[MS-EMMWS]:
Microsoft Enterprise Managed Metadata Web Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

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

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
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	1.05	Major	Significantly changed the technical content.
12/17/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

Release: Sunday, December 19, 2010

1   Introduction

This document specifies the Enterprise Managed Metadata Web Service Protocol. This protocol enables a protocol client to interact with a managed metadata store. A typical scenario for using this protocol is tagging a document with managed metadata through a rich client application.

1.1   Glossary

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

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
language code identifier (LCID)

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

claim-based authentication mode
group
internal identifier
service application
SOAP (Simple Object Access Protocol)
SOAP action
SOAP body
SOAP fault
source term
term
term label
term set
term set group
term store
WSDL message
WSDL operation
XML namespace
XML namespace prefix
XML Schema

The following terms are specific to this document:

XML encode: A process that is used to convert XML content from one code to another to use that content without affecting the structure or validity of an XML document.

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

1.2   References

1.2.1   Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

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

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

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

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

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

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation 8 December 2009, http://www.w3.org/TR/REC-xml-names/

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

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

1.2.2   Informative References

[MS-EMMSTORE] Microsoft Corporation, "Enterprise Metadata Service Database Schema Specification", July 2009.

[MS-EMMWCF] Microsoft Corporation, "Metadata Shared Service WCF Service Protocol Specification", July 2009.

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

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

[MS-SPSTWS] Microsoft Corporation, "SharePoint Security Token Service Web Service Protocol Specification", July 2009.

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

[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   Protocol Overview (Synopsis)

Enterprises wanting to interact with digital assets and resources by way of metadata applied to the items often find that the inability to manage, share, update or restrict metadata values from one or many centralized services in the enterprise results in items that have inconsistent or unreliable metadata.

To achieve consistency in metadata use and behavior, metadata can be organized into **terms** and **term sets** which are managed centrally in **term stores**. Term stores may be located externally to client applications that apply managed metadata to their digital assets and resources. This protocol

specifies a web service that allows client applications to retrieve and add terms from remote term stores, thus bringing rich interaction with managed metadata to clients that are external to the term store servers.

For each term store maintained by the protocol server, the protocol server stores the **LCIDs** of the languages supported by the term store and the LCID of the default language for the term store.

Each term store can store one or more term sets. Each term set contains a hierarchy of terms such that each term set contains 0 or more root terms (terms without parents) and each term in the term set can have 1 or more child terms.

Each term in a term set can have a **term label** for each language supported by its term store.

1.4 Relationship to Other Protocols

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

This protocol is dependent on the [MS-EMMSTORE] protocol which defines the operations for creating and managing a term store. Whereas the [MS-EMMWCF] protocol provides access to term stores from the local server application, this protocol is specifically designed to provide external client applications with access to term stores.

The following diagram shows the underlying messaging and transport stack used by the protocol:
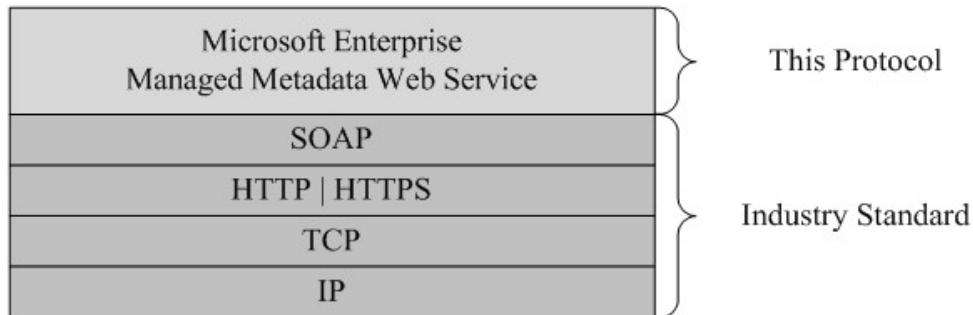
Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a site that is identified by a URL that is known by the protocol client. The protocol server endpoint is formed by appending "/_vti_bin/taxonomyclientservice.asmx" to the URL of the site. For example, if the URL of the site were http://www.contoso.com, the protocol server endpoint would be http://www.contoso.com/_vti_bin/taxonomyclientservice.asmx.

