list (1).

ViewID: The identifier of the view.

RootFolder: A folder in the list (1).

3.1.4.7.2 The Using Parameter

The *Using* parameter enables the protocol client to retrieve data from the protocol server in the form of a specific file format. If it is used in conjunction with the *List* parameter and the *ID* parameter, the list item is returned in the file format specified by the file name in the *Using* parameter, as specified in section 3.1.4.7.

If the *Using* parameter specifies a URL whose trailing file name is one of the values specified in section 3.1.4.7, then the **Content-Type** HTTP header MUST be sent with the associated value.

3.1.4.7.3 The XMLDATA Parameter

If the *XMLDATA* parameter is specified, the format of the output conforms to a common XML schema that is used to represent data from the protocol server. The schema format is specified in [MS-PRSTFR] section 2.

3.1.4.7.4 Return Values

If the **Display** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs. If it finishes successfully, the protocol server MUST return the requested data. For information about the format of the data that is returned see the parameters description in section 3.1.4.7.1.

If the **Display** method is not called through DisplayPost method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
200	Success. The protocol server response body MUST contain the requested data. For information about the format of the data that is returned see the parameters description in section 3.1.4.7.1.
401	Security not validated. The credentials that were supplied by the user are not valid.

Value	Description
404	Error.

3.1.4.8 DisplayPost

The **DisplayPost** method is called to run methods that are invoked through HTTP POST operations, render Collaborative Application Markup Language (CAML), or, when used independently, display information about lists (1) or list views.

If called in HTTP POST operation, the **DisplayPost** method takes the following parameters**:**

Cmd: A string that specifies the method name. The *Cmd* parameter MUST be present and have the value "DisplayPost".

NextUsing: A string that specifies the URL to which the client is to be redirected after the method is finished. The URL MUST be either a server-relative URL or absolute URL on the same server.

PostBody: A string that specifies either a single method in the form of a **Method** element or multiple methods in the form of a **Batch** element, as specified in section 2.2.3.1.

If the **DisplayPost** method is called through itself using the **Method** or **Batch** elements, as specified in section 2.2.3.1, the **Method** element in the **PostBody** element takes the following parameters**:**

View: A string that specifies if the protocol server is to display information about lists or list views. For information about the format of the information that is returned by the protocol server, see section 3.1.4.8.1.

To display information about a list view, the GUID of the associated list MUST be specified in the **SetList** element. For a specific list view, the *View* parameter MUST be the list view GUID. For the default view of the list (1), the *View* parameter MUST be empty or not present.

To display information about all of the lists (1) on the site, the *View* parameter MUST be set to the value "EnumLists". To display information about only the document libraries on the site, the *View* parameter MUST be set to the value "FileDialogView". In both cases, the **SetList** element MUST be set to the value "Lists".

For more information about the **SetList** element, see the **Method** element, as specified in section 2.2.3.1.

PostBody: A string that specifies the CAML to render. This parameter MUST be present when using the **DisplayPost** method to render CAML.

XMLDATA: A **Boolean** value that tells the protocol server to display the schema and data of or list views. The value of this parameter MUST be **true**. This parameter MUST be present when using the **DisplayPost** method to display the schema and data of a list (1) or list view and MUST NOT be present when using the **DisplayPost** method to render CAML.

3.1.4.8.1 Return Values

If the **DisplayPost** method is called directly, it MUST return "0" (zero) in the Result element if the method is successful or a negative value if an error occurs. The contents of the Result element that are returned by the protocol server, as specified in section 2.2.3.2, MUST be one of the following:

View	SetList	PostBody	XMLDATA	Result
		CAML	not present	The protocol server MUST return the rendered CAML.
List view GUID	list GUID		TRUE	The protocol server MUST return an enumeration of all of the fields in the specified view for the specified list (1), and the data in those fields for each item in the specified view.
Empty or not present	list GUID		TRUE	The protocol server MUST return an enumeration of all of the fields in the default view for the specified list (1), and the data in those fields for each item in the default view.
EnumLists	Lists		TRUE	The protocol server MUST return an enumeration of all of the metadata fields for lists, and the data in those fields for each list (1).
FileDialogView	Lists		TRUE	The protocol server MUST return an enumeration of all of the metadata fields for document libraries, and the data in those fields for each document library.

For information about the format of **ListDefinition** metadata field enumerations, see [MS-WSSCAML] section 2.3.2.6 For information about the format of field data, see [MS-LISTSWS] section 3.1.4.21.2.2.

As an HTTP POST call, the **DisplayPost** method sends an HTTP response to the protocol client that MUST be one of the following values:

Value	Description
200	The protocol server called the **DisplayPost** method successfully. The protocol server MUST send an XML response with this HTTP response. For information about the format of the XML response, see Result and WECFileList.
302	The protocol server MUST give this response only when the protocol client supplies a *NextUsing* parameter. If the *NextUsing* parameter is set to a server-relative URL or a URL on the protocol server, the redirection URL MUST be the value of the *NextUsing* parameter**.** If the *NextUsing* parameter is set to a URL that is not on the protocol server, the protocol server MUST redirect the protocol client to the home page for the site.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.9 ExportList

The **ExportList** method is called to export the schema of a list (1) in Collaborative Application Markup Language (CAML) format, as specified in [MS-WSSCAML] section 2.3.2.6.

If the **ExportList** method is called through the **DisplayPost** method, see section 2.2.3.1 for information about the **Method** element parameters that are used to call **ExportList**:

Cmd: A string that specifies the method name. If the **ExportList** method is used, the *Cmd* parameter MUST be present and have the value "ExportList".

List: Specifies the list (1) to export. This parameter MUST be set to the GUID of the requested list (1). This parameter MUST be present.

ExtendedFieldsProperties: A **Boolean** value, "1" or "0" (zero) that specifies whether to include additional information for fields with calculated values or calculated defaults. If *ExtendedFieldsProperties* is omitted, its value MUST be assumed to be "0". If *ExtendedFieldsProperties* is "1" the protocol server MUST include the **FormulaDisplayNames** elements, as specified in [MS-WSSCAML], for fields with calculated values, and **DefaultFormulaValue** elements, as specified in [MS-WSSCAML], for fields with calculated defaults. If the *ExtendedFieldsProperties* parameter value is "0" the protocol server MUST exclude these elements.

ExcludeViews: A string. The content of this parameter MUST be ignored by the protocol server. Any value in this parameter specifies that the protocol server MUST exclude the view information from the returned schema. If the value of the *ExcludeViews* parameter is a non-empty string, the protocol server MUST exclude the **Views** element and all of its child elements from the **MetaData** portion of the returned schema, as specified in [MS-WSSCAML] section 2.3.2.17.3. If the *ExcludeViews* parameter is omitted, the protocol server MUST NOT exclude the **Views** element from the schema.

ExcludeFields: A string**.** The content of this parameter MUST be ignored by the protocol server. Any value in this parameter specifies that the protocol server MUST exclude the field (2) information from the returned schema. If *ExcludeFields* is a non-empty string, the protocol server MUST exclude the **Fields** element and all of its child elements from the **MetaData** portion of the returned schema, as specified in [MS-WSSCAML] section 2.3.2.17.3. If *ExcludeFields* is omitted, the protocol server MUST NOT exclude the **Fields** element from the schema.

3.1.4.9.1 Return Values

If the **ExportList** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs. The return **List** element in **Result** element MUST be the schema of the requested list (1), in Collaborative Application Markup Language (CAML) format as specified in [MS-WSSCAML] section 2.3.2.12.

If it is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
200	Success. The body of the response MUST be the schema of the requested list (1) in CAML format, as specified in [MS-WSSCAML] section 2.3.2.12.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.10 GetProjSchema

The **GetProjSchema** method is used to obtain XML schemas from a site. For calling **GetProjSchema** through the **DisplayPost** method, see section 2.2.3.1 for the **Method** elements parameters.

Cmd: A string that specifies the method name. If the **GetProjSchema** method is used, the *Cmd* parameter MUST be present and have the value "GetProjSchema."

SiteTemplate: A string that specifies the site template to return. This parameter is either empty or SHOULD<8> be set to one of three values: "docicon", "fldtypes", or "vwstyles".

GetProjSchema returns localized schemas depending on the *SiteTemplate* parameter as follows:

Value	Description
docicon	Specifies that the content of a document icon description file, which is stored on the front-end Web server, is returned.
fldtypes	Specifies that the content of a field (2) type description file, which is stored on the front-end Web server, is returned.
vwstyles	Specifies that the content of a view style description file, which is stored on the front-end Web server, is returned.

If any other value is specified or this parameter is not specified, a project schema is formed by merging the contents of a site definition configuration description file, which is stored on the front-end Web server for the current site template, and a list template definition, as specified in [MS-WSSCAML] section 2.3.2.13.

3.1.4.10.1 Return Values

If the **GetProjSchema** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs. If it finishes successfully, the **Result** element MUST contain the requested schema.

If the **GetProjSchema** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
200	Success. The body of the response MUST contain the requested schema.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.11 GetUsageBlob

The **GetUsageBlob** method is called to obtain usage data for a site. It takes the following parameters:

Cmd: A string that specifies the method name. If the **GetUsageBlob** method is used, the *Cmd* parameter MUST be present and have the value "GetUsageBlob."

BlobType: A string that specifies the type of the usage data requested. This parameter MUST be present. This parameter MUST be set to the value "current" or "old". If this parameter is set to the value "current", the protocol server MUST return the daily usage data. If this parameter is set to "old", the protocol server MUST return the monthly usage data.

3.1.4.11.1 Return Values

If the **GetUsageBlob** method is successful, it returns the **Usage Data** binary field structure, as specified in section 2.2.9.1.1, in an HTTP response.

The HTTP status code MUST be one of the following values:

Value	Description
200	Success.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

Protocol clients cannot use the **DisplayPost** method to call the **GetUsageBlob** method.

3.1.4.12 HitCounter

The **HitCounter** method is called to increment the hit count on a page that contains a hit counter and to obtain the image of the hit counter. It takes the following parameters:

Cmd: A string that specifies the method name. If the **HitCounter** method is used, the *Cmd* parameter MUST be present and have the value "HitCounter."

Page: A string that specifies the site-relative URL of the page that contains the hit counter. This parameter MUST be present.

Image: A string that specifies a set of default images that are used by the protocol server for the hit counter. This parameter MUST be set to a number between "0" and "4", inclusive. Either this parameter or the *Custom* parameter MUST be present.

Custom: A string that specifies the site-relative URL of a custom image file that is used for the hit counter. The image file MUST be in Graphics Interchange Format (.gif) format. Either this parameter or the *Image* parameter MUST be present.

Digits: A string that specifies a fixed number of digits for the counter. This parameter MUST be set to a number between "1" and "10", inclusive.

The behavior of the *Image* and *Custom* parameters is mutually exclusive. If both parameters are present then the *Image* parameter takes precedence. If the *Image* parameter is not set between "0" and "4", inclusive, it MUST default to "0".

3.1.4.12.1 Return Values

The **HitCounter** method MUST send an HTTP response to the protocol client and that response MUST be one of the following values.

Value	Description
200	Success. The body of the response SHOULD<9> include an image of the hit counter in .gif format.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.13 ModListSettings

The **ModListSettings** method is called to modify the settings of a list (1). If it is called through the **DisplayPost** method, see section 2.2.3.1 for information about the **Method** element parameters

that are used to call **ModListSettings**. Otherwise, the **ModListSettings** method takes the following parameters:

Cmd: A string that specifies the method name. If the **ModListSettings** method is used, the *Cmd* parameter MUST be present and have the value "ModListSettings."

List: The GUID of the list (1) for which the settings are to be changed. This parameter MUST be present.

OldListTitle: A string that the protocol server MUST ignore.

NewListTitle: A string corresponding to the **Title** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. This MUST be unique within the site. This MUST NOT be more than 255 characters.

Description: A string corresponding to the **Description** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. This MUST NOT be more than 1500 characters.

