[MS-OOWQWS]:
Office Online Web Query Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Minor	Updated the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	1.5	Minor	Clarified the meaning of the technical content.
04/11/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2013	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
07/30/2013	1.5	No change	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
			the technical content.
11/18/2013	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
02/10/2014	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
04/30/2014	1.5	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Office Online Web Query Web Service Protocol. This protocol enumerates details about files in the Office Online repository. This protocol enables the user to traverse categories in which the files are organized and query details about the categories, along with the subcategories and files organized under them.

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]:

XML namespace

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

category
file
HTTP GET
profile site
thumbnail
Uniform Resource Locator (URL)
XML element
XML namespace prefix

The following terms are specific to this document:

asset identifier: A unique string that is used to identify a file or a category.

provider category: An integer that specifies whether a file was published by Microsoft Corporation.

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 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.

[ISO-3166] International Organization for Standardization, "Codes for the Representation of Names of Countries and Their Subdivisions", ISO 3166, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24591

Note There is a charge to download the specification.

[ISO-639] International Organization for Standardization, "Codes for the Representation of Names of Languages", ISO 639, http://www.loc.gov/standards/iso639-2/

[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

[RFC3023] Murata, M., St.Laurent, S., and Kohn, D., "XML Media Types", RFC 3023, January 2001, http://www.ietf.org/rfc/rfc3023.txt

[RFC4646] A. Phillips, Ed., and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006, http://www.ietf.org/rfc/rfc4646.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., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., and Malhotra, A., Eds., "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".

1.3 Protocol Overview (Synopsis)

The Office Online Web Query Web Service Protocol allows applications to query the Office Online server for details about the **files** that are either provided by Microsoft Corporation or by other parties. These files are organized into categories in a tree-like structure. Each category and each file is uniquely identified using an **asset identifier**. For further details about the data model refer to section 3.1.1.

The protocol allows the user to traverse through the tree by enumerating the contents of each node. The user has the flexibility to query for only subcategories, files (leaf nodes), or both subcategories and files within a given node. The user can also filter the results of a query based on the **provider category** associated with the provider of the file.

The following sequence diagram indicates a basic example of identifying the root category and then enumerating through the contents of its subcategory.