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

1.6 Applicability Statement

This protocol is applicable when the protocol client and protocol server are connected via a high-bandwidth, low-latency connection. It is not applicable to low-bandwidth or high-latency connections.

1.7 Versioning and Capability Negotiation

Supported transports: This protocol uses multiple transports with SOAP as specified in section 2.1.

Protocol versions: This document specifies the only version of this protocol.

Security and authentication methods: This protocol uses **claim-based authentication mode** to verify read and write access to the connected term store. Claim-based authentication is described in [MS-SPSTWS].

Capability negotiation: None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

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

2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses the XML Schema as defined in [XMLSCHEMA1] and [XMLSCHEMA2], and Web Services Description Language as defined in [WSDL].

2.2.1 Namespaces

This specification defines and references **XML namespaces** using the mechanisms specified in [XMLNS]. Although this specification 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.

Prefix	Namespace URI	Reference
tm	http://microsoft.com/wsdl/mime/textMatching/	
s1	http://microsoft.com/wsdl/types/	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
tns	http://schemas.microsoft.com/sharepoint/taxonomy/soap/	
wsdl	http://schemas.xmlsoap.org/wsdl/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	
soapenc	http://schemas.xmlsoap.org/soap/encoding/	
http	http://schemas.xmlsoap.org/wsdl/http/	
mime	http://schemas.xmlsoap.org/wsdl/mime/	

2.2.2 Messages

None.

2.2.3 Elements

This specification does not define any common XML Schema element definitions.

2.2.4   Complex Types

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

Complex type	Description
AddTermsTermStoreResponseObject	Specifies the collection of terms that were added to the term store via the **AddTerms** operation.
ClientTimeStampsForGetTermSetsClientService	Specifies the time stamp of the last time term set data was received from each requested term set.
ClientVersionsForGetTermSetsClientService	Specifies a collection of versions used to determine what server version the term set data on the client was retrieved from.
GetTermSetsResponseObject	Specifies the data for a collection of term sets.
NewTermCollectionForAddTermsClientService	Specifies a collection of terms to be added to a term store.
NewTermForAddTermsClientServiceChild	Specifies a sub-tree of terms to be added to a term set.
NewTermForAddTermsClientServiceParent	Specifies a sub-tree of terms to be added to a term set and the location within the term set at which the sub-tree should be rooted.
SerializedSharedItemForGetTermSets	Specifies a term set and all of its terms.
SerializedSharedTerm	Specifies a term.
SerializedSharedTermMembership	Describes a term set to which a term belongs.
SerializedSharedTermSet	Describes a term set.
ServerTermSetTimeStampsForGetTermSetsClientService	Specifies the time stamps at which a collection of term sets have most recently been updated.
SspIdsForGetTermSetsClientService	Specifies which term store should be used when retrieving term set data with **GetTermSets**.
TermIdsForGetKeywordsClientService	Specifies a collection of term identifiers.
TermIdsForGetTermsByLabelClientService	Specifies a collection of term identifiers to be validated for existence in the term store.
TermSetIdsForGetTermSetsClientService	Specifies the term sets that should be retrieved.