ReadSecurity: An integer that specifies the security level for viewing items in the list (1). The *ReadSecurity* parameter MUST be set to one of the following values:

Value	Description
1	Users can read all items.
2	Users can read only the items they created.

WriteSecurity: An integer that specifies the security level for writing items in the list (1). The *WriteSecurity* parameter MUST be set to one of the following values.

Value	Description
1	Users can edit all items or add new items.
2	Users can edit only the items that they created or add new items.
4	Users can neither edit nor add any item.

displayOnLeft: A **Boolean** value that specifies whether a link to the new list (1) appears in the site navigation.

DisableAttachments: An integer corresponding to the **DisableAttachments** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. This MUST be one of the following values**:**

Value	Description
1	Items in the list (1) can have attachments.
2	Items in the list (1) cannot have attachments.

Direction: An integer corresponding to the **Direction** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. This MUST be one of the following values:

Value	Description
0	List inherits direction from **parent site**.
1	List direction is left-to-right.
2	List direction is right-to-left.

Moderated: A **Boolean** value corresponding to the **ModerationType** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

NewListTemplate: A string that specifies the URL of the **document template (2)** that is associated with the list (1). The document template (2) MUST exist in the "Forms" directory of the list (1).

EnableAssignToEmail: A **Boolean** value or a **Boolean** one/zero value corresponding to the **EmailAssignTo** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

NewDefaultView: The identifier of a list view to be displayed by default when navigating to the list (1). This MUST be a GUID.

DisableFolders: A **Boolean** value corresponding to the **FolderCreation** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

AddToDefaultView: A **Boolean** value that indicates whether the attachment column is to be added to the **default list view**.

showUsernames: A **Boolean** value corresponding to the inverted **Boolean** value of the **SuppressNameDisplay** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

EventSinkAssembly: A string corresponding to the **EventSinkAssembly** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6, The *EventSinkAssembly* parameter value MUST be no more than 255 characters and the parameter MUST be present if *EventSinkClass* parameter is present.

EventSinkClass: A string corresponding to the **EventSinkClass** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. The *EventSinkClass* parameter value MUST be no more than 255 characters, and the parameter MUST be present if *EventSinkAssembly* parameter is present. The *EventSinkClass* MUST implement the IListEventSink **interface (2)**.

EventSinkData: A string corresponding to the **EventSinkData** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. The *EventSinkData* parameter value MUST be no more than 255 characters.

AllowDeletion: A **Boolean** value corresponding to the **AllowDeletion** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

OrderedList: A **Boolean** value corresponding to the **OrderedList** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

PublicList: A **Boolean** value corresponding to the **PublicList** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

HiddenList: A **Boolean** value corresponding to the **HiddenList** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

RootWebOnly: A **Boolean** value corresponding to the **RootWebOnly** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6

GlobalMtgDataList: A string corresponding to the inverted **Boolean** value of the **MultipleMtgDataList** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. The *GlobalMtgDataList* parameter value MUST be "TRUE".

VersioningEnabled: A **Boolean** value corresponding to the **VersioningEnabled** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

ForceCheckout: A **Boolean** value corresponding to the **ForceCheckout** attribute of the **ListDefinition** type, as specified by [MS-WSSCAML] section 2.3.2.6.

EnableMinorVersions: A **Boolean** value corresponding to the **EnableMinorVersions** attribute of the ListDefinition type, as specified in[MS-WSSCAML] section 2.3.2.6.

DraftVersionVisibility: An integer corresponding to **DraftVersionVisibility** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. The *DraftVersionVisibility* parameter value MUST be one of the following values:

Value	Description
0	Users who can read items can see drafts.
1	Users are required to have edit rights to see drafts.
2	Users are required to have approval rights to see drafts.

DefaultItemOpen: A one/zero **Boolean** value corresponding to the **DefaultItemOpen** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

AllowMultiVote: A **Boolean** on/off value corresponding to the **AllowMultiVote** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

syndicationDisabled: A **Boolean** on/off value that enables or disables RSS syndication for the list.

IrmEnabled: A **Boolean** value that enables or disables **Information Rights Management (IRM)** for the list.

IrmExpire: A **Boolean** value that specifies whether IRM is expired.

IrmReject: A **Boolean** value specifying whether IRM is rejected.

CacheSchema: A **Boolean** value corresponding to the **CacheSchema** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

NoCrawl: A **Boolean** value that disables or enables indexing of the list in a **full-text index catalog**.

WorkflowsAssociated: A **Boolean** value indicating whether the list is associated with a **workflow (2)**.

EnableContentTypes: A **Boolean** value corresponding to the **EnableContentTypes** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

EnableThumbnails: A **Boolean** value corresponding to the **EnableThumbnails** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

AllowEveryoneViewItems: A **Boolean** value corresponding to the **AllowEveryoneViewItems** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

DisableDeployWithDependentList: A **Boolean** value corresponding to the **DisableDeployWithDependentList** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

EmailInsertsFolder: A string. It MUST be ignored by the protocol server.

EmailAlias: A string that specifies the e-mail address for a list (1). This MUST be no more than 128 characters.

SendToLocationName: A string specifying a location name used in the "Send to" feature of the list (1). This MUST be no more than 255 characters and MUST NOT contain the pipe ('|') character.

SendToLocationUrl: A string to be appended to the *SendToLocationName* parameter value specifying the URL to be used in the "Send to" feature of the list (1). This MUST be no more than 255 characters and MUST NOT contain the pipe ('|') character.

ThumbnailSize: An integer corresponding to the **ThumbnailSize** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

WebImageWidth: An integer corresponding to the **WebImageWidth** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

WebImageHeight: An integer corresponding to the **WebImageHeight** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

EmailInsertsLastSyncTime: A **DateTime**. It MUST be ignored by the protocol server.

MajorVersionLimit: An integer corresponding to the **MajorVersionLimit** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. The *MajorVersionLimit* value MUST be greater than or equal to "0".

MajorWithMinorVersionsLimit: An integer corresponding to the **MajorWithMinorVersionsLimit** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. The *MajorWithMinorVersionsLimit* parameter MUST be greater than or equal to "0".

DefaultWorkflowId: A GUID specifying the identifier of the default workflow of the list.

NextUsing: A string that specifies the URL to which the client is to be redirected after the method is finished. The URL MUST be either a server-relative URL or an absolute URL on the same server.

DisplayForm: A string specifying the URL of the **default form** for the **display form** for the list.

EditForm: A string specifying the URL of the default form for the **edit form** for the list.

NewForm: A string specifying the URL of the default form for the **new form** for the list.

owshiddenversion: An integer that specifies the list **version** of the list on which this operation is to be performed.

ValidationFormula: A string, whose syntax is specified in [MS-WSSTS] section 2.9. The length of this string MUST not exceed 1024 characters. When this parameter is set to a non- empty string, it is used as a formula to validate list items when they are added to this list (1). If the validation fails, and if the *ValidationMessage* parameter value is a non-empty string, the string specified by the *ValidationMessage* parameter SHOULD<10> be returned to the caller.

ValidationMessage: A string. The length of this string SHOULD<11> not exceed 1024 characters. The usage of this parameter is specified in the *ValidationFormula* parameter.

DefaultItemOpenUseListSetting: A **Boolean** value. When *DefaultItemOpenUseListSetting* is set to "TRUE", the *DefaultItemOpen* parameter of the list (1) MUST be used. Otherwise, the protocol server MUST ignore the *DefaultItemOpen* parameter for the list (1). Instead, the protocol server SHOULD<12> use a setting that is global to all lists (1) in the site collection as the value of the *DefaultItemOpen* parameter.

enablePeopleSelector: A **Boolean** value. If the *enablePeopleSelector* is set to "TRUE", and if the value of the *ListTemplate* parameter is "Events", the list (1) SHOULD<13> displays a UI that enables the user to select user identities when adding new list items.

enableResourceSelector: A **Boolean** value. If the value of this parameter is set to "TRUE" and the value of **ListTemplate** is "Events", the list (1) SHOULD<14> display a UI that enables the user to select resources when adding new list items.

EnforceDataValidation: A **Boolean** value. If the value of this parameter is set to "TRUE", the protocol server SHOULD<15> validate that list data meets the requirement of the definition for each field.

StrictTypeCoercion: A **Boolean** value that specifies how calculated column formulas, validation formulas, and default value formulas are calculated for the list (1). If the value is true, no automatic coercion will be performed between data types except between numbers and dates; instead, the error #VALUE! will be returned if a type mismatch is detected. If the value is false, automatic type coercion SHOULD<16> take place between all data types.

PreserveEmptyValues: A **Boolean** value that specifies the behavior of calculated column values. If the value is true, the value of any formula containing the reference to the undefined column will be #NULL; otherwise undefined columns SHOULD<17> be treated as having default values.

NavigateForFormsPages: A **Boolean** value that specifies how links to the list (1) are handled. If the value is true, links to list forms will navigate the browser window to the list form. If the value is false, links to list forms MAY<18> open the list form in a modal dialog in the browser.

HasExternalDataSource: A **Boolean** value. If the value of this parameter is set to "TRUE", the list (1) uses external data source stored outside the content database of the server to display its content. Otherwise the list (1) SHOULD<19> uses data source stored inside the content data base of the server to display its content.

DisableGridEditing: A **Boolean** value. When *DisableGridEditing* is set to "TRUE", the protocol server SHOULD<20> NOT offer an option to bulk edit this list (1) by a spreadsheet application.

ExcludeFromOfflineClient: A **Boolean** value. If *ExcludeFromOfflineClient* is set to "TRUE", and the protocol client requests the list of lists (1) to save offline from the protocol server, then the protocol server MUST NOT include this list (1) in the returned set.

IsApplicationList: A **Boolean** value. When the *IsApplicationList* parameter is set to "TRUE", the items in the list are treated as part of the site rather than content within it, and the list (1) is created and managed by the site, not by the users.

BrowserFileHandling: A string. This MUST be either "strict" or "permissive". If the protocol client sends anything other than these two values, the protocol server MUST interpret it as "permissive". When the value of the *BrowserFileHandling* parameter is "strict", the protocol server adds the **X-ContentType-Option: nosniff** HTTP header when the protocol server returns a document that is stored in this list (1). In addition, the protocol server adds the **X-Download-Options: noopen**

HTTP header if the document is deemed unsafe by the protocol server. It is the protocol server's responsibility to determine which types of documents MAY<21> be are unsafe.

The protocol server MUST NOT change list settings corresponding to parameters that were omitted.

If the list type is an "issue tracking list", the protocol server MUST ignore the *ReadSecurity* and *WriteSecurity* parameters.

If the list type is a "**survey list**" or a document library, the protocol server MUST ignore the *DisableAttachments* parameter.

If *DisableAttachments* is set to "2", the protocol server MUST delete all of attachments for the list.

If *Moderated* is "TRUE", the protocol server MUST add the Approval Status column to the default view of the list.

If the list type is not a document library, the protocol server MUST ignore the *NewListTemplate* parameter.

If the *NewListTemplate* parameter is blank, the protocol server MUST remove the association of the list with any document template (2).

If the *NewDefaultView* parameter is present, then the protocol client MUST also specify **DisplayForm**.

If *DisableAttachments* is not set to "1" or attachments are currently enabled, the protocol server MUST ignore the *AddToDefaultView* parameter.

If **GlobalMtgDataList** is present, the list template MUST NOT be "200" (meeting), "202" (meeting user), or "212" (home page).

If *VersioningEnabled* is set to "TRUE", the protocol server MUST ignore this parameter for lists of type Survey List.

If *VersioningEnabled* is not set to "TRUE", the protocol server MUST ignore the *MajorVersionLimit* parameter.

If *EnableMinorVersions* is set to "TRUE", the protocol server MUST consider *VersioningEnabled* to also be set to "TRUE". If *VersioningEnabled* is set to "FALSE", the protocol server MUST consider *EnableMinorVersions* to also be set to "FALSE". If *VersioningEnabled* is set to "FALSE" and *EnableMinorVersions* is set to "TRUE", the protocol server MUST consider *VersioningEnabled* to also be set to "TRUE".