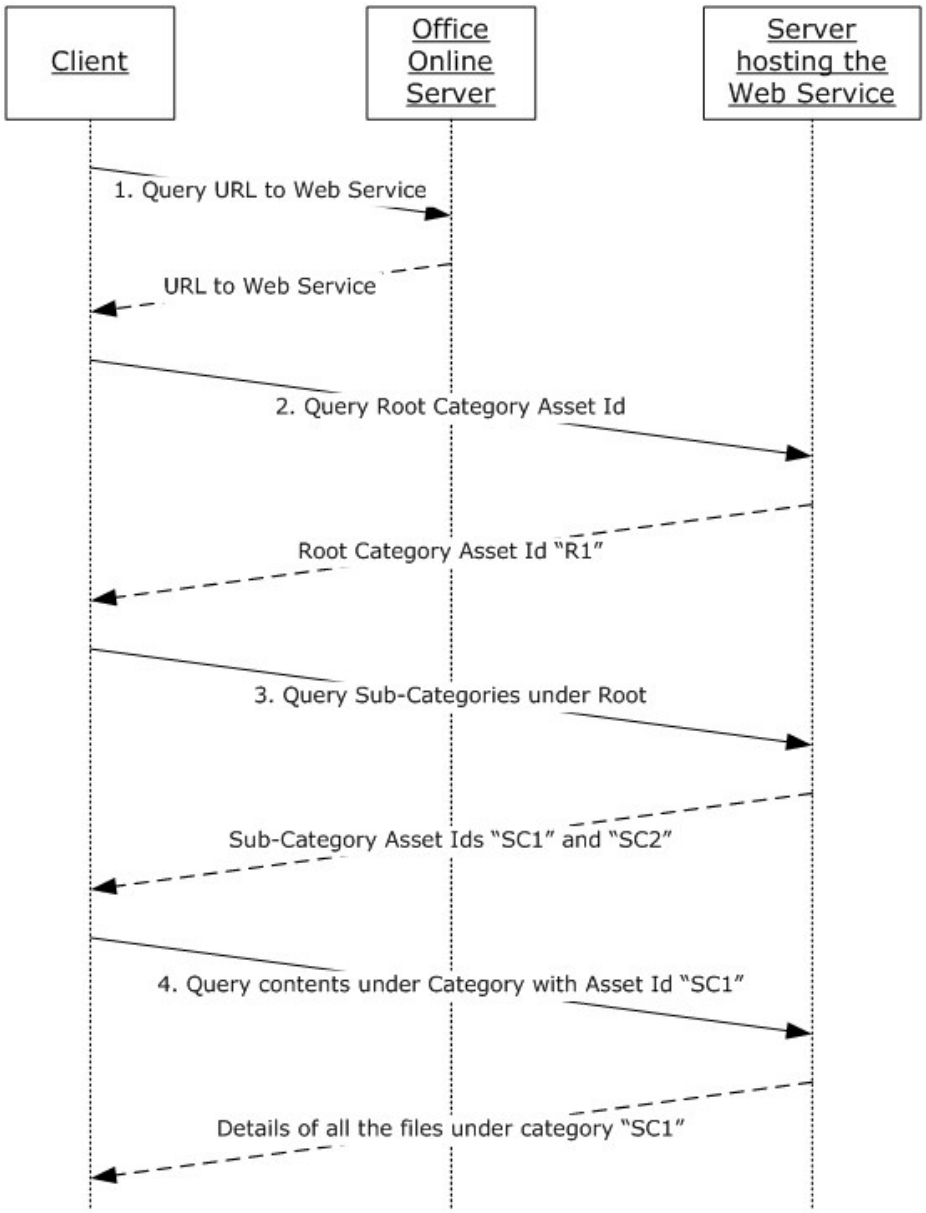

Figure 1: Sequence diagram indicating the common sequence of steps involved in the protocol

1. The client queries the Office Online server to determine the web service URL that exposes the necessary functionality to query details about the files.

2. The client queries for the asset identifier of the root node so that it can traverse the through the tree structure of files.

3. The client queries for the sub-categories under the given category (that is, the root category).

4. The client queries for the files contained in a specific sub-category.

The protocol is implemented as a sequence of **HTTP GET** method calls that accept a set of parameters and returns an XML string (refer to section 2.2.3.2 for details) as an HTTP response message as described in [RFC3023]. The client sends method call requests to the server and the server sends return values to the client as an XML string. The server never initiates any communication with the client. All communication is transported over HTTP, as described in [RFC2616] section 9.

1.4 Relationship to Other Protocols

The Office Online Web Query Web Service Protocol works by transmitting messages using the HTTP protocol as described in [RFC2616].

The following diagram shows the underlying messaging and transport stack that the protocol uses:

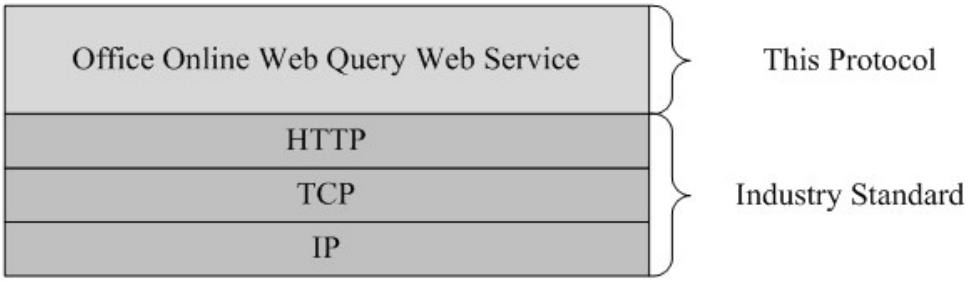

Figure 2: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

The protocol client is expected to know the **URL** of the server it wants to communicate with. This protocol assumes that authentication has been performed by the underlying protocols.

1.6 Applicability Statement

The Office Online Web Query Web Service Protocol is applicable in the following scenarios:

- Navigating through the tree structure consisting of **categories**, sub-categories and files contained in the categories.
- Retrieving the details about the files under a category.

1.7 Versioning and Capability Negotiation

None.

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].

Client requests to the server MUST be transmitted as HTTP GET methods with appropriate query string parameters appended to the request URL. For details about the syntax, see section 2.2.1. If an error is encountered while processing the client request, the server SHOULD<1> return an appropriate HTTP status code (as defined in [RFC2616] section 10) and stop parsing the request.

Server responses to client requests MUST be transmitted as HTTP response messages. The data in the response MUST be in XML, as specified in [RFC3023]. If the server response does not conform to the message syntax, the client MUST ignore the server response and stop communication with the server.

2.2 Message Syntax

This section specifies the syntax and data types that are used when the client requests messages from the server and when the server responds to client requests.

2.2.1 Client Request Syntax

To query category or file details from the Office Online Web Query web service, the client MUST send an HTTP GET request to the server. The URL for the request MUST be constituted by appending the required query string to the web service URL.

The query string MUST be a set of "name=value" pairs separated by the ampersand (&). The query string format is illustrated as follows:

```
<param1-name>=<param1-value>&<param2-name>=<param2-value>...,
```

Where:

- <*param1-name*> represents the name of the first parameter.
- <*param1-value*> represents the value of the first parameter.
- <*param2-name*> represents the name of the second parameter.
- <*param2-value*> represents the value of the second parameter.

For details about the supported parameters refer to section 2.2.3.1.