Release: Sunday, December 19, 2010

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

Complex Type	Description
AddTermsTermStoreResponseObject	**AddTermsTermStoreResponseObject** is used to return data that describes a collection of terms and is defined as follows.
ClientTimeStampsForGetTermSetsClientService	**ClientTimeStampsForGetTermSetsClientService** specifies a collection of time stamps used to determine whether or not to send term set data to the client. The time stamps in this collection MUST be in one-to-one correspondence with a collection of term set identifiers specified in the same client request. It is defined as follows.
ClientVersionsForGetTermSetsClientService	**ClientVersionsForGetTermSetsClientService** specifies a collection of versions used to determine what server version the term set data on the client was retrieved from. The versions in this collection MUST be in one-to-one correspondence with a collection of term set identifiers specified in the same client request. It is defined as follows.
GetTermSetsResponseObject	**GetTermSetsResponseObject** is used to return data for each of the term sets that was requested. For each term set requested an empty node will be returned which means it is up to date, or the entire term set will be returned if it has been updated in any way since the last time it was retrieved. It is defined as follows.
NewTermCollectionForAddTermsClientService	**NewTermCollectionForAddTermsClientService** is used to request that new terms be added to the term store and is defined as follows.
NewTermForAddTermsClientServiceChild	**NewTermForAddTermsClientServiceChild** specifies a sub-tree of new terms to be added to a term set and is defined as follows.
NewTermForAddTermsClientServiceParent	**NewTermForAddTermsClientServiceParent** specifies a sub-tree of new terms to be added to a term set and the location within the term set at which the sub-tree should be rooted. It is defined as follows.
SerializedSharedItemForGetTermSets	**SerializedSharedItemForGetTermSets** is used to fully specify a single term set with all of its terms and is defined as follows.
SerializedSharedTerm	**SerializedSharedTerm** is used to fully specify a single term and is defined as follows.
SerializedSharedTermMembership	**SerializedSharedTermMembership** is used to fully describe a term set to which a term belongs and is defined as follows.
SerializedSharedTermSet	**SerializedSharedTermSet** is used to fully describe a term set and is defined as follows.
ServerTermSetTimeStampsForGetTermSetsClientS	**ServerTermSetTimeStampsForGetTermSetsClientS**

Complex Type	Description
ervice	**ervice** is used to return the time stamps that a term set has most recently been updated and is defined as follows.
SspIdsForGetTermSetsClientService	**SspIdsForGetTermSetsClientService** is used to specify which term store should be used when getting term set data in the **GetTermSets** web service and is defined as follows.
TermIdsForGetKeywordsClientService	Specifies a set of term identifiers and is defined as follows.
TermIdsForGetTermsByLabelClientService	**TermIdsForGetTermsByLabelClientService** specifies a collection of term identifiers to be validated for existence in the term store and is defined as follows.
TermSetIdsForGetTermSetsClientService	**TermSetIdsForGetTermSetsClientService** specifies the term sets that should be retrieved and is defined as follows.

2.2.4.1 AddTermsTermStoreResponseObject

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

AddTermsTermStoreResponseObject is used to return data that describes a collection of terms and is defined as follows.

```
<s:complexType name="AddTermsTermStoreResponseObject">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="TermStore">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="unbounded" name="T"
type="SerializedSharedTerm"/>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

TermStore: This is the collection of terms that have been returned from the request.

TermStore.T: Each term requested in this request is returned as a **SerializedSharedTerm**.

2.2.4.2 ClientTimeStampsForGetTermSetsClientService

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

ClientTimeStampsForGetTermSetsClientService specifies a collection of time stamps used to determine whether or not to send term set data to the client. The time stamps in this collection MUST be in one-to-one correspondence with a collection of term set identifiers specified in the same client request. It is defined as follows.

```
<s:complexType name="ClientTimeStampsForGetTermSetsClientService">
  <s:sequence>
```

```
        <s:element minOccurs="1" maxOccurs="1" name="dateTimes">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="0" maxOccurs="unbounded" name="dateTime" type="s:long"/>
            </s:sequence>
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
```

dateTimes: The collection of time stamps.

dateTimes.dateTime: A number representing a count of ticks (100-nanosecond units) since the last time the system hosting the term store was started. Any non-null value of this element that cannot be parsed as a count of ticks MUST be treated as 0.

2.2.4.3   ClientVersionsForGetTermSetsClientService

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

ClientVersionsForGetTermSetsClientService specifies a collection of versions used to determine what server version the term set data on the client was retrieved from. The versions in this collection MUST be in one-to-one correspondence with a collection of term set identifiers specified in the same client request. It is defined as follows.