If *Moderated* is set to "FALSE" and *EnableMinorVersions* is set to "FALSE", the protocol server MUST consider *DraftVersionVisibility* to be "0" (zero).

If *VersioningEnabled* is set to "FALSE" or *EnableMinorVersions* and *Moderated* are both set to "FALSE", the protocol server MUST ignore the *MajorWithMinorVersionsLimit* parameter.

If *owshiddenversion* is present, the list version of the list (1) MUST be the same as the one that was passed as a parameter. If they are not the same, the protocol server MUST fail the method and return a negative error code specifying that there was a version mismatch.

When the method finishes successfully, the protocol server MUST increment the list version of the list (1).

3.1.4.13.1 Return Values

If the **ModListSettings** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **ModListSettings** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the value of the *NextUsing* parameter, if that value was specified. If *NextUsing* is not specified, the redirection URL MUST be the settings page for the list.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.14 MtgKeep

The **MtgKeep** method is called to clear the orphaned state of a meeting instance. If called through the **DisplayPost** method, see section 2.2.3.1 for information about the **Method** parameters used to call **MtgKeep**:

Cmd: A string that specifies the method name. If the **MtgKeep** method is used, the *Cmd* parameter MUST be present and have the value of "MtgKeep."

List: A GUID that specifies the recurring meeting list (1) containing the occurrence. This parameter MUST be present.

EditInstanceID: A required integer that specifies the instance identifier of a single-occurrence meeting that is orphaned, or of an individual occurrence within a recurring meeting that is orphaned. Each occurrence within a recurring meeting MUST be a unique instance identifier by which items in the list (1) are identified.

NextUsing: A string that specifies the URL to which the protocol client is to be redirected once the method is finished. The URL MUST be either a server-relative URL or an absolute URL on the same server.

3.1.4.14.1 Return Values

If the **MtgKeep** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **MtgKeep** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the value of the *NextUsing* parameter if it was specified. If *NextUsing* parameter is not specified, the redirection URL MUST be Meeting Workspace site page.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.15 MtgMove

The **MtgMove** method is called to delete an orphaned meeting instance, or to move the Meeting Workspace site content associated with an orphaned meeting instance to another meeting instance. If called through the **DisplayPost** method, see section 2.2.3.1 for information about the **Method** parameters used to call **MtgMove**:

Cmd: A string that specifies the method name. If the **MtgMove** method is used, the *Cmd* parameter MUST be present and have the value of "MtgMove."

List: A GUID that specifies the recurring meeting list (1) containing the occurrence. This parameter MUST be present.

FromInstanceID: A required integer that specifies the instance identifier of the source of a single-occurrence meeting that is orphaned, or of an individual occurrence within a recurring meeting that is orphaned. Each occurrence within a recurring meeting MUST be a unique instance identifier by which items in the list (1) are identified.

ToInstanceID: A required integer that specifies the instance identifier of the destination of a single-occurrence meeting that is orphaned, or of an individual occurrence within a recurring meeting that is orphaned. Each occurrence within a recurring meeting SHOULD<22> be a unique instance identifier by which items in the list (1) are identified.

Overwrite: If set to "1", the protocol server MUST delete the associated workspace content of the meeting instance specified by *ToInstanceID* parameter if *ToInstanceID* is set to an instance identifier that is currently applied to a meeting instance.

The *Overwrite* parameter is optional. If not specified or not set to "1", the protocol server MUST not delete the associated workspace content of the meeting instance specified by *ToInstanceID* parameter.

NextUsing: A string that specifies the URL to which the protocol client is to be redirected once the method is finished. The URL MUST be either a server-relative URL or an absolute URL on the same server.

3.1.4.15.1 Return Values

If the **MtgMove** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **MtgMove** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the value of the *NextUsing* parameter if it was specified. If the *NextUsing* parameter is not specified, the redirection URL MUST be Meeting Workspace site page.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.16 NewField

The **NewField** method is called to add a field (2) to a list (1). If called through the **DisplayPost** method, see section 2.2.3.1 for information about the **Method** parameters used to call **NewField**:

Cmd: A string that specifies the method name. If the **NewField** method is used, the *Cmd* parameter MUST be present and have the value of "NewField."

List: The identifier of the list (1) to which the new field (2) is to be added. This MUST be a GUID. This parameter MUST be present.

FieldXML: A Collaborative Application Markup Language (CAML) string, as specified in [MS-WSSCAML] section 2.3.2.9, that corresponds to a field (2) element definition. This parameter MUST be present.

AddToDefaultView: A **Boolean** on/off value that specifies whether the newly created field (2) is to be added to the default view of the list (1). If omitted, its value MUST be assumed to be "OFF".

DatesInGregorian: A **Boolean** true/false value that specifies whether dates are stored in the Gregorian format. If omitted, its value MUST be assumed to be "TRUE".

NextUsing: A string that specifies the URL to which the protocol client is to be redirected once the method is finished. The URL MUST be either a server-relative URL or an absolute URL on the same server.

owshiddenversion: An integer that specifies the version of the list (1) on which this operation is to be performed. If present, the version of the list (1) specified by *List* MUST match this parameter.

3.1.4.16.1 Return Values

If the **NewField** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs. On success, **NewField** also returns a **FieldName** XML element containing the name of the newly created field.

If the **NewField** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the value of the *NextUsing* parameter if it was specified. If *NextUsing* is not specified, the redirection URL MUST be the settings page for the list.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.17 NewList

The **NewList** method is called to create a list (1). If **NewList** is called through the **DisplayPost** method, see section 2.2.3.1 for information about the method parameters used to call **NewList**.

Cmd: A string that specifies the method name. If the **NewList** method is used, the *Cmd* parameter MUST be present and have the value of "NewList."

ListTemplate: The identifier of the built-in template, from which the new list (1) is created. *ListTemplate* MUST be present. The *ListTemplate* parameter value MUST have one of the values from the table listed in [MS-WSSFO2] section 2.2.3.12.

CustomTemplate: A string that specifies the name of a custom template, from which the new list (1) is created.

Title: A string that specifies the title of the new list (1). The default value for *Title* is an empty string, unless otherwise specified.

Description: A string that specifies the description of the new list (1).

displayOnLeft: A **Boolean** value that specifies whether a link to the new list (1) will appear in the site navigation.

GlobalMtgDataList: A **Boolean** value that specifies the **MultipleMtgDataList** attribute of the new list as specified in [MS-WSSCAML] section 2.3.2.6, ListDefinition Type.

FeatureId: A GUID that specifies the **FeatureId** attribute of the list as specified in [MS-WSSCAML] section 2.3.2.6, ListDefinition Type.

DocumentTemplateType: An integer that specifies the document template for the new list (1). If *DocumentTemplateType* is present, a value from the following table MAY<23> be used.

Value	Document template
100	None
101	Word
103	Excel<24>
104	PowerPoint<25>
121	Word
122	Excel
123	PowerPoint
111	OneNote
102	FrontPage
105	BasicPage (.aspx)
106	WebPartPage

Direction: An integer that specifies the [MS-WSSCAML] section 2.3.2.6, **ListDefinition** attribute of the list (1).

AllowMultiVote: A **Boolean** on/off value that corresponds to the **AllowMultiVote** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

showUsernames: A **Boolean** on/off value that specifies whether the user's name is displayed in a survey list.

EnableAssignToEmail: A **Boolean** one/zero value that specifies the **EmailAssignTo** attribute, as specified in [MS-WSSCAML] section 2.3.2.6, ListDefinition Type, of the new list (1).

ThumbnailSize: An integer that corresponds to the **ThumbnailSize** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

WebImageWidth: An integer that corresponds to the **WebImageWidth** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

WebImageHeight: An integer that corresponds to the **WebImageHeight** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

VersioningEnabled: A **Boolean** value that corresponds to the **VersioningEnabled** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

EnableMinorVersions: A **Boolean** value that corresponds to the **EnableMinorVersions** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

ForceCheckout: A **Boolean** value that corresponds to the **ForceCheckout** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

EnableContentTypes: A **Boolean** value that corresponds to the **EnableContentTypes** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

RootFolder: A string that specifies the **root folder** for the new list in its Web server-relative URL.

DocUrlForWebPart: The server-relative URL of the page where a list view Web Part will be added.

WebPartZoneID: The identifier of the **Web Part zone** where the list view Web Part will be put in.

NextUsing: A string that specifies the URL to which the protocol client is to be redirected once the method is finished. The URL MUST be either a server-relative URL or an absolute URL on the same server.

DisableAttachments: A **Boolean** value that corresponds to the **DisableAttachments** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6. The *DisableAttachments* parameter MUST be ignored.

Moderated: A **Boolean** value that corresponds to the **ModeratedList** attribute of the **ListDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.6.

enablePeopleSelector: A **Boolean** value. Only used if the *ListTemplate* parameter value is "Events". If the *enablePeopleSelector* is "TRUE", it specifies that the list (1) SHOULD<26> display the UI which enables the user to select user identities when adding new list items.

enableResourceSelector: A **Boolean** value. Only used if the *ListTemplate* parameter value is set to "Events". If the *enableResourceSelector* is "TRUE", it specifies that the list (1) SHOULD<27> displays the UI that enables the user to select resources when adding new list items.

If *CustomTemplate* is specified, the new list (1) MUST be created from the specified custom template instead of the built-in template specified by the *ListTemplate* parameter. The parameters that follow *CustomTemplate* MUST take the default values from the specified custom template, except the following four parameters: *Title*, *Description*, *displayOnLeft* and *GlobalMtgDataList*. If any of these four parameters are unspecified, the list (1) MUST take the default value from the specified custom template.

If *CustomTemplate* is unspecified, all the parameters that follow MUST take the values from the input. If any of them is unspecified, it MUST take the default value from the built-in template specified by the *ListTemplate* parameter.

If *Title* is not unique, the **NewList** method will return an error.

3.1.4.17.1 Return Values

Protocol server responses MUST have the HTTP **Content-Type** "application/xml".

If the **NewList** method is called through the DisplayPost method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **NewList** method is not called through the DisplayPost method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. If the *NextUsing* parameter is specified, the redirection URL MUST be the value of the *NextUsing* parameter appended by "?List=" and the GUID of the newly created list. If *NextUsing* parameter is unspecified, the redirection URL MUST be the default view of the newly created list.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.18 NewView

The **NewView** method is called to create a new view of a list (1) with a more complex set of parameters than the **NewViewPage** method (See section 3.1.4.19). If **NewView** is called through the **DisplayPost** method, see section 2.2.3.1 for information about the method parameters used to call **NewView**.

Cmd: A string that specifies the method name. If the **NewView** method is used, the *Cmd* parameter MUST be present and have the value of "NewView".

List: A GUID identifying the list (1) to add the view to. This parameter MUST be present.

Personal: A **Boolean** value that is used to set the value of the **Personal** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.2,

ViewFPModified: A **Boolean** value. This parameter MUST be ignored by the protocol server.

ReadOnly: A **Boolean** value. This parameter MUST be ignored by the protocol server.

Level: An integer. This parameter MUST be ignored by the protocol server.

ContentTypeId: A string that contains the identifier of the content type. The *ContentTypeID* parameter is used to set the value of the **ContentTypeID** property, as specified in [MS-WSSCAML] section 2.3.1.4.

ContentTypeIdSelector: A string consisting of the value of the *ContentTypeId* parameter. If the value of the *ContentTypeId* parameter is "Other", then the *ContentTypeIdSelector* parameter MUST contain the content type identifier of the view to be created. Otherwise the *ContentTypeIdSelector* parameter MUST be ignored.