2.2.2 Server Response Syntax

All server responses MUST be transmitted as HTTP response messages. The data in response to the client query MUST be transmitted as an XML string. Section 2.2.3.2 specifies the XML schema for the transmitted XML.

2.2.3 Data Types

2.2.3.1 Request Message Parameters

The client can request specific information from the server through the HTTP GET method by appending one or more of the following parameters as a query string to the URL:

- **cid:** The asset identifier of the category to browse or query. This parameter MUST only be specified for queries in which details about a category or file are sought.
- **lc:** Specifies the locale in which the search is to be made. The value MUST be valid culture name as specified in [RFC4646]. Typically, the value is a combination of an ISO 639 two-letter lowercase culture code associated with a language (as specified in [ISO-639]) and an ISO 3166 two-letter uppercase subculture code associated with a country or region (as specified in [ISO-3166]).
- **tl:** Integer specifying the provider category of the files to be returned. Possible values are described in the following table.

Value	Description
1	Only files that are provided by Microsoft Corporation.
2	Files that are provided by Microsoft Corporation and files provided by trusted community users.
3	Files that are provided by Microsoft Corporation, files provided by trusted community users and files provided by new community users that have not become trusted yet.

- **type:** Integer specifying the type of the query. The value passed for this parameter MUST be one of those described in the following table.

Value	Description
1	Queries only the sub-categories in a given category.
2	Queries only the files (leaf nodes) in a given category.
3	Queries both the sub-categories and the files (leaf nodes) in a given category.
5	Queries only the root category.

- **max:** Integer specifying the maximum number of files to be returned. The value MAY be specified to limit the number of files returned. The server MUST default to returning a maximum of 250 files if the value is not specified.

The same parameter MUST NOT be repeated in the client request. The server MUST treat all the parameters in the client request in a case-insensitive manner. That is, parameters differing only by case MUST be treated as identical.

2.2.3.2 Response Message XML

This section specifies the XML definitions used by this protocol. The syntax of the definitions uses the XML Schema as defined in [XMLSCHEMA1] and [XMLSCHEMA2].

2.2.3.2.1 Namespaces

This protocol specifies and references **XML namespaces** using the mechanisms specified in [XMLNS]. Although this document associates an 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. These namespaces are described in the following table.

Alias (prefix)	Namespace URI/URN	Reference
o	urn:schemas-microsoft-com:office:office	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]

2.2.3.2.2 Elements

The following table summarizes the set of **XML element** definitions in this specification.

Element	Description
AppConfig	Specifies the URL to the Office Online Web Query Web service against which all subsequent calls need to be made.
results	Specifies details about the queried nodes in the tree structure containing the files.

2.2.3.2.2.1 AppConfig Element

This element is returned by the server when the client requests the URL to the Office Online Web Query web service. The structure of this element MUST adhere to the following schema definition:

```
<xs:element name="AppConfig">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="URLs">
         <xs:complexType>
           <xs:sequence>
             <xs:element name="SharepointTCQuery14" type="xs:string" minOccurs="1"
maxOccurs="1" />
             <xs:element name="AwsTcQueryWac14" type="xs:string" minOccurs="1" maxOccurs="1"
/>
             <xs:element name="AwsTrRatingWac14" type="xs:string" minOccurs="1" maxOccurs="1"
/>
             <xs:element name="AwsTcQueryMac14" type="xs:string" minOccurs="1" maxOccurs="1"
/>
           </xs:sequence>
         </xs:complexType>
       </xs:element>
     </xs:sequence>
     <xs:attribute name="Version" type="xs:string" use="required" />
     <xs:attribute name="ExpireMinutes" type="xs:unsignedShort" use="required" />
     <xs:attribute name="GraceMinutes" type="xs:unsignedShort" use="required" />
     <xs:attribute name="GenerationDate" type="xs:string" use="required" />
     <xs:attribute name="GenerationTime" type="xs:string" use="required" />
   </xs:complexType>
</xs:element>
```

URLs.SharepointTCQuery14: Specifies the URL to the web service exposing the functionalities to query details about the files and browse the tree-structure in which they are organized.

URLs.AwsTcQueryWac14: This element MUST be ignored.

URLs.AwsTrRatingWac14: This element MUST be ignored.

URLs.AwsTcQueryMac14: This element MUST be ignored.

Version: This attribute MUST be ignored.

ExpireMinutes: This attribute MUST be ignored.

GraceMinutes: This attribute MUST be ignored.

GenerationDate: This attribute MUST be ignored.

GenerationTime: This attribute MUST be ignored.