```
  <s:complexType name="ClientVersionsForGetTermSetsClientService">
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versions">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="unbounded" name="version" type="s:long"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
```

versions: The collection of versions.

versions.version: A number representing the protocol version of the server that the stored term set was retrieved from. Currently there is only one protocol version which is version 1. If another version of the protocol ships it will increment the version number to 2. If no term set is stored on the client this MUST be 0. Otherwise, it MUST be 1. If this element is NULL or not specified, an empty data node MUST be returned to the client.

2.2.4.4   GetTermSetsResponseObject

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

GetTermSetsResponseObject is used to return data for each of the term sets that was requested. For each term set requested an empty node will be returned which means it is up to date, or the entire term set will be returned if it has been updated in any way since the last time it was retrieved. It is defined as follows.

```
  <s:complexType name="GetTermSetsResponseObject">
```

```
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="Container">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="unbounded" name="TermStore"
  type="SerializedSharedItemForGetTermSets"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
```

Container: This is the collection of term sets which were requested. If the client request contained a mismatch in the number of client term set time stamps, versions, and identifiers, then this element MUST NOT be present.

Container.TermStore: This either specifies that a term set is up to date or fully describes an entire term set and all its terms if it has been updated.

2.2.4.5 NewTermCollectionForAddTermsClientService

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

NewTermCollectionForAddTermsClientService is used to request that new terms be added to the term store and is defined as follows.

```
  <s:complexType name="NewTermCollectionForAddTermsClientService">
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="newTerms">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="unbounded" name="newTerm"
  type="NewTermForAddTermsClientServiceParent"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
```

newTerms: This is the collection of new terms which will be added to the term store.

newTerms.newTerm: Each new term that will be added to the term store.

2.2.4.6 NewTermForAddTermsClientServiceChild

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

NewTermForAddTermsClientServiceChild specifies a sub-tree of new terms to be added to a term set and is defined as follows.

```
  <s:complexType name="NewTermForAddTermsClientServiceChild">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded" name="newTerm"
  type="NewTermForAddTermsClientServiceChild"/>
    </s:sequence>
```

```
    <s:attribute name="label" type="s:string"/>
    <s:attribute name="clientId" type="s:int"/>
    <s:attribute name="parentTermId" type="s1:guid"/>
  </s:complexType>
```

newTerm: A branch of the sub-tree. The term described by the **label** attribute of the
NewTermForAddTermsClientServiceChild type is the parent term of this branch. If this node is
absent, no new terms will be added to this branch of the term set.

label: Default label of the new term at the root of this sub-tree. The following characters MUST NOT
be used in a term label:

- Opening square bracket ([)

- Semicolon (;)

- Quotation mark (")

- Angle brackets (<>)

- Vertical bar (|)

- Ampersand (&)

The label MUST be a string with minimum length 1 character and maximum length 255 characters
after spaces are trimmed.

clientId: An identifier for the new term. There MUST be a unique **clientId** value for each new term
specified within a single enclosing object of type **NewTermCollectionForAddTermsClientService**.
These **clientId** values MUST lie in the range 1 to n where n is the number of new terms being
specified by the **NewTermCollectionForAddTermsClientService** object.

parentTermId: This value MUST be the empty guid.

2.2.4.7   NewTermForAddTermsClientServiceParent

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

NewTermForAddTermsClientServiceParent specifies a sub-tree of new terms to be added to a
term set and the location within the term set at which the sub-tree should be rooted. It is defined as
follows.