BaseViewID: An integer that is used to set the value of the **BaseViewID** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

ViewType: A string that is used to set the value of the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

NewViewName: A string that is used to set the value of the **DisplayName** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

ViewOrder*n*: A string that is used to specify the field (2) on which to sort a view, where *n* is an integer starting at 0 that designates the priority of this sort field (2). The value of the ViewOrder*n* parameter MUST be in the following format**:** *n+1_Field* where *n* is the same *n* as the parameter name and *Field* is the **fname** of the field being used to sort the view. The values of *n* MUST be consecutive. That is, if the ViewOrder*n* parameter is not present for *n=x* then any ViewOrder*n* parameters where *n* is greater than *x* MUST be ignored by the protocol server. If the *ViewType* parameter value is set to "Calendar", the protocol server MUST ignore all ViewOrder*n* parameters where *n* is greater than 1. If the corresponding ShouldDisplay*Field* parameter is absent or set to "FALSE", the ViewOrder*n* parameter MUST be ignored by the protocol server.

ShouldDisplay*Field*: A **Boolean** value. If *ViewType* parameter value is not set to "Calendar", specifies whether the field with the given internal name MUST be displayed in the view, where *Field* is the internal name of the field in question. *Field* MUST be referenced in a *ViewOrder* parameter. For example, if *ViewOrder0*="1_Name", then the *ShouldDisplayName* parameter MUST be checked by the protocol server if it is present. Conversely, if the *ShouldDisplayName* parameter is present, but there is no corresponding ViewOrder*n* parameter, then *ShouldDisplayName* MUST be ignored by the protocol server.

IsExplicit*Field*: A **Boolean** value. If **ViewType** is not set to "Calendar", the IsExplicit*Field* parameter determines whether the field with the given internal name MUST be explicitly declared, where *Field* is the internal name of the field in question. *Field* MUST be referenced in a *ViewOrder* parameter. For example, if *ViewOrder0*="1_Name" is present in the query string, then *IsExplicitName* MUST be checked by the protocol server if it is present. Conversely, if *IsExplicitName* is present, but there is no corresponding *ViewOrder* parameter, then *IsExplicitName* MUST be ignored by the protocol server.

Total*Field*: A string. If the ***ViewType*** parameter value is not set to "Calendar", the Total*Field* parameter specifies the [MS-WSSCAML] section 2.3.1.2, Aggregations Type for the field with the given internal name, where *Field* is the internal name of the field (2) in question. *Field* MUST be referenced in a *ViewOrder* parameter. For example, if ViewOrder0="1_Name", then *TotalName* MUST be checked by the protocol server if it is present. Conversely, if the *TotalName* parameter is present, but there is no corresponding *ViewOrder* parameter, then the *TotalName* parameter MUST be ignored by the protocol server. The Total*Field* parameter MUST be one of the following values:

Value	Description
AVG	Arithmetic Mean of the values
COUNT	Count of records
MAX	Maximum value
MIN	Minimum value
SUM	Sum of the values
STDEV	Standard deviation of the values
VAR	Variance of the values

DefaultView: A **Boolean** value. If the *ContentTypeId* parameter is set to "Other" then this specifies whether this view MUST be set as the default view for the content type designated by the *ContentTypeIdSelector* parameter. Otherwise, this specifies whether the view MUST be set as the

default view of the list (1) designated by the *List* parameter. If the *Personal* parameter value is set to "TRUE", this parameter MUST be ignored.

Recursive: A **Boolean** true/false. If "TRUE", the **Scope** property of the **ViewDefinition** type of the view, as specified in [MS-WSSCAML] section 2.3.2.17, is set to the same value as the **Recursive** property in the **ViewScope** type, as specified in [MS-WSSCAML] section 2.3.1.11.

RecursiveAll: A **Boolean** true/false. If *Recursive* is present, this parameter MUST be ignored by the protocol server. If "TRUE", the **Scope** property of the **ViewDefinition** type of the view, as specified in [MS-WSSCAML] section 2.3.2.17, is set to the same value as the **RecursiveAll** property in the **ViewScope** type, as specified in [MS-WSSCAML] section 2.3.1.11.

ViewStyle: An integer or a string that corresponds to the **ViewStyle** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

IncludeRootFolder: A **Boolean** value that corresponds to the **IncludeRootFolder** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

Ordered: A **Boolean** value that corresponds to the **OrderedView** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

MobileView: A **Boolean** true/false. If *Personal* is set to FALSE, the *MobileView* parameter sets the **MobileView** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

MobileDefaultView: A **Boolean** true/false. If *Personal* is set to FALSE, the *MobileDefaultView* parameter sets the **MobileDefaultView** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

GanttStartDate: A string. If the *ViewType* parameter is set to "Gantt", the *GanttStartDate* parameter sets the **GanttStartDate** property of the **FieldRefDefinitionViewData**, as specified in [MS-WSSCAML] section 2.3.2.18.2.

GanttEndDate: A string. If the *ViewType* parameter is set to "Gantt", the *GanttEndDate* parameter sets the **GanttEndDate** property of the **FieldRefDefinitionViewData**, as specified in [MS-WSSCAML] section 2.3.2.18.2.

GanttTitle: A string. If the *ViewType* parameter is set to "Gantt", the *GanttTitle* parameter sets the **GanttTitle** property of the **FieldRefDefinitionViewData**, as specified in [MS-WSSCAML] section 2.3.2.18.2.

GanttPercentComplete: A string. If the *ViewType* parameter is set to "Gantt", the *GanttPercentComplete* parameter sets the **GanttPercentComplete** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2.

CalViewStyles: A string. If the *ViewType* parameter is set to "Calendar", the *CalViewStyles* parameter sets the **CalendarViewStyle** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

CalendarMonthTitle: A string. If the *ViewType* parameter is set to "Calendar", the *CalendarMonthTitle* parameter sets the **CalendarMonthTitle** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2.

CalendarWeekTitle: A string. If the *ViewType* parameter is set to "Calendar", the *CalendarWeekTitle* parameter sets the **CalendarWeekTitle** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2.

CalendarWeekLocation: A string. If the *ViewType* parameter is set to "Calendar", the *CalendarWeekLocation* parameter sets the **CalendarWeekLocation** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2.

CalendarDayTitle: A string. If the *ViewType* parameter is set to "Calendar", the *CalendarDayTitle* parameter sets the **CalendarDayTitle** property of the **FieldRefDefinitionViewData**, as specified in [MS-WSSCAML] section 2.3.2.18.2.

CalendarDayLocation: A string. If the *ViewType* parameter is set to "Calendar", the *CalendarDayLocation* parameter sets the **CalendarDayLocation** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2.

GroupField*n*: A string. Specifies the internal name of the field to [MS-WSSCAML] section GroupBy where *n* represents an integer starting at "1" that designates the priority of this group-by clause. If the parameter value is set to "None", the protocol server MUST ignore this parameter. If *n* is greater than 1 and GroupAscending*(n-1)* is not present, then GroupField*n* for all *n* greater than or equal to the current *n* MUST be ignored by the protocol server. The values of *n* MUST be consecutive. That is, if GroupField*n* is not present for *n=x*, then any GroupField*n* parameters where *n* is greater than *x* MUST be ignored by the server.

GroupAscending*n*: A **Boolean** value that determines whether the corresponding **GroupBy** field, as determined by *n* where *n* is an integer starting at 1, is sorted in ascending order. If this parameter is not present for a given value of *n*, then GroupAscending*n* for all *n* greater than or equal to the current *n* MUST be ignored by the protocol server.

CollapseGroups: A **Boolean** value that determines whether the **Collapse** property of the **GroupByDefinition** type is "TRUE" or "FALSE", as specified in [MS-WSSCAML] section 2.2.2.5.2. If there is not at least one GroupField*n* parameter present, this parameter MUST be ignored by the protocol server.

SortField*n*: A string that specifies the internal name of the field by which to filter the view, where *n* represents an integer starting at 1 that designates the priority of this sort clause. If set to "None", the server MUST ignore this parameter. If *n* is greater than "1" and SortAscending*(n-1)* is not present, then SortField*n* for all *n* greater than or equal to the current *n* MUST be ignored by the protocol server. The values of *n* MUST be consecutive. That is, if SortField*n* is not present for *n=x* then any SortField*n* parameters where *n* is greater than *x* MUST be ignored by the protocol server.

SortAscending*n*: A **Boolean** value. Specifies whether the corresponding sort field as determined by *n,* where *n* is an integer starting at 1, is sorted in ascending order. If this parameter is not present for a given value of *n*, then SortAscending*n* for all *n* greater than or equal to the current *n* MUST be ignored by the protocol server.

IsThereAQuery: A **Boolean** value that determines whether a **Query** type is attached to this view, as specified in [MS-WSSCAML] section 2.2.

FieldPicker*n*: A string. Specifies the internal name of the field by which to filter the view, where *n* is an integer starting at 1. If set to "None", the protocol server MUST ignore this parameter. If the OperatorPicker*n* or CompareWithValue*n* fields (for the same value of *n* as this parameter) are missing or are not valid, then this parameter MUST be ignored by the protocol server. For example, if *n*="1", then *FieldPicker1*, *OperatorPicker1*, and *CompareWithValue1* MUST all be present and valid or else they will all be ignored by the protocol server. If *IsThereAQuery* is set to "FALSE" or is not present, this parameter MUST be ignored by the protocol server. The values of *n* MUST be consecutive. For example, if five FieldPicker*n* parameters are sent to the protocol server with their respective *n*'s set to 1, 2, 3, 5, 6; 5 and 6 MUST be ignored by the protocol server. For *n*>1, if NextIsAnd*(n-1)* is not present, this parameter MUST be ignored by the protocol server.

OperatorPicker*n*: A string that is used to specify the operator to use in the filter, where *n* is an integer starting at 1 that matches FieldPicker*n*. If the FieldPicker*n* or CompareWithValue*n* fields (for the same value of *n* as this parameter) are missing or are not valid, then this parameter MUST be ignored by the protocol server. For example, if *n*="1", then *FieldPicker1*, *OperatorPicker1*, and *CompareWithValue1* MUST all be present and valid or else they MUST all be ignored by the protocol server. If the *IsThereAQuery* parameter value is set to "FALSE", or if the *IsThereAQuery* parameter is not present, the OperatorPicker*n* parameter MUST be ignored by the protocol server. The values of *n* MUST be consecutive. For example, if five OperatorPicker*n* parameters are sent to the protocol server with their respective *n*'s set to 1, 2, 3, 5, 6; 5 and 6 MUST be ignored by the protocol server. For *n*>1, if NextIsAnd*(n-1)* is not present, this parameter MUST be ignored by the protocol server. The OperatorPicker*n* parameter MUST be set to one of the following values:

Value	Description
Eq	Equal to
Neq	Not equal to
Geq	Greater than or equal to
Leq	Less than or equal to
Gt	Greater than
Lt	Less than
BeginsWith	Begins with
Contains	Contains

CompareWithValue*n*: This parameter can be an integer, string, datetime, or **Boolean** value. CompareWithValue*n* sets the value against which to compare FieldPicker*n*, where *n* is an integer starting at 1 that matches FieldPicker*n*. If the FieldPicker*n* or OperatorPicker*n* fields (for the same value of *n* as this parameter) are missing or are not valid, then this parameter MUST be ignored by the protocol server. For example, if *n*="1", then *FieldPicker1*, *OperatorPicker1*, and *CompareWithValue1* MUST all be present and valid or else they MUST all be ignored by the protocol server. If the *IsThereAQuery* parameter value is set to "FALSE", or the *IsThereAQuery* parameter is not present, then the CompareWithValue*n* parameter MUST be ignored by the protocol server. The values of *n* MUST be consecutive. For *n*>1, if NextIsAnd*(n-1)* is not present, this parameter MUST be ignored by the protocol server. If *LocalizedTodayString* is used in the value of this parameter, it can be followed by a negative or positive offset value. For example, if the *LocalizedTodayString* parameter value is "[Today]" then the value of *CompareWithValue* can be "[Today]", "[Today]+*x*", or "[Today]-*x*" where *x* is an integer. If the value of this parameter is set to "[ID]" then the field designated by the corresponding FieldPicker*n* parameter is compared with the **list item identifier**.