2.2.3.2.2.2 results Element

This element is returned by the server in response to the client's request to obtain details about the nodes in the tree structure containing the Office Online files (organized as described in section 3.1.1). The structure of this element MUST adhere to the following schema definition:

```
<xs:element name="results">
   <xs:complexType>
     <xs:sequence>
       <xs:element maxOccurs="1" minOccurs="1" name="hdr">
         <xs:complexType>
           <xs:attribute ref="o:key" use="required" />
         </xs:complexType>
       </xs:element>
       <xs:element maxOccurs="unbounded" minOccurs="0" name="ct" type="o:CategoryDetails" />
       <xs:element maxOccurs="unbounded" minOccurs="0" name="tc" type="o:TemplateDetails" />
     </xs:sequence>
   </xs:complexType>
</xs:element>
```

hdr.key: See section 2.2.3.2.5.7.

ct: See section 2.2.3.2.3.1.

tc: See section 2.2.3.2.3.2.

2.2.3.2.3 Complex Types

The following table summarizes the set of common XML schema complex type definitions defined by this specification.

Complex type	Description
CategoryDetails	Specifies details about a category.
TemplateDetails	Specifies details about a file.

2.2.3.2.3.1 CategoryDetails ComplexType

Contains a set of XML attributes that describe details of a category. The following XML schema defines the structure of the complex type:

```
<xs:complexType name="CategoryDetails">
   <xs:attribute ref="o:assetid" use="required" />
   <xs:attribute ref="o:title" use="required" />
</xs:complexType>
```

assetid: Refer to section 2.2.3.2.5.1.

title: Refer to section 2.2.3.2.5.17.

2.2.3.2.3.2 TemplateDetails ComplexType

Contains a set of XML attributes that describe the details of a file. The following XML schema defines the structure of the complex type:

```
<xs:complexType name="TemplateDetails">
   <xs:attribute ref="o:assetid" use="required" />
   <xs:attribute ref="o:url" use="required" />
   <xs:attribute ref="o:title" use="required" />
   <xs:attribute ref="o:lmod" use="required" />
   <xs:attribute ref="o:cdate" use="required" />
   <xs:attribute ref="o:size" use="required" />
   <xs:attribute ref="o:rat" use="required" />
   <xs:attribute ref="o:votes" use="required" />
   <xs:attribute ref="o:prov" use="required" />
   <xs:attribute ref="o:dls" use="required" />
   <xs:attribute ref="o:imgurl" use="required" />
   <xs:attribute ref="o:tnurl" use="required" />
   <xs:attribute ref="o:provurl" use="required" />
   <xs:attribute ref="o:desc" use="required" />
   <xs:attribute ref="o:fsize" use="required" />
   <xs:attribute ref="o:tl" use="required" />
   <xs:attribute ref="o:logurl" use="required" />
   <xs:attribute ref="o:propbag" use="required" />
   <xs:attribute ref="o:suplvl" use="required" />
   <xs:attribute ref="o:eulaurl" use="required" />
</xs:complexType>
```

assetid: See section 2.2.3.2.5.1.

url: See section 2.2.3.2.5.20.

title: See section 2.2.3.2.5.17.

lmod: See section 2.2.3.2.5.9.

cdate: See section 2.2.3.2.5.2.

size: See section 2.2.3.2.5.15.

rat: See section 2.2.3.2.5.14.

votes: See section 2.2.3.2.5.21.

prov: See section 2.2.3.2.5.12.

dls: See section 2.2.3.2.5.4.

imgurl: See section 2.2.3.2.5.8.

tnurl: See section 2.2.3.2.5.19.

provurl: See section 2.2.3.2.5.13.

desc: See section 2.2.3.2.5.3.

fsize: See section 2.2.3.2.5.6.

tl: See section 2.2.3.2.5.18.

logurl: See section 2.2.3.2.5.10.

propbag: See section 2.2.3.2.5.11.

suplvl: See section 2.2.3.2.5.16.

eulaurl: See section 2.2.3.2.5.5.

2.2.3.2.4 Simple Types

None.

2.2.3.2.5 Attributes

The following table summarizes the set of common XML schema attribute definitions defined by this specification.

Attribute	Description
assetid	Specifies the asset identifier for a category or file.
cdate	This attribute MUST be ignored.
desc	Specifies the description for the file.
dls	Specifies the number of times the file has been downloaded.
eulaurl	Specifies the URL for the end user license agreement for the file.
fsize	Specifies the size of the file along with the time needed for download.
key	Specifies the asset identifier of the object containing the queried data.
imgurl	Specifies the URL to the preview image of the file or category.
lmod	This attribute MUST be ignored.
logurl	Specifies the URL that can be used to log an event about the usage of the file.
propbag	Specifies an XML string that defines additional properties about the file.

Attribute	Description
prov	Specifies the name of the user or company who has provided the file.
provurl	Specifies the URL for the provider of the file.
rat	Specifies the average ratings for the file.
size	Specifies the size of the file.
suplvl	Specifies the support level of the file.
title	Specifies the name of the file or category.
tl	Specifies the provider category associated with the file.
tnurl	Specifies the URL to the **thumbnail** image for the file.
url	Specifies the URL to the file.
votes	Specifies the total number of users who rated the file.