```
  <s:complexType name="NewTermForAddTermsClientServiceParent">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded" name="newTerm"
  type="NewTermForAddTermsClientServiceChild"/>
    </s:sequence>
    <s:attribute name="label" type="s:string"/>
    <s:attribute name="clientId" type="s:int"/>
    <s:attribute name="parentTermId" type="s1:guid"/>
  </s:complexType>
```

newTerm: A branch of the sub-tree. The term described by the **label** attribute of the
NewTermForAddTermsClientServiceParent type is the parent term of this branch. If this node is
absent, no new terms will be added to this branch of the term set. See
NewTermForAddTermsClientServiceChild (section 2.2.4.6) for a description of that type.

label: Default label of the new term at the root this sub-tree. The label MUST be a string with minimum length 1 character and maximum length 255 characters after spaces are trimmed, and MUST NOT contain the invalid characters specified in the previous section.

clientId: An identifier for the new term. There MUST be a unique **clientId** value for each new term specified within a single enclosing object of type **NewTermCollectionForAddTermsClientService**. These **clientId** values MUST lie in the range 1 to *n* where *n* is the number of new terms being specified by the **NewTermCollectionForAddTermsClientService** object.

parentTermId: The identifier of the term to which this sub-tree will be attached. MUST be either the GUID of an existing term in the **term set** or the empty GUID, if the sub-tree is to be added at the root of the **term set**.

2.2.4.8 SerializedSharedItemForGetTermSets

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

SerializedSharedItemForGetTermSets is used to fully specify a single term set with all of its terms and is defined as follows.

```
<s:complexType name="SerializedSharedItemForGetTermSets">
  <s:sequence>
    <s:choice minOccurs="0" maxOccurs="unbounded">
      <s:element minOccurs="1" maxOccurs="1" name="TS" type="SerializedSharedTermSet"/>
      <s:element minOccurs="1" maxOccurs="unbounded" name="T" type="SerializedSharedTerm"/>
    </s:choice>
    <s:element minOccurs="0" maxOccurs="unbounded" name="VI">
      <s:complexType>
        <s:attribute name="a9" type="s1:guid" use="required"/>
      </s:complexType>
    </s:element>
    <s:element minOccurs="0" maxOccurs="unbounded" name="II">
      <s:complexType>
        <s:attribute name="a9" type="s1:guid" use="required"/>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

TS: Container node for term set data.

T: Collection of terms in term set.

VI: Each term that was found in the term store and was specifically requested. If no terms were requested or if none of the requested terms were found, this node MUST be absent.

VI.a9: Identifier of the valid term.

II: Each term that was not found in the term store but was requested. If no terms were requested, or if all of the requested terms were found, this node MUST be absent.

II.a9: Identifier of the invalid term.

2.2.4.9 SerializedSharedTerm

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

SerializedSharedTerm is used to fully specify a single term and is defined as follows.

```xml
<s:complexType name="SerializedSharedTerm">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="LS">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="unbounded" name="TL">
            <s:complexType>
              <s:attribute name="a32" type="s:string" use="optional"/>
              <s:attribute name="a31" type="s:boolean" use="required"/>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element minOccurs="1" maxOccurs="1" name="DS">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="TD">
            <s:complexType>
              <s:attribute name="a11" type="s:string" use="required"/>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element minOccurs="1" maxOccurs="1" name="TMS" type="SerializedSharedTermMembership"/>
    <s:element minOccurs="0" maxOccurs="1" name="MIS">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="unbounded" name="MI">
            <s:complexType>
              <s:attribute name="a9" type="s1:guid" use="required"/>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
  <s:attribute name="a9" type="s1:guid" use="required"/>
  <s:attribute name="a21" type="s:boolean" use="required"/>
  <s:attribute name="a61" type="s:int" use="required"/>
  <s:attribute name="a72" type="s:int" use="optional"/>
  <s:attribute name="a1000" type="s:string" use="optional"/>
</s:complexType>
```

LS: This is the collection of term labels that this term contains.

LS.TL: This fully describes a single term label.

LS.TL.a32: Term label value. The label MUST be a string with minimum length 1 character and maximum length 255 characters after spaces are trimmed.

LS.TL.a31: MUST be true if the term label is the default term label, otherwise false. Only one term label MUST be the default term label for each language.

DS: Container node for description.

DS.TD: term description node. If the term does not contain a description in the requested language or the term store default language, this node MUST NOT be present.

DS.TD.a11: Description value. The description MUST be a string with minimum length 0 characters and maximum length 1000 characters.

TMS: term set node for the term set that the term belongs to.

MIS: This is the collection of terms that have been merged with this term. If no terms have been merged with this term, this node MUST NOT be present.

MIS.MI: This node fully describes a merged term.