NextIsAnd*n*: A **Boolean** value that specifies whether the **Boolean** operator that relates the n^{th} and $(n+1)^{th}$ filtered fields is an AND. If the *IsThereAQuery* parameter value is set to "FALSE", or if the *IsThereAQuery* parameter is not present, the *NextIsAndn* parameter MUST be ignored by the protocol server. The values of *n* MUST be consecutive. That is, if five NextIsAnd*n* parameters are sent to the protocol server with their respective *n*'s set to 1, 2, 3, 5, 6; 5 and 6 MUST be ignored by the protocol server. If this parameter is present for a given *n* then FieldPicker*(n+1)*, OperatorPicker*(n+1)*, and CompareWithValue*(n+1)* MUST be present, or else the NextIsAnd*n* parameter MUST be ignored. The value of this parameter MUST be either "TRUE" or "FALSE", as specified in the following table:

Value	Description
TRUE	The **Boolean** operator relating the two filters is AND.
FALSE	The **Boolean** operator relating the two filters is OR.

HiddenFilter: A string. Specifies a hidden **restriction (1)** on the items shown in the view as prescribed by [MS-WSSCAML] section 2.2.2.1.3.

LocalizedTodayString: A string. Specifies the string that will be replaced with today's date if used in CompareWithValue*n*. For example, if this parameter is set to "[Today]" then all instances of "[Today]" in any CompareWithValue*n* parameters MUST be replaced with today's date. If this parameter is set to the same value as *LocalizedMeString*, then this parameter MUST be ignored by the protocol server.

LocalizedMeString: A string. Specifies the string that will be replaced with the **current user** identifier if used in CompareWithValue*n*. For example, if this parameter is set to "[Me]" then all instances of "[Me]" in any CompareWithValue*n* parameters MUST be replaced with the current user identifier.

GroupLimit: An integer that MUST be between "1" and "2147483647". If the *GroupLimit* parameter is outside of that range, then the protocol server MUST ignore it and assume "2147483647".The *GroupLimit* parameter sets the **GroupLimit** property of the **GroupByDefinition** type, as specified in [MS-WSSCAML] section 2.2.2.5.2. If there is not at least one valid GroupField*n* parameter, the *GroupLimit* parameter MUST be ignored by the server.

OverrideOrderBy: A **Boolean** value that sets the **Override** property of the **OrderByDefinition** type, as specified in [MS-WSSCAML] section 2.2.2.3.2. If there is not at least one valid SortField*n* parameter, the *OverrideOrderBy* parameter MUST be ignored by the server.

RowLimit: An integer value that sets the **RowLimit** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.3. The value of the *RowLimit* parameter MUST be between "1" and "2147483647". If it is outside of that range, then the server MUST ignore the value provided and assume "2147483647". If the *ViewType* parameter value is set to "Calendar" the *RowLimit* parameter MUST be ignored by the server The attribute is set to "0" (zero) in the generated Collaborative Application Markup Language (CAML).

Paged: A **Boolean** value that sets the **Paged** property of the **RowLimitDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.14. If the *ViewType* parameter value is set to "Calendar" then the *Paged* parameter MUST be ignored by the server.

TabularView: A **Boolean** value. When the *TabularView* parameter is set to "TRUE", the view SHOULD<28> display a checkbox next to each list item in the view, and enable operations to be performed on all the selected items.

InlineEdit: A **Boolean** value. When the *InlineEdit* parameter is set to "TRUE", the view SHOULD<29> enable the user to edit the list items in the current view, without having to navigate to an edit page.

GanttPredecessors: A string that is only used if the *ViewType* parameter value is set to "Gantt". This parameter specifies the name of the field (2) that contains the task that SHOULD<30> be finished prior to starting the task in the current view item.

MethodXml: A string that SHOULD<31> contain the XML fragment for the view method. The schema of the **MethodXml** fragment is specified in section 2.2.5.2.

MobileItemLimit: An integer. The value MUST be between 1 and 99. If it is outside of that range, the server MUST ignore it and assume the value is 99. This parameter specifies the number of items to display if the *MobileView* parameter value is "TRUE". The server SHOULD<32> ignore this value if the *MobileView* parameter value of this view is "FALSE".

3.1.4.18.1 Return Values

If the **NewView** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **NewView** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the URL of the view that was created.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.19 NewViewPage

The **NewViewPage** method is called to create a new view of a list (1). If called through the DisplayPost method, see section 2.2.3.1 for information about the **Method** parameters used to call **NewViewPage:**

Cmd: A string that specifies the method name. If the **NewViewPage** method is used, the *Cmd* parameter MUST be present and have the value of "NewViewPage".

List: The identifier of the list (1) for which the new view is to be added. This MUST be a GUID. This parameter MUST be present.

PageUrl: A site-relative URL where the new view is to be created. This parameter MUST be present. This path MUST be under the root folder for the list (1).

HiddenView: Boolean value that specifies the value of `VIEWFLAG_HIDDEN` This parameter MUST be present.

DisplayName: A string value that specifies the display name for the newly created view. This parameter MUST be present.

ContentTypeID: Specifies the content type identifier for the newly created view. If not specified, it is set to "0x".

BaseViewID: An integer value that specifies the **base view identifier** for the newly created view. If value is passed that is not valid, or if this parameter is omitted, the value of the **BaseViewID** attribute of the **ViewDefinition** type is used, as specified in [MS-WSSCAML] section 2.3.2.17.

3.1.4.19.1 Return Values

If the **NewViewPage** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs. When the method finishes successfully, the **Result** element MUST contain the view metadata as specified by [MS-WSSCAML] for the newly created view.

If the **NewViewPage** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
200	Success. The body of the response MUST contain the view metadata for the newly created view.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.20 NewWebPage

The **NewWebPage** method is called to create a new Web Part Page or a new basic page in a document library. If called through the **DisplayPost** method, see section 2.2.3.1 for information about the **Method** parameters used to call **NewWebPage**:

Cmd: A string that specifies the method name. If the **NewWebPage** method is used, the C*md* parameter MUST be present and have the value of "NewWebPage".

List: The GUID of the list (1). The parameter MUST be present.

ID: A string that MUST be the value "New". This parameter MUST be present.

Type: The type of page to create. This parameter MUST be either "WebPartPage" or "BasicPage". This parameter MUST be present.

WebPartPageTemplate: The template of the Web Part Page. If the parameter *Type* is set to "BasicPage", this parameter is ignored. If the parameter *Type* is set to "WebPartPage", this parameter MUST be present and MUST have one of the following values:

Value	Description
1	Full Page, Vertical
2	Header, Footer, 3 Columns
3	Header, Left Column, Body
4	Header, Right Column, Body
5	Header, Footer, 2 Columns, 4 Rows
6	Header, Footer, 4 Columns, Top Row
7	Left Column, Header, Footer, Top Row, 3 Columns
8	Right Column, Header, Footer, Top Row, 3 Columns

Overwrite: A **Boolean** true/false value that specifies whether to overwrite any file of the same name that exists. This parameter MUST be present.

Title: The file name of the new page. This parameter MUST be present.

RelativeFolderPath: The folder path relative to the root folder of the document library, under which the page is created. If the parameter is not set, the page MUST be created under the root folder of the document library.

NextUsing: A string that specifies the URL to which the client is to be redirected once the method is finished. The URL MUST be either a server-relative URL or an absolute URL on the same server.

If the *Type* parameter is Web Part Page, the title property of the new Web Part Page is set to the *Title* parameter.

3.1.4.20.1 Return Values

If the **NewWebPage** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **NewWebPage** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be to a page to edit the newly created Web Part Page or Basic page. The value of the **NextUsing** parameter MUST be ignored in this case.
401	Security not validated. The credentials that were supplied by the User are not valid.
404	Error.

3.1.4.21 RenderView

The **RenderView** method is called to obtain the HTML that renders a view and view definition (specified in [MS-WSSCAML] section 2.3.2.17) for a list view Web Part. If **RenderView** is called through the **DisplayPost** method, see section 2.2.3.1 for information about the method parameters used to call **RenderView**.

Cmd: A **string** that specifies the method name. If the **RenderView** method is used, the C*md* parameter MUST be present and have the value of "RenderView".

List: A GUID that specifies the list (1) whose view definition (specified in [MS-WSSCAML] section 2.3.2.17) and HTML the client is requesting. This parameter MUST be specified.

View: A GUID that specifies the list view whose view definition (specified in [MS-WSSCAML] section 2.3.2.17) and HTML the client is requesting. This parameter MUST be specified. Alphabetic characters passed in this parameter MUST be uppercase.

UrlBase: Unused. The protocol server MUST ignore this parameter.

3.1.4.21.1 Return Values

If the **RenderView** method is called through the DisplayPost method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs. On success, the **Result** element MUST include the HTML to render the view under the **ViewHTML** element and the view definition, as specified in [MS-WSSCAML] section 2.3.2.17, under the **ViewXML** element.

If the **RenderView** method is not called through the DisplayPost method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
200	Success. The body of the response MUST contain the HTML to render the view under the **ViewHTML** element and the view definition, as specified in [MS-WSSCAML] section 2.3.2.17,

Value	Description
	under the **ViewXML** element.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.22 ReorderFields

The **ReorderFields** method is called to change the order in which fields appear in a list (1). If called through the DisplayPost method, see section 2.2.3.1 for information about the **Method** parameters used to call **ReorderFields**:

Cmd: A string that specifies the method name. If the **ReorderFields** method is used, the *Cmd* parameter MUST be present and have the value of "ReorderFields".

List: The identifier of the list (1) in which the order of the fields is to be changed. This parameter MUST be a GUID. This parameter MUST be present.

ReorderedFields: A Collaborative Application Markup Language (CAML) string corresponding to the new order of the fields within the list (1). This parameter MUST be present. This MUST contain all fields exported by the **ExportList** method (see section 3.1.4.9).

NextUsing: A string that specifies the URL to which the protocol client is to be redirected once the method is finished. The URL MUST be either a server-relative URL or an absolute URL on the same server.

owshiddenversion: An integer that specifies the version of the list (1) on which this operation is to be performed.

If *owshiddenversion* is present, the version of the list (1) MUST match the parameter.

3.1.4.22.1 Return Values

If the **ReorderFields** method is called through the DisplayPost method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **ReorderFields** method is not called through the DisplayPost method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the value of the *NextUsing* parameter if it was specified. If *NextUsing* is not specified, the redirection URL MUST be the settings page for the list (1).
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.23 SiteProvision

The **SiteProvision** method is called to apply a site template or custom site template to an existing site and optionally to add the default set of lists specified in the site template or custom site template. If called through the DisplayPost method, see section 2.2.3.1 for information about the **Method** parameters used to call **SiteProvision**:

Cmd: A string that specifies the method name. If the **SiteProvision** method is used, the *Cmd* parameter MUST be present and have the value of "SiteProvision".

CreateLists: A **Boolean** true/false value that specifies whether the default list (1) types of MUST be created as specified in the site template or custom site template stored on the front-end Web server. If the parameter value is set to "FALSE", the server MUST only store the identifier of the site template. If "TRUE", the server MUST store the specified site template identifier and MUST create the default lists specified in the site template. If *CreateLists* is any other value, its value MUST be assumed to be "FALSE". If it is not present, its value MUST be assumed to be "FALSE" .

SiteTemplate: A string that specifies the site template or custom site template. If it is present, it MUST be the identifier of the site template of a site template or custom site template stored on the front-end Web server. If it is not present, its value MUST be assumed to be "STS#0", which is the default site template.

Protocol clients can retrieve the set of site templates by calling the **GetSiteTemplates** method, as specified in [MS-SITESS] section 3.1.4.5.

3.1.4.23.1 Return Values

If the **SiteProvision** method is called through the DisplayPost method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **SiteProvision** method is not called through the DisplayPost method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
200	Success.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.24 UpdateProject

The **UpdateProject** method is called to update the display name or description of a site. If called through the DisplayPost method, see section 2.2.3.1 for information about the **Method** parameters used to call **UpdateProject**:

Cmd: A string that specifies the method name. If the **UpdateProject** method is used, the C*md* parameter MUST be present and have the value of "UpdateProject".