2.2.3.2.5.1 assetid Attribute

Specifies the asset identifier for a category or file.

```
<xs:attribute name="assetid" type="xs:string" />
```

2.2.3.2.5.2 cdate Attribute

This attribute MUST be ignored.

2.2.3.2.5.3 desc Attribute

Specifies the description for the file.

```
<xs:attribute name="desc" type="xs:string" />
```

2.2.3.2.5.4 dls Attribute

Specifies the number of times the file has been downloaded by users.

```
<xs:attribute name="dls" type="xs:unsignedInt" />
```

2.2.3.2.5.5 eulaurl Attribute

Specifies the URL to the end user license agreement for the file.

```
<xs:attribute name="eulaurl" type="xs:string" />
```

2.2.3.2.5.6 fsize Attribute

Specifies the size of the file along with the time needed to download it. The value is specified in the following format:

```
<size> <size-units> (<download-time> @ <connection-bandwidth>)
```

Where:

- <*size*> refers to the size of the file.
- <*size-units*> refers to the units used to measure the size. This will be kilobyte, megabyte, or gigabyte based on the size of the file.
- <*download-time*> refers to the time needed to download the file.
- <*connection-bandwidth*> refers to the bandwidth of the connection in Kbps.

A sample value is as follows: 564 kilobytes (1 min @ 56 Kbps).

```
<xs:attribute name="fsize" type="xs:string" />
```

2.2.3.2.5.7 key Attribute

Specifies the asset identifier of the object containing the queried data. If the data being queried contains categories or files, it specifies the asset identifier of the category containing the queried items. If the data being queried is the root category, it specifies the asset identifier of the object that contains the root node, and its value is not used anywhere in the protocol.

```
<xs:attribute name="key" type="xs:string" />
```

2.2.3.2.5.8 imgurl Attribute

Specifies the URL to the preview image for the file or category.

```
<xs:attribute name="imgurl" type="xs:string" />
```

2.2.3.2.5.9 lmod Attribute

This attribute MUST be ignored.

2.2.3.2.5.10 logurl Attribute

Specifies the URL against which an HTTP request can be placed to get the server to log an event about the usage of the associated file.

```
<xs:attribute name="logurl" type="xs:string" />
```

2.2.3.2.5.11 propbag Attribute

Specifies an XML string that defines additional properties about the file.

```
<xs:attribute name="propbag" type="xs:string" />
```

2.2.3.2.5.12 prov Attribute

Specifies the name of the user or company who has provided the file.

```
<xs:attribute name="prov" type="xs:string" />
```

2.2.3.2.5.13 provurl Attribute

Specifies the URL for the provider of the file. If the provider is Microsoft, the value will refer to the root of a site containing this file and if the provider is a user the value will refer to the user's **profile site**.

```
<xs:attribute name="provurl" type="xs:string" />
```

2.2.3.2.5.14 rat Attribute

Specifies the average ratings associated with the file on a scale of 0 to 100.

```
<xs:attribute name="rat" type="xs:unsignedByte" />
```

2.2.3.2.5.15 size Attribute

Specifies the size (in kilobytes) of the file.

```
<xs:attribute name="size" type="xs:unsignedInt" />
```

2.2.3.2.5.16 suplvl Attribute

Specifies the support level of the file.

```
<xs:attribute name="suplvl" type="xs:unsignedInt" />
```

The following table describes possible values.

Value	Description
1	Microsoft SharePoint Support
2	Microsoft Developer
100	No Support (Microsoft Corporation provided file.)
101	Community Platinum

Value	Description
102	Community Premium
103	Community Standard
104	Community Basic
105	Community Limited
200	No Support (Community user provided file.)

2.2.3.2.5.17 title Attribute

Specifies the name of the file or the category to which it is associated.

```
<xs:attribute name="title" type="xs:string" />
```

2.2.3.2.5.18 tl Attribute

Specifies the provider category associated with the file.

```
<xs:attribute name="tl" type="xs:unsignedByte" />
```

2.2.3.2.5.19 tnurl Attribute

Specifies the URL to the thumbnail image for the file.

```
<xs:attribute name="tnurl" type="xs:string" />
```

2.2.3.2.5.20 url Attribute

Specifies the URL to the file.

```
<xs:attribute name="url" type="xs:string" />
```

2.2.3.2.5.21 votes Attribute

Specifies the total number of users who rated the file.

```
<xs:attribute name="votes" type="xs:unsignedInt" />
```

2.2.3.2.6 Attribute Groups

None.

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.

Except where specified, protocol clients SHOULD interpret HTTP Status Codes returned by the protocol server as specified in [RFC2616] section 10.