MIS.MI.a9: Identifier for merged term.

a9: Identifier of term.

a21: MUST be **false** if the term is not deprecated.

a61: internal identifier for term.

a72: Absolute sort order of the term in regards to all other terms in term set.

a1000: Comma-separated list of **WssId**s that MAY be returned by the **GetChildTermsInTerm** and **GetChildTermsInTermSet**. It MUST NOT be present on the response for the **AddTerms** operation.

2.2.4.10   SerializedSharedTermMembership

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

SerializedSharedTermMembership is used to fully describe a term set to which a term belongs and is defined as follows.

```
<s:complexType name="SerializedSharedTermMembership">
  <s:sequence>
    <s:element name="TM">
      <s:complexType>
        <s:attribute name="a24" type="s1:guid" use="required"/>
        <s:attribute name="a12" type="s:string" use="required"/>
        <s:attribute name="a25" type="s1:guid" use="optional"/>
        <s:attribute name="a40" type="s:string" use="required"/>
        <s:attribute name="a17" type="s:boolean" use="required"/>
        <s:attribute name="a67" type="s:string" use="required"/>
        <s:attribute name="a45" type="s:string" use="required"/>
        <s:attribute fixed="true" name="a69" type="s:boolean" use="optional"/>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

TM: This node fully describes a term set to which a term belongs.

TM.a24: Identifier of term set.

TM.a12: Term set name in the language requested by the client. If the term set does not have a name in the client's language, the name in the term store default language is returned. The name

MUST be a string with minimum length 1 character and maximum length 255 characters after spaces are trimmed.

TM.a25: Identifier of term that this term set information is pertaining to.

TM.a40: term path of term with term labels. This path starts from the root term and goes until the parent of the term. Consequently this attribute is an empty string for root terms. The maximum length of each term label is 255 characters.

TM.a17: MUST be true if term set is available for tagging or false otherwise. A term set is available for tagging if the terms it contains may be used to tag list items or documents.

TM.a67: A string value that indicates a custom sort order for the child terms of the term identified by **TM.a25**. If the term has no child terms, or if the default sort order is to be applied, the value of this attribute MUST be an empty string. In that case when the child terms are requested, they MUST be sorted alphabetically by their default label. If a custom sort order has been specified for the child terms, the value of this node MUST be a string of the form "$identifier_1:identifier_2:…$" where $identifier_n$ is a valid child term identifier. The custom sort order then corresponds to the order of these identifiers.

TM.a45: term path of term with identifiers. This path starts from the root term and goes until the term itself. Consequently this property MUST NOT be an empty string.

TM.a69: MUST be **true** if term identified by **a25** has child terms.

2.2.4.11   SerializedSharedTermSet

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

SerializedSharedTermSet is used to fully describe a term set and is defined as follows.

```
<s:complexType name="SerializedSharedTermSet">
  <s:attribute name="a9" type="s1:guid" use="required"/>
  <s:attribute name="a17" type="s:boolean" use="required"/>
  <s:attribute name="a11" type="s:string" use="required"/>
  <s:attribute name="a16" type="s:boolean" use="required"/>
  <s:attribute name="a12" type="s:string" use="required"/>
  <s:attribute name="a68" type="s:string" use="required"/>
</s:complexType>
```

a9: Identifier of term set.

a17: MUST be **true** if term set is available for tagging, or **false** otherwise.

a11: Term set description. The maximum length of this string is 1000 characters; it can be an empty string.

a16: MUST be **true** if the term set is open, or **false** otherwise. A term set is open if users without explicit rights to the **group** are allowed to add terms to the term set.

a12: Term set name in the language requested by client. If the term set does not have a name in the client's language, the name in the term store default language is returned. The name MUST be a string with minimum length 1 character and maximum length 255 characters after spaces are trimmed.

a68: Term set contact. The contact is the user or group that should be contacted if a user wants to request the addition of a term to the term set, and the term set is not open. The contact MUST be a string with a maximum length of 320 characters; it can be an empty string.

2.2.4.12   ServerTermSetTimeStampsForGetTermSetsClientService

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

ServerTermSetTimeStampsForGetTermSetsClientService is used to return the time stamps that a term set has most recently been updated and is defined as follows.