NewListTitle: A string that specifies the display name of the site.

Description: A string that specifies the description of the site.

owshiddenversion: An integer that specifies the version of the site that is to be updated . This parameter MUST be present. It MUST be the current version of the site or -1. If this parameter is set to -1, the current version of the site will be used. When the method finishes successfully, the version of the site will be incremented by 1.

NextUsing: A string that specifies the URL to which the client is to be redirected once the method is finished. The URL MUST be either a server-relative URL or an absolute URL on the same server.

3.1.4.24.1 Return Values

If the **UpdateProject** method is called through the DisplayPost method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **UpdateProject** method is not called through the DisplayPost method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. If the *NextUsing* parameter is specified, the redirection URL MUST be the value of the *NextUsing* parameter. If the *NextUsing* parameter is unspecified, the redirection URL MUST be the site settings page.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.4.25 UpdateView

The **UpdateView** method is called to update the properties of a View. If called through the DisplayPost method, see section 2.2.3.1 for information about the **Method** parameters used to call **UpdateView**:

Cmd: A string that specifies the method name. If the **UpdateView** method is used, the *Cmd* parameter MUST be present and have the value of "UpdateView".

List: A GUID identifying the list that contains the view to be updated. This parameter MUST be present.

View: A GUID identifying the view to be updated. This parameter MUST be present. This parameter's format includes braces and dashes separating the GUID components, and any alphabetic characters. Alphabetic characters passed in this parameter MUST be uppercase. For example, {B79FF069-A491-4827-97E4-9E92CC979619}, not b79ff069a491482797e49e92CC979619

Personal: A **Boolean** value that sets the **Personal** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17. If this parameter is present, then it MUST be set to the current value of the **Personal** property of the ViewDefinition type.

ViewFPModified: A **Boolean** value that sets the **FPModified** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

ReadOnly: A **Boolean** value that sets the **ReadOnly** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

Level: An integer sets the **Level** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17. If this parameter is present, then it MUST be set to the current value of the **Level** property . If this parameter is absent and the current value of the **Level** property of the view is not "1", then this method MUST fail and return a negative error code.

ContentTypeId: A string that can contain a content type identifier. The *ContentTypeID* parameter sets the value of the **ContentTypeID** property, as specified in [MS-WSSCAML] section 2.3.2.17.

ContentTypeIdSelector: A string that can contain a content type identifier. If the content type identifier is "Other", then this parameter MUST contain the content type identifier for this view.

ViewType: A string. This parameter MUST be ignored by the server.

NewViewName: A string that sets the **Name** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.

NewViewFileName: A string. The file name of the view to be updated. If the *Personal* parameter is set to "FALSE", the *NewViewFileName* parameter MUST be present.

ViewOrder*n*: A string. Specifies the n^{th} field to order by in the view where *n* is an integer starting at 0 that designates the priority of this sort field (2). The value of the ViewOrder*n* parameter MUST be in the following format**:** *n+1_Field*, where *n* is the same *n* as the parameter name and *Field* is the name of the field (2) being used to sort the view. The values of *n* MUST be consecutive. That is, if the ViewOrder*n* parameter value is not present for *n=x,* then any ViewOrder*n* parameters where *n* is greater than *x* MUST be ignored by the server. If the ViewType parameter value is set to "Calendar", the server MUST ignore all ViewOrder*n* parameters where *n* is greater than 1. If the corresponding ShouldDisplay*Field* parameter is absent or set to "FALSE", the ViewOrder*n* parameter MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the **ViewOrder*n*** parameter MUST be ignored by the server.

ShouldDisplay*Field*: A **Boolean** value. If the *ViewType* parameter value is not set to "Calendar", the ShouldDisplay*Field* parameter determines whether the field with the given internal name MUST be displayed in the view, where *Field* is the internal name of the field (2) in question. *Field* MUST be referenced in a ViewOrder*n* parameter.

For example, if *ViewOrder0*="1_Name", then the *ShouldDisplayName* parameter MUST be checked by the server (but it can be absent). Conversely, if a *ShouldDisplayName* parameter is present, but there is no corresponding ViewOrder*n* parameter, then the *ShouldDisplayName* parameter MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", the ShouldDisplay*Field* parameter MUST be ignored by the server.

IsExplicit*Field*: A **Boolean** value. If *ViewType* is not set to "Calendar", the IsExplicit*Field* parameter determines whether the field with the given internal name MUST be declared (see [MS-WSSCAML] section 2.3.2.19.2), where *Field* is the internal name of the field (2) in question. *Field* MUST be referenced in a ViewOrder parameter. For example, if *ViewOrder0*="1_Name" is present in the query string, then *IsExplicitName* MUST be checked by the server if it is present. Conversely, if *IsExplicitName* is present, but there is no corresponding *ViewOrder* parameter, then *IsExplicitName* MUST be ignored by the server. If *ViewFPModified* or *ReadOnly* are "TRUE", this parameter MUST be ignored by the server.

Total*Field*: A string. If the *ViewType* parameter value is not set to "Calendar", specifies the [MS-WSSCAML] section 2.3.1.2, Aggregations Type for the field with the given internal name, where *Field* is the internal name of the field (2) in question. *Field* MUST be referenced in a *ViewOrder* parameter. For example, if ViewOrder0="1_Name" is present in the query string, then a *TotalName* parameter MUST be checked by the server if it is present. Conversely, if a *TotalName* parameter is present, but there is no corresponding *ViewOrder* parameter, then the *TotalName* parameter MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the Total*Field* parameter MUST be ignored by the server.

The Total*Field* parameter MUST be set to one of the following values:

Value	Description
AVG	Arithmetic Mean of the values
COUNT	Count of records

Value	Description
MAX	Maximum value
MIN	Minimum value
SUM	Sum of the values
STDEV	Standard deviation of the values
VAR	Variance of the values

DefaultView: A **Boolean** value. If the *ContentTypeId* parameter is set to "Other" then this specifies whether this view MUST be set as the default view for the content type designated by the *ContentTypeIdSelector* parameter. Otherwise, this specifies whether the view MUST be set as the default view of the list designated by the *List* parameter. If the *Personal* parameter is set to "TRUE", this parameter MUST be ignored.

Recursive: A **Boolean** true/false. If "TRUE", the **Scope** property of the **ViewDefinition** type of the view, as specified in [MS-WSSCAML] section 2.3.2.17, is set to the same value as the **Recursive** property in the **ViewScope** type, as specified in [MS-WSSCAML] section 2.3.1.11. If *ViewFPModified* or *ReadOnly* are "TRUE", this parameter MUST be ignored by the server.

RecursiveAll: A **Boolean** value. If the *Recursive* parameter is present, this parameter MUST be ignored by the server. If "TRUE", the **Scope** property of the **ViewDefinition** type of the view, as specified in [MS-WSSCAML] section 2.3.2.17, is set to the same value as the **RecursiveAll** property in the **ViewScope** type, as specified in [MS-WSSCAML] section 2.3.1.11. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the *RecursiveAll* parameter MUST be ignored by the server.

ViewStyle: An integer or a string that corresponds to the **ViewStyle** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17. If the *ViewFPModified or ReadOnly* parameter values are "TRUE", the *ViewStyle* parameter MUST be ignored by the server.

IncludeRootFolder: A **Boolean** value that corresponds to the **IncludeRootFolder** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17. If the *ViewFPModified or ReadOnly* parameter values are "TRUE", the *IncludeRootFolder* parameter MUST be ignored by the server.

Ordered: A **Boolean** that corresponds to the **OrderedView** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17. If the *ViewFPModified* or *ReadOnly* parameter values are "TRUE", the *Ordered* parameter MUST be ignored by the server.

MobileView: A **Boolean** true/false. If *Personal* is set to FALSE, the *MobileView* parameter sets the **MobileView** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17. If the *ViewFPModified* or *ReadOnly* parameter values are "TRUE", the *MobileView* parameter MUST be ignored by the server.

MobileDefaultView: A **Boolean** true/false. If *Personal* is set to FALSE, the *MobileDefaultView* property sets the **MobileDefaultView** attribute of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17. If the *ViewFPModified or ReadOnly* parameter values are "TRUE", the *MobileDefaultView* parameter MUST be ignored by the server.

GanttStartDate: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12, is set to "Gantt", the *GanttStartDate* parameter sets the **GanttStartDate** property value of the **FieldRefDefinitionViewData** type, as specified in [MS-

WSSCAML] section 2.3.2.18.2. If the *ViewFPModified* or *ReadOnly* values are "TRUE", the *GanttStartDate* parameter MUST be ignored by the server.

GanttEndDate: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12, is set to "Gantt", the *GanttEndDate* parameter sets the **GanttEndDate** property of the **FieldRefDefinitionViewData**, as specified in [MS-WSSCAML] section 2.3.2.18.2. If the *ViewFPModified* or *ReadOnly* values are "TRUE", the *GanttEndDate* parameter MUST be ignored by the server.

GanttTitle: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12, is set to "Gantt", the *GanttTitle* parameter sets the **GanttTitle** property of the **FieldRefDefinitionViewData**, as specified in [MS-WSSCAML] section 2.3.2.18.2. If the *ViewFPModified* or *ReadOnly* values are "TRUE", the *GanttTitle* parameter MUST be ignored by the server.

GanttPercentComplete: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12,is set to "Gantt", the *GanttPercentComplete* parameter sets the **GanttPercentComplete** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2. If the *ViewFPModified* or *ReadOnly* values are "TRUE", the *GanttPercentComplete* parameter MUST be ignored by the server.

CalViewStyles: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12, is set to "Calendar", the *CalViewStyles* parameter sets the **CalendarViewStyle** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17. If the *ViewFPModified* or *ReadOnly* parameters values are "TRUE", the *CalViewStyles* parameter MUST be ignored by the server.

CalendarMonthTitle: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12, is set to "Calendar", the *CalendarMonthTitle* parameter sets the **CalendarMonthTitle** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2. If the *ViewFPModified* or *ReadOnly* values are "TRUE", the *CalendarMonthTitle* parameter MUST be ignored by the server.

CalendarWeekTitle: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12, is set to "Calendar", the *CalendarWeekTitle* parameter sets the **CalendarWeekTitle** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2. If the *ViewFPModified* or *ReadOnly* values are "TRUE", the *CalendarWeekTitle* parameter MUST be ignored by the server.

CalendarWeekLocation: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12, is set to "Calendar", the *CalendarWeekLocation* parameter sets the **CalendarWeekLocation** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2. If the *ViewFPModified* or *ReadOnly* values are "TRUE", the *CalendarWeekLocation* parameter MUST be ignored by the server.

CalendarDayTitle: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12, is set to "Calendar", the *CalendarDayTitle* parameter sets the **CalendarDayTitle** property of the **FieldRefDefinitionViewData**, as specified in [MS-WSSCAML] section 2.3.2.18.2. If *ViewFPModified* or *ReadOnly* are "TRUE", this parameter MUST be ignored by the server.

CalendarDayLocation: A string. If the **ViewType** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.12, is set to "Calendar", the *CalendarDayLocation* parameter sets the **CalendarDayLocation** property of the **FieldRefDefinitionViewData** type, as specified in [MS-WSSCAML] section 2.3.2.18.2. If the *ViewFPModified* or *ReadOnly* values are "TRUE", the *CalendarDayLocation* parameter MUST be ignored by the server.

GroupField*n*: A string. Specifies the internal name of the field to [MS-WSSCAML] section 2.2.2.1, CamlQueryRoot Type\GroupBy where *n* represents an integer starting at 1 that designates the priority of this group-by clause. If set to "None", the server MUST ignore this parameter. If *n* is greater than 1 and GroupAscending*(n-1)* is not present, then GroupField*n* for all *n* greater than or equal to the current *n* MUST be ignored by the server. The values of *n* MUST be consecutive. That is, if GroupField*n* is not present for *n=x*, then any GroupField*n* parameters where *n* is greater than *x* MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the GroupField*n* parameter MUST be ignored by the server.