The client is responsible for obtaining the necessary data from the server through HTTP GET requests. All connections MUST be initiated by the client. This protocol, like HTTP 1.1, as specified in [RFC2616], is a stateless protocol. As long as the client has the appropriate parameters (as described in section 2.2.3.1) to formulate the query to the server, it can make the call and retrieve the data. As such, connections do not need to be closed.

3.1 Server Details

3.1.1 Abstract Data Model

This protocol enables applications to query the Office Online server for files that are either provided by Microsoft Corporation or by partners. For convenience, files that serve a similar purpose or have a similar set of features are grouped together into a category. To identify the right set of files, the client would first identify its category and query for more details about that category. These functionalities are achieved through a conceptual data model that is structured as a tree as shown in the following diagram.

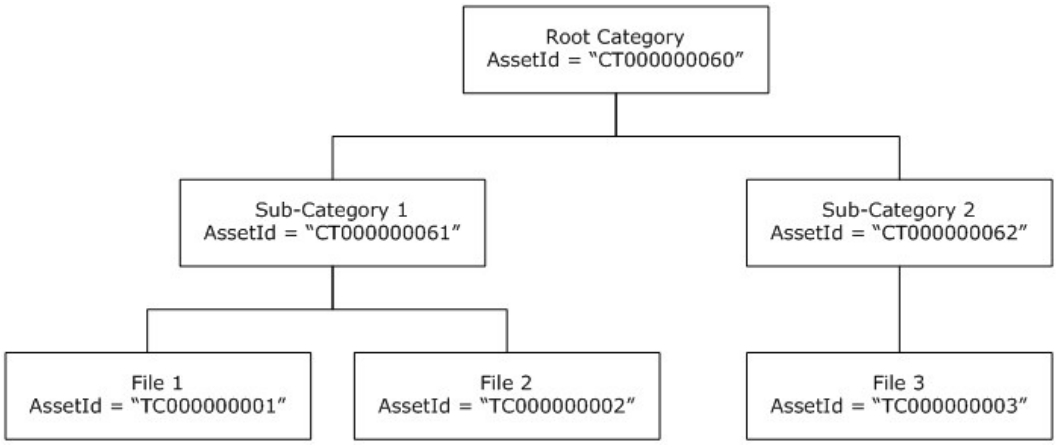

Figure 3: A sample tree structure containing a set of categories and files

This diagram illustrates how files are organized into categories to form a tree structure. The rules governing the structure of the model are as follows:

- There MUST be only one root node.
- Every node in the tree MUST have an asset identifier associated with it.
- The root node MUST be a category node.
- A category node in the tree MAY be empty.

- A category node MAY contain other category nodes, files or both.
- Every file MUST be the leaf node in the tree. That is, a node corresponding to a file MUST NOT contain sub-nodes.

3.1.2 Timers

None.

3.1.3 Initialization

The following operations are involved during the initialization of this protocol:

3.1.3.1 Determining Web service URL

All the queries to obtain information about the files and their categories provided by the Office Online server have to be made against the Office Online Web Query Web service. Hence, the client first needs to query the server for the URL to the Web service.

Consider "www.contoso.com" to be the URL to the server against which the requests are to be made. The following actions MUST be performed in the specified order to obtain the URL to the web service:

- The client MUST submit an HTTP GET request to the URL http://www.contoso.com/config14wc.
- The server MUST respond to this request with **AppConfig** XML element defined in section 2.2.3.2.2.1**.**
- The client MUST read the **SharepointTCQuery14** element from the AppConfig element to obtain the URL to the web service. The client MUST make all subsequent requests against this URL.

3.1.4 Message Processing Events and Sequencing Rules

Post initialization, the client MAY send any number of HTTP GET requests to the server to obtain the appropriate parameters (as described in section 2.2.3.1) about a category or file. If no asset identifier corresponding to a category is available with the client, the first logical step is to determine the root category as described in section 3.1.4.1. Once the root category asset identifier is available the client can traverse through the tree structure in a top to bottom manner by first requesting for the details of sub-categories (as described in section 3.1.4.2) and then requesting details about files in the categories of interest (as described in section 3.1.4.3).

By adjusting the parameters and their values in the HTTP GET request sent to the server, the client can obtain the necessary details about the files and categories contained in a specific category. For every valid client request, the server MUST respond with an XML string specifying the requested details. This section specifies the details of both the client requests and the server responses during the basic operations involved in this protocol. New operations can be defined by changing the parameter list in the client request.

3.1.4.1 Obtaining the Root Category

To obtain the root category the client MUST submit a valid request to the server passing in the appropriate parameters (as described in section 2.2.3.1) and the server MUST respond with an XML string specifying the requested details.

3.1.4.1.1 Messages

3.1.4.1.1.1 Request Message

The client MUST send an HTTP GET request against the web service URL with the following query string format:

```
lc=<locale>&type=5&tl=<provider-category>
```

Where:

- <*locale*> MUST be a valid value for the lc parameter.
- <*provider-category*> MUST be a valid value for the tl parameter.