```
<s:complexType name="ServerTermSetTimeStampsForGetTermSetsClientService">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Container">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="unbounded" name="Node">
            <s:complexType>
              <s:attribute name="Time" type="s:long" use="required"/>
              <s:attribute name="TermId" type="s1:guid" use="required"/>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

Container: This is the collection of time stamps which represent the time that the requested **term set**s were updated at.

Container.Node: Time stamp for a term set.

Container.Node.Time: MUST be time of last update in ticks, or empty if passed in time was current.

Container.Node.TermId: Identifier of the term set.

2.2.4.13   SspIdsForGetTermSetsClientService

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

SspIdsForGetTermSetsClientService is used to specify which term store should be used when getting term set data in the **GetTermSets** web service and is defined as follows.

```
<s:complexType name="SspIdsForGetTermSetsClientService">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="sspIds">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="sspId" type="s1:guid"/>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

sspIds: This is the collection of term store identifiers that specify which term store each term set should be retrieved from.

sspIds.sspId: Identifier of the term store. If this node is absent, no data will be retrieved for this web service request.

2.2.4.14 TermIdsForGetKeywordsClientService

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

Specifies a set of term identifiers and is defined as follows.

```
<s:complexType name="TermIdsForGetKeywordsClientService">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="unbounded" name="termId" type="s1:guid"/>
  </s:sequence>
</s:complexType>
```

termId: The identifier of a **term**.

2.2.4.15 TermIdsForGetTermsByLabelClientService

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

TermIdsForGetTermsByLabelClientService specifies a collection of term identifiers to be validated for existence in the term store and is defined as follows.

```
<s:complexType name="TermIdsForGetTermsByLabelClientService">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="termIds">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="termId" type="s1:guid"/>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

termIds: The collection of terms to be validated.

termIds.termId: The identifier of a single term to be validated. If this node is absent, no term identifiers will be validated for existence in the term store.

2.2.4.16 TermSetIdsForGetTermSetsClientService

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

TermSetIdsForGetTermSetsClientService specifies the term sets that should be retrieved and is defined as follows.

```
<s:complexType name="TermSetIdsForGetTermSetsClientService">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="termSetIds">
      <s:complexType>
```

```
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="termSetId" type="s1:guid"/>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

termSetIds: The collection of term set identifiers which will be used to retrieve term set data.

termSetIds.termSetId: The identifier of a term set. If this node is absent, no term sets will be retrieved.

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
guid	A globally unique identifier that is defined as follows.
MatchOptions	Matching options for term search defined as follows. Matches MUST be case-insensitive.

2.2.5.1 guid

Namespace: http://microsoft.com/wsdl/types/

A globally unique identifier that is defined as follows.

```
<xs:simpleType name="guid">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
  </xs:restriction>
</xs:simpleType>
```

2.2.5.2 MatchOptions

Namespace: http://schemas.microsoft.com/sharepoint/soap/taxonomy/

Matching options for term search defined as follows. Matches MUST be case-insensitive.