GroupAscending*n*: A **Boolean** value that determines whether the corresponding **GroupBy** field as determined by *n* where *n* is an integer starting at 1, is sorted in ascending order. If this parameter is not present for a given value of *n*, then GroupAscending*n* for all *n* greater than or equal to the current *n* MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the GroupAscending*n* parameter MUST be ignored by the server.

CollapseGroups: A **Boolean** value that determines whether the **Collapse** property of the **GroupByDefinition** type is "TRUE" or "FALSE", as specified in [MS-WSSCAML] section 2.2.2.5.2. If there is not at least 1 GroupField*n* parameter present, this parameter MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the *CollapseGroups* parameter MUST be ignored by the server.

SortField*n*: A string. Specifies the internal name of the field by which to filter the view, where *n* represents an integer starting at 1 that designates the priority of this sort clause. If set to "None", the server MUST ignore this parameter. If *n* is greater than "1" and SortAscending*(n-1)* is not present, then SortField*n* for all *n* greater than or equal to the current *n* MUST be ignored by the server. The values of *n* MUST be consecutive. That is, if SortField*n* is not present for *n=x* then any SortField*n* parameters where *n* is greater than *x* MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the SortField*n* parameter MUST be ignored by the server.

SortAscending*n*: A **Boolean** value. Specifies whether the corresponding sort field as determined by *n,* where *n* is an integer starting at 1, is sorted in ascending order. If this parameter is not present for a given value of *n*, then SortAscending*n* for all *n* greater than or equal to the current *n* MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the SortAscending*n* parameter MUST be ignored by the server.

IsThereAQuery: A **Boolean** value determines whether a **Query** type is attached to this view, as specified in [MS-WSSCAML] section 2.2. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the *IsThereAQuery* parameter MUST be ignored by the server.

FieldPicker*n*: A string that specifies the internal name of a field by which to filter the view, where *n* is an integer starting at 1. If set to "None", the server MUST ignore this parameter. If the OperatorPicker*n* or CompareWithValue*n* fields (for the same value of *n* as this parameter) are missing or are not valid, then this parameter MUST be ignored by the server. For example, if *n*="1", then *FieldPicker1*, *OperatorPicker1*, and *CompareWithValue1* MUST all be present and valid or else they will all be ignored by the server. If *IsThereAQuery* is set to "FALSE" or is not present, this parameter MUST be ignored by the server. The values of *n* MUST be consecutive. For example, if five FieldPicker*n* parameters are sent to the server with their respective *n*'s set to 1, 2, 3, 5, 6; 5 and 6 MUST be ignored by the server. For *n*>1, if *NextIsAnd(n-1)* is not present, this parameter MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the FieldPicker*n* parameter MUST be ignored by the server.

OperatorPicker*n*: A string that specifies the operator to use in the filter, where *n* is an integer starting at 1 that matches FieldPicker*n*. If the FieldPicker*n* or CompareWithValue*n* fields (for the same value of *n* as this parameter) are missing or are not valid, then this parameter MUST be ignored by the server. For example, if *n*="1", then *FieldPicker1*, *OperatorPicker1*, and

CompareWithValue1 MUST all be present and valid or else they MUST all be ignored by the server. If *IsThereAQuery* is set to "FALSE" or is not present, this parameter MUST be ignored by the server. The values of *n* MUST be consecutive. For example, if five OperatorPicker*n* parameters are sent to the server with their respective *n*'s set to 1, 2, 3, 5, 6; 5 and 6 MUST be ignored by the server. For *n*>1, if *NextIsAnd(n-1)* is not present, this parameter MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the OperatorPicker*n* parameter MUST be ignored by the server. The OperatorPicker*n* parameter MUST be set to one of the following values:

Value	Description
Eq	Equal to
Neq	Not equal to
Geq	Greater than or equal to
Leq	Less than or equal to
Gt	Greater than
Lt	Less than
BeginsWith	Begins with
Contains	Contains

CompareWithValue*n*: This parameter can be an integer, string, datetime, or **Boolean** value. Specifies the value against which to compare FieldPicker*n*, where *n* is an integer starting at 1 that matches FieldPicker*n*. If the FieldPicker*n* or OperatorPicker*n* fields (for the same value of *n* as this parameter) are missing or are not valid, then this parameter MUST be ignored by the server. For example, if *n*="1", then *FieldPicker1*, *OperatorPicker1*, and *CompareWithValue1* MUST all be present and valid or else they MUST all be ignored by the server. If *IsThereAQuery* is set to "FALSE" or is not present, this parameter MUST be ignored by the server. The values of *n* MUST be consecutive. For *n*>1, if NextIsAnd*(n-1)* is not present, this parameter MUST be ignored by the server. If the *LocalizedTodayString* parameter is used in the value of this parameter, it can be followed by a negative or positive offset value. For example, if *LocalizedTodayString* is "[Today]" then the value of the value of the *CompareWithValue* parameter can be "[Today]", "[Today]+*x*", or "[Today]-*x*" where *x* is an integer. If the value of this parameter is set to "[ID]" then the field (2) designated by the corresponding FieldPicker*n* parameter is compared with the list item identifier. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the CompareWithValue*n* parameter MUST be ignored by the server.

NextIsAnd*n*: A **Boolean** value that specifies whether the **Boolean** operator that relates the n^{th} and $(n+1)^{th}$ filtered fields is an AND. If *IsThereAQuery* is set to "FALSE" or is not present, this parameter MUST be ignored by the server. The values of *n* MUST be consecutive. For example, if five NextIsAnd*n* parameters are sent to the server with their respective *n*'s set to 1, 2, 3, 5, 6; 5 and 6 MUST be ignored by the server. If this parameter is present for a given *n* then FieldPicker*(n+1)*, OperatorPicker*(n+1)*, and CompareWithValue*(n+1)* MUST be present, or else NextIsAnd*n* MUST be ignored. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the NextIsAnd*n* parameter MUST be ignored by the server. The NextIsAnd*n* parameter MUST be set to one of the following values:

Value	Description
TRUE	The Boolean operator relating the two filters is AND

Value	Description
FALSE	The Boolean operator relating the two filters is OR

HiddenFilter: A string. Specifies a hidden restriction (1) on the items shown in the view, as specified in [MS-WSSCAML] section 2.3.2.17. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the *HiddenFilter* parameter MUST be ignored by the server.

LocalizedTodayString: A string. Specifies the string that will be replaced with today's date if used in CompareWithValue*n*. For example, if this parameter is set to "[Today]" then all instances of "[Today]" in any CompareWithValue*n* parameters MUST be replaced with today's date. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the LocalizedTodayString parameter MUST be ignored by the server.

LocalizedMeString: A string. Specifies the string that will be replaced with the current user identifier if used in CompareWithValue*n*. For example, if this parameter is set to "[Me]" then all instances of "[Me]" in any CompareWithValue*n* parameters MUST be replaced with the current user identifier. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the *LocalizedMeString* parameter MUST be ignored by the server.

GroupLimit: An integer that MUST be between "1" and "2147483647". If the *GroupLimit* parameter is outside of that range, then the server MUST ignore it and assume "2147483647".The *GroupLimit* parameter sets the **GroupLimit** property of the **GroupByDefinition** type, as specified in [MS-WSSCAML] section 2.2.2.5.2. If there is not at least one valid GroupField*n* parameter, the *GroupLimit* parameter MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the *GroupLimit* parameter MUST be ignored by the server.

OverrideOrderBy: A **Boolean** value that sets the **Override** property of the **OrderByDefinition** type, as specified in [MS-WSSCAML] section 2.2.2.3.2. If there is not at least one valid SortField*n* parameter, the *OverrideOrderBy* parameter MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the *OverrideOrderBy* parameter MUST be ignored by the server.

RowLimit: An integer value that sets the **RowLimit** property of the **ViewDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.17.3. The value of the *RowLimit* parameter MUST be between "1" and "2147483647". If it is outside of that range, then the server MUST ignore the value provided and assume "2147483647". If the *ViewType* parameter value is set to "Calendar" the *RowLimit* parameter MUST be ignored by the server (the attribute is set to "0" (zero) in the generated Collaborative Application Markup Language (CAML). If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the *RowLimit* parameter MUST be ignored by the server.

Paged: A **Boolean** value that sets the **Paged** property of the **RowLimitDefinition** type, as specified in [MS-WSSCAML] section 2.3.2.14. If the *ViewType* parameter value is set to "Calendar" then the *Paged* parameter MUST be ignored by the server. If the *ViewFPModified* or *ReadOnly* parameter values are set to "TRUE", then the *Paged* parameter MUST be ignored by the server.

TabularView: A **Boolean** value. When the *TabularView* parameter value is set to "TRUE", the view MAY<33> display a checkbox next to each list item in the view, and enable operations to be performed on all the selected items.

InlineEdit: A **Boolean** value. When *InlineEdit* is set to "TRUE", the view SHOULD<34> enable the user to edit the list items in the current view, without having to navigate to a different page.

GanttPredecessors: A string that is only used if the *ViewType* parameter value is set to "Gantt". This parameter specifies the name of the field (2) that contains the task that SHOULD<35> be finished prior to starting the task in the current view item.

MethodXml: A string that SHOULD<36> contain the XML fragment for the view method. The MethodXml schema of this fragment is specified in section 2.2.5.2.

MobileItemLimit: An integer. The value MUST be between 1 and 99. If it is outside of that range, the server MUST ignore it and assume the value is 99. This parameter specifies the number of items to display if *MobileView* parameter value is "TRUE". The server SHOULD<37> ignore this value if the *MobileView* parameter of this view is "FALSE".

3.1.4.25.1 Return Values

If the **UpdateView** method is called through the **DisplayPost** method, it MUST return "0" (zero) if it is successful or a negative value if an error occurs.

If the **UpdateView** method is not called through the **DisplayPost** method, it MUST send an HTTP response to the protocol client and that response MUST be one of the following values:

Value	Description
302	Success. The redirection URL MUST be the URL of the view that is being updated.
401	Security not validated. The credentials that were supplied by the user are not valid.
404	Error.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 Create a New List

In this example, a protocol client creates a new list named "Activity", by using the following command:

```
http://server/_vti_bin/owssvr.dll?Cmd=NEWLIST&ListTemplate=104&Title=Activity
```

4.2 Delete a List

If the GUID for the list is "D6ACBFE8-5BC9-4269-9CB9-B45586365DDB" a protocol client deletes the "Activity" announcement list by using the following command:

```
http://server/_vti_bin/owssvr.dll?Cmd=DELETELIST&List=D6ACBFE8-5BC9-4269-9CB9-B45586365DDB
```

4.3 Add a Field to a List

In this example, a protocol client adds a new, integer type field named "MyNumber" to a list. The new field is to be required and added to the default view of the list.