The parameter name-value pair "type=5" MUST be specified to obtain the root category. The parameters in the query string MAY be defined in any order. For details about the possible values for parameters refer to section 2.2.3.1**.**

3.1.4.1.1.2 Response Message

The server response MUST be an HTTP response message containing the result XML string (described in section 2.2.3.2.2.2) with only an **hdr** element with a **key** attribute that describes the root category. The results element MUST NOT contain any **ct** or **tc** element, because details about a sub-category or file were not queried.

3.1.4.2 Obtaining the Subcategories under a Category

To obtain the subcategories under a given category the client MUST submit a valid request to the server passing in the appropriate parameters and the server MUST respond with an XML string specifying the requested details.

3.1.4.2.1 Messages

3.1.4.2.1.1 Request Message

The client MUST send an HTTP GET request to the web service URL with the following query string format:

```
lc=<locale>&type=<query-type>&tl=<provider-category>&cid=<assetid>
```

Where:

- <*locale*> MUST be a valid value for the lc parameter.
- <*query-type*> MUST be either 1 or 3.
- <*provider-category*> MUST be a valid value for the tl parameter.
- <*assetid*> MUST be the asset identifier of the category whose sub-categories are sought.

The parameters in the query string MAY be defined in any order. For details about the possible values for parameters refer to section 2.2.3.1.

3.1.4.2.1.2 Response Message

The server response MUST be an HTTP response message containing the result XML string (described in section 2.2.3.2.2.2) with one **ct** element for each sub-category in the given category. The results element MAY contain **tc** elements if the client has also queried for the details of files in the given category.

3.1.4.3 Obtaining the Files under a Category

To obtain the files under a given category the client MUST submit a valid request to the server passing in the appropriate parameters and the server MUST respond with an XML string specifying the requested details.

3.1.4.3.1 Messages

3.1.4.3.1.1 Request Message

The client MUST send an HTTP GET request against the web service URL with the following query string format:

```
lc=<locale>&type=<query-type>&tl=<provider-category> &cid=<assetid>
```

Where:

- <*locale*> MUST be a valid value for the lc parameter.
- <*query-type*> MUST be either 2 or 3.
- <*provider-category*> MUST be a valid value for the tl parameter.
- <*assetid*> MUST be the asset identifier of the category whose files are sought.

The parameters in the query string MAY be defined in any order. For details about the possible values for parameters refer to section 2.2.3.1.

3.1.4.3.1.2 Response Message

The server response MUST be an HTTP response message containing the result XML string (described in section 2.2.3.2.2.2) with one **tc** element for each file in the given category. The results element MAY contain **ct** elements if the client has also queried for the details of sub-categories in the given category.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 Query All Expense Report Files Provided by Microsoft from Office Online

Let "www.contoso.com" be the URL to the Office Online server that contains a set of files organized into different categories as shown in the tree in the following diagram.