```
<s:simpleType name="MatchOptions">
  <s:restriction base="s:string">
    <s:enumeration value="StartsWith"/>
    <s:enumeration value="ExactMatch"/>
  </s:restriction>
</s:simpleType>
```

The following table specifies the allowable values for MatchOptions:

Value	Meaning
StartsWith	Term label begins with a given string.
ExactMatch	Term label exactly matches a given string.

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3   Protocol Details

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   Server Details

3.1.1   Abstract Data Model

The protocol server is used to access term stores in multiple **service applications**.

Each term store maintains term sets and terms. Term sets contain a hierarchy of terms (potentially re-using individual terms in multiple term sets, for example, "JavaScript" being reused as a programming language name and as a product name); that is, the term set has root terms (terms without parents), and each term has 0 or more child terms.

Each term set and each term has values for one or more of the languages. For example, a term has a term label for each language.

For terms that appear in multiple term sets, only one instance will be the **source term**, that is, the term that is used for permission checking when editing.

Each term store maintains groups of related term sets, such as "Cities in Washington", "Cities in Arizona", and so on. Each **term set group** has permission settings controlling access.

The following diagram shows these relationships graphically.
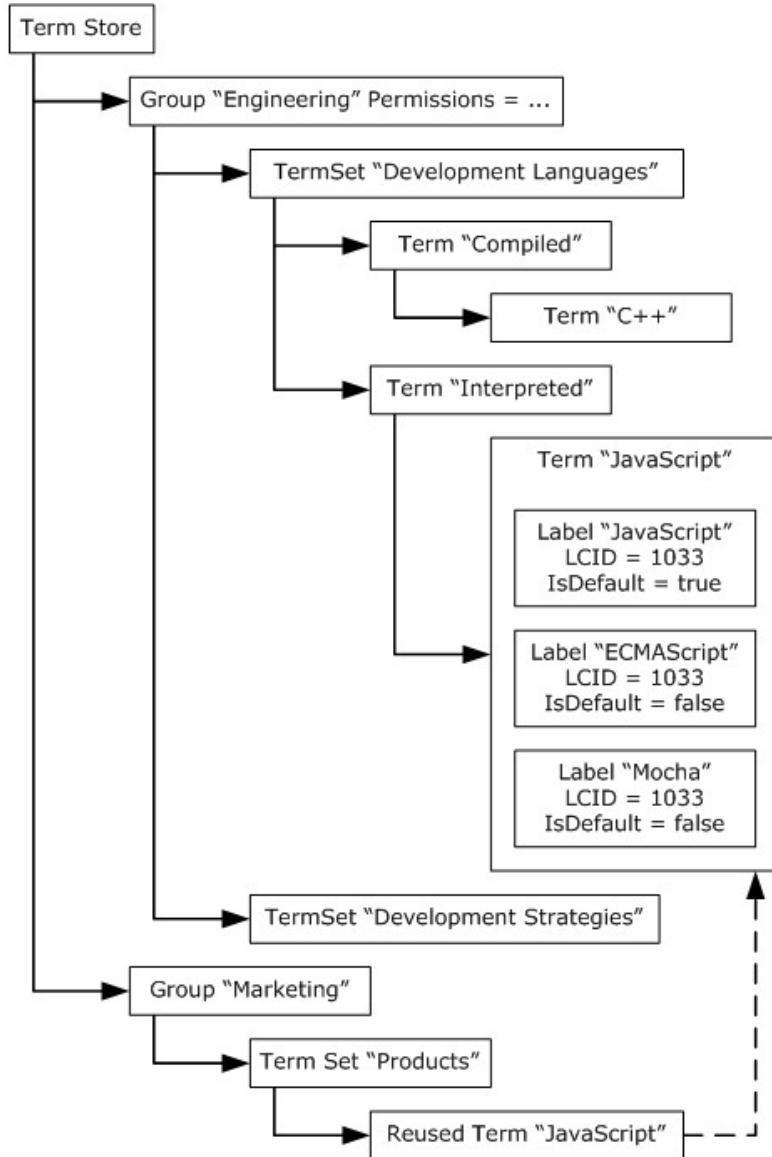
Figure 2: Abstract data model

3.1.2   Timers

None.

3.1.3   Initialization

A connection to the term store through the protocols specified in section 1.4 MUST be established before using this protocol.

3.1.4   Message Processing Events and Sequencing Rules

This specification includes the following **WSDL operations**:

WSDL Operation	Description
AddTerms	This operation adds new terms to a term set and specifies the hierarchical relations of the new terms to each other and to existing terms. This operation is defined as follows.
GetChildTermsInTerm	This operation retrieves the immediate child terms for a specific term and is defined as follows.
GetChildTermsInTermSet	This operation retrieves the immediate child terms for a specific term set and is defined as follows.
GetKeywordTermsByGuids	This operation allows the protocol client to retrieve full data on a specified set of terms from the protocol server. The operation is defined as follows.
GetTermsByLabel