The name of the server is "server". "{90dc847d-d805-44f1-826e-d42e8dc6d798}" is the GUID of the list. This example has been URL-encoded.

```
http://server/_vti_bin/owssvr.dll?Cmd=NEWFIELD&List={90dc847d-d805-44f1-826e-
d42e8dc6d798}&AddToDefaultView=ON&FieldXML=%3CField%20Name%3D%22MyNumber%22%20DisplayName%3D%
22MyNumber%22%20Description%3D%22The%20new%20field%22%20Type%3D%22Integer%22%20Required%3D%22
TRUE%22%2F%3E
```

4.4 Delete a Field from a List

To delete the field named "MyNumber" from a list, the protocol client uses the following command, where "server" is the name of the server and "{90dc847d-d805-44f1-826e-d42e8dc6d798}" is the GUID of the list.

```
http://server/_vti_bin/owssvr.dll?Cmd=DELETEFIELD&List={90dc847d-d805-44f1-826e-
d42e8dc6d798}&Field=MyNumber
```

4.5 Reorder Fields in a List

In this example, a protocol client switches the **Title** and **Body** fields in a newly created list. The protocol client uses the **REORDERFIELDS** action through the **DisplayPost** operation. The name of the server is "server". The GUID of the list is "{90dc847d-d805-44f1-826e-d42e8dc6d798}".

```
http://server/_vti_bin/owssvr.dll?Cmd=DisplayPost
```

The POST body:

```
<Method ID="REORDERFIELDS,0">
  <SetList Scope="Request">{90dc847d-d805-44f1-826e-d42e8dc6d798}</SetList>
```

```
    <SetVar Name="Cmd">REORDERFIELDS</SetVar>
    <SetVar Name="ReorderedFields">
      %3cFields%3e
        %3cField+Name%3d%22ID%22/%3e
        %3cField+Name%3d%22ContentTypeId%22/%3e
        %3cField+Name%3d%22ContentType%22/%3e
        %3cField+Name%3d%22Body%22/%3e
        %3cField+Name%3d%22Modified%22/%3e
        %3cField+Name%3d%22Created%22/%3e
        %3cField+Name%3d%22Author%22/%3e
        %3cField+Name%3d%22Editor%22/%3e
        %3cField+Name%3d%22_HasCopyDestinations%22/%3e
        %3cField+Name%3d%22_CopySource%22/%3e
        %3cField+Name%3d%22owshiddenversion%22/%3e
        %3cField+Name%3d%22WorkflowVersion%22/%3e
        %3cField+Name%3d%22_UIVersion%22/%3e
        %3cField+Name%3d%22_UIVersionString%22/%3e
        %3cField+Name%3d%22Attachments%22/%3e
        %3cField+Name%3d%22_ModerationStatus%22/%3e
        %3cField+Name%3d%22_ModerationComments%22/%3e
        %3cField+Name%3d%22Edit%22/%3e
        %3cField+Name%3d%22LinkTitleNoMenu%22/%3e
        %3cField+Name%3d%22LinkTitle%22/%3e
        %3cField+Name%3d%22SelectTitle%22/%3e
        %3cField+Name%3d%22InstanceID%22/%3e
        %3cField+Name%3d%22Order%22/%3e
        %3cField+Name%3d%22GUID%22/%3e
        %3cField+Name%3d%22WorkflowInstanceID%22/%3e
        %3cField+Name%3d%22FileRef%22/%3e
        %3cField+Name%3d%22FileDirRef%22/%3e
        %3cField+Name%3d%22Last_x0020_Modified%22/%3e
        %3cField+Name%3d%22Created_x0020_Date%22/%3e
        %3cField+Name%3d%22FSObjType%22/%3e
        %3cField+Name%3d%22PermMask%22/%3e
        %3cField+Name%3d%22FileLeafRef%22/%3e
        %3cField+Name%3d%22UniqueId%22/%3e
        %3cField+Name%3d%22ProgId%22/%3e
        %3cField+Name%3d%22ScopeId%22/%3e
        %3cField+Name%3d%22File_x0020_Type%22/%3e
        %3cField+Name%3d%22HTML_x0020_File_x0020_Type%22/%3e
        %3cField+Name%3d%22_EditMenuTableStart%22/%3e
        %3cField+Name%3d%22_EditMenuTableEnd%22/%3e
        %3cField+Name%3d%22LinkFilenameNoMenu%22/%3e
        %3cField+Name%3d%22LinkFilename%22/%3e
        %3cField+Name%3d%22DocIcon%22/%3e
        %3cField+Name%3d%22ServerUrl%22/%3e
        %3cField+Name%3d%22EncodedAbsUrl%22/%3e
        %3cField+Name%3d%22BaseName%22/%3e
        %3cField+Name%3d%22MetaInfo%22/%3e
        %3cField+Name%3d%22_Level%22/%3e
        %3cField+Name%3d%22_IsCurrentVersion%22/%3e
        %3cField+Name%3d%22Title%22/%3e
        %3cField+Name%3d%22Expires%22/%3e
      %3c/Fields%3e
    </SetVar>
  </Method>
```

4.6 Modify Properties of a List

In this example, a protocol client changes the title of a list to "Aloha". The name of the server is "server". The GUID of the list is "{90dc847d-d805-44f1-826e-d42e8dc6d798}".

```
http://server/_vti_bin/owssvr.dll?Cmd=MODLISTSETTINGS&List={90dc847d-d805-44f1-826e-
d42e8dc6d798}&NewListTitle=Aloha
```

4.7 Delete a View of a List

To delete a specific view from a specific list, a protocol client can construct a request as follows by using the POST method.

```
http://server/_vti_bin/owssvr.dll?Cmd=DeleteView&List={37D04126-3773-4adf-AC59-
328A1382BD1F}&View={863ABF34-5F93-415b-B27C-4066B3B73F09}
```

The protocol server then deletes the view described n the *View* parameter from the list designated by the *List* parameter, and redirects the protocol client to the default list view.

4.8 Create a New View of a List

To create a new view of a specific list, a protocol client can construct an HTTP POST request as follows. The view shows only those items that are assigned to current user:

```
http://server/_vti_bin/owssvr.dll?Cmd=NewView&List={37D04126-3773-4adf-AC59-
328A1382BD1F}&Personal=FALSE&ViewType=HTML&NewViewName=My
Items&ViewOrder0=1_Title&ShouldDisplayTitle=TRUE&SortField1=tp_Title&SortAscending1=TRUE&IsTh
ereAQuery=TRUE&RowLimit=25&Paged=TRUE&LocalizedMeString=[Me]&FieldPicker1=Owner&OperatorPicke
r1=Eq&CompareWithValue1=[Me]
```

The protocol server then creates a view with the **NewView** attribute and redirects the protocol client to the view URL.

4.9 Update an Existing View of a List

In this example, a protocol client sends the following HTTP POST request to update the attributes of a specific view:

```
http://server/_vti_bin/owssvr.dll?Cmd=UpdateView&List=
{37D04126-3773-4adf-AC59-328A1382BD1F}&View=
{863ABF34-5F93-415B-B27C-4066B3B73F09}&Personal=FALSE&ViewType=
HTML&NewViewName=My Items&NewViewFileName=myitems.aspx&ViewOrder0=
1_Title&ShouldDisplayTitle=TRUE&SortField1=tp_Title&SortAscending1=
TRUE&IsThereAQuery=TRUE&RowLimit=25&Paged=TRUE&LocalizedMeString=
[Me]&FieldPicker1=Owner&OperatorPicker1=Eq&CompareWithValue1=[Me]
```

The protocol server then edits the attributes of the view that is described by the *View* parameter, and redirects the protocol client to the URL of the edited view.

4.10 Method XML Fragment

A protocol client can describe a **Method XML Fragment** as part of a view query to filter results from the data source (1) for the list (1). In the following example, a data source (1) for a list returns a list of products. The **Method XML Fragment** is as follows:

```
<Method Name="ProductFinderInstance">  <Filter Name="NameFilter" Value="LL%"/></Method>
```

When rendering the view, a **Finder** with the name "ProductFinderInstance" is found on the **Entity** described by the data source (1) for the list. If a **FilterDescriptor** named "NameFilter" is found on the Finder, then its value is set to "LL%" when the view is rendered.

It is up to the implementation of the data source (1) for the list how to interpret the value "LL%". However, in this example it is assumed that NameFilter interprets this value to mean only products whose name begins with "LL" will be returned.

5 Security

5.1 Security Considerations for Implementers

It is possible for a malicious user to lure another user to a malicious page, for example, by sending a URL for a malicious page in an e-mail message. When the user visits the malicious page, that page can perform a silent POST to the protocol server. Because this protocol uses HTTP POST, the malicious user can lure the user into performing any operation that is used by this protocol against any protocol server that implements this protocol. This type of attack is referred to as a "one-click attack."

To prevent this, protocol servers that implement this protocol can require that all incoming requests have the HTTP header value "X-Vermeer-Content-Type". Because Web browsers typically do not send this header value, requiring the value helps prevent users from browsing to a page that calls a silent method that is used by this protocol. It is strongly recommended that all implementations of this protocol require inclusion of this HTTP header value to help prevent one-click attacks.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

None.

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:

- The 2007 Microsoft® Office system
- Microsoft® Office 2010 suites
- Microsoft® Office SharePoint® Designer 2007
- Microsoft® SharePoint® Foundation 2010
- Windows® SharePoint® Services 3.0
- Microsoft® SharePoint® Foundation 2013

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 2.2.3.1: Windows SharePoint Services 3.0, Windows SharePoint Services 3.0 SP 1 and SharePoint Foundation 2010 all require the version to be greater than 6.0.0.3630. sppt2007 and sppt2010 implementations will only fail the first method of the requests if **OnError** is set to "Continue".

<2> Section 3.1.4: This parameter is not used by SharePoint Foundation 2010, but is used in Windows SharePoint Services 3.0 and Windows SharePoint Services 2.0.

<3> Section 3.1.4.1: This command is not supported by SharePoint Foundation 2010.

<4> Section 3.1.4.1: In Office 2007 SP1, the *URL* parameter occurs directly before the *ACT* parameter in the query string.

<5> Section 3.1.4.1.1: In Office 2007 SP1, the value of the *Version* parameter in the **Common Response** header is "6219".

<6> Section 3.1.4.1.1: If a failure occurs, Office 2007 SP1 returns a second, identical **Common Response** header.

<7> Section 3.1.4.1.3.2: These values are optional in SharePoint Foundation 2010, Windows SharePoint Services 3.0, and Windows SharePoint Services 2.0.

<8> Section 3.1.4.10: Any value that is not described in this section is ignored by SharePoint Foundation 2010 and all released versions of Windows SharePoint Services 3.0, except the first release of Windows SharePoint Services 3.0.

<9> Section 3.1.4.12.1: Windows SharePoint Services 3.0 SP 1 does not respond with the hit counter image when the page parameter is missing or is not valid, when neither the image nor custom parameter is present, or when the custom parameter is not valid.

<10> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010.

<11> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010

<12> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010.

<13> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010.

<14> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010.

<15> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010.

<16> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010.

<17> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010

<18> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010

<19> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010

<20> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010.

<21> Section 3.1.4.13: This command is used only by SharePoint Foundation 2010. SharePoint Foundation 2010 uses a configurable list of unsafe file types on the server.

<22> Section 3.1.4.15: If set to "-3", Windows SharePoint Services 3.0 SP 1 deletes the source occurrence.

<23> Section 3.1.4.17: word97 document template

<24> Section 3.1.4.17: xl97 document template

<25> Section 3.1.4.17: ppt97 document template

<26> Section 3.1.4.17: This command is used only by SharePoint Foundation 2010.

<27> Section 3.1.4.17: This command is used only by SharePoint Foundation 2010.

<28> Section 3.1.4.18: This command is used only by SharePoint Foundation 2010.

<29> Section 3.1.4.18: This command is used only by SharePoint Foundation 2010.

<30> Section 3.1.4.18: This command is used only by SharePoint Foundation 2010.

<31> Section 3.1.4.18: This command is used only by SharePoint Foundation 2010.

<32> Section 3.1.4.18: This command is used only by SharePoint Foundation 2010.

<33> Section 3.1.4.25: This command is used only by SharePoint Foundation 2010.

<34> Section 3.1.4.25: This command is used only by SharePoint Foundation 2010.

<35> Section 3.1.4.25: This command is used only by SharePoint Foundation 2010.

<36> Section 3.1.4.25: This command is used only by SharePoint Foundation 2010.

<37> Section 3.1.4.25: This command is used only by SharePoint Foundation 2010.

8 Change Tracking

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.9.1.1 Usage Data Binary Field Structure	Added new section.	Y	New content added.
2.2.9.1.1.1 Usage Data Header Structure	Added new section.	Y	New content added.
2.2.9.1.1.2 Usage Record Structure	Added new section.	Y	New content added.
	Added definition number for term "field" where it appears throughout document.	N	Content updated.

9 Index

P

R

S

T

U

V

W