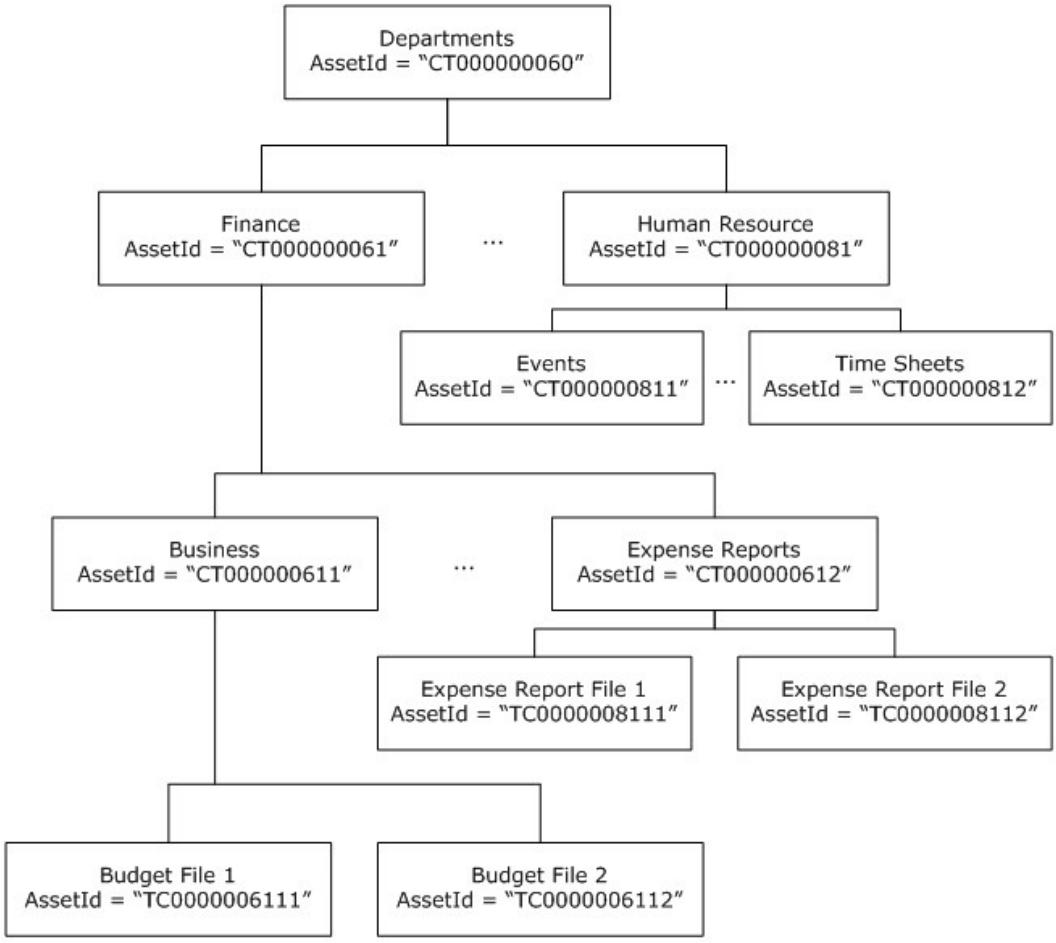

Figure 4: Files organized in a tree layout

The goal is to obtain details of all Expense Report files, which are organized under the Finance category and Expense Reports sub-category. The following sequence of steps describes one of the ways to obtain the details for the desired files:

1. **Determine the URL to the Office Online Web Query Web Service**.

   The client places an HTTP GET request to the URL http://www.contoso.com/config14wc.

   The server responds with a HTTP response message containing an XML string. The structure of the XML is as defined in section 2.2.3.2.2.1.

   The client reads the value of the **SharepointTCQuery14** element in the XML and obtains the following URL to the web service: http://contoso.com/SharepointTCQuery14.aspx.

2. **Determine the asset identifier for the Root category in the language desired by the client.**

   The client places an HTTP GET request to the web service URL with the query string "lc=en-US&tl=1&type=5". The complete URL (with the query string) is as follows:

   ```
   http://contoso.com/SharepointTCQuery14.aspx?lc=en-us&tl=1&type=5
   ```

   The server responds with a HTTP response message containing an XML string. The structure of the XML is as defined in section 2.2.3.2.2.2. The XML contains only an **hdr** element with a **key** attribute from which the client obtains the asset identifier "CT000000060" for the root node.

3. **Determine the asset identifier for the Finance category.**

The client places an HTTP GET request to the web service URL with the query string "lc=en-US&type=1&tl=1&cid=CT000000060&max=250". The complete URL (with the query string) is as follows:

```
http://contoso.com/SharepointTCQuery14.aspx?lc=en-US&type=1&tl=1&cid=CT000000060&max=250
```

The server responds with a HTTP response message containing an XML string. The structure of the XML is as defined in section 2.2.3.2.2.2. The client determines the **ct** element from the XML that has the title of "Finance" and then obtains the asset identifier "CT000000061" corresponding to the Finance category.

- **Determine the asset identifier for the Expense Reports sub-category.**

The client places an HTTP GET request to the web service URL with the query string "lc=en-US&type=1&tl=1&cid=CT000000061&max=250". The complete URL (with the query string) is as follows:

```
http://contoso.com/SharepointTCQuery14.aspx?lc=en-US&type=1&tl=1&cid=CT000000061&max=250
```

The server responds with a HTTP response message containing an XML string. The structure of the XML is as defined in section 2.2.3.2.2.2. The client determines the **ct** element from the XML that has the title of "Expense Reports" and then obtains the asset identifier "CT000000612" corresponding to the "Expense Reports" sub-category.

- **Determine all the expense report files.**

The client places an HTTP GET request to the web service URL with the query string "lc=en-US&type=2&tl=1&cid=CT000000612&max=250". The complete URL (with the query string) is as follows:

```
http://contoso.com/SharepointTCQuery14.aspx?lc=en-US&type=2&tl=1&cid=CT000000612&max=250
```

The server responds with a HTTP response message containing an XML string. The structure of the XML is as defined in section 2.2.3.2.2.2. The client obtains information regarding all the expense report files through the **tc** elements in the XML.

5 Security

5.1 Security Considerations for Implementers

This protocol introduces no additional security considerations beyond those applicable to its underlying protocols.

It is advisable that implementers take security precautions to ensure that only the necessary resources on the server are accessible to clients and only specific domains are allowed access to the server's resources.

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:

- Microsoft SharePoint Foundation 2010

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

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

<1> Section 2.1: The current implementation of Office Online returns the HTTP status code of 500 whenever an error is encountered while processing the client request.

8 Change Tracking

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

9 Index

N

O

P

Q

R

S

T

U

V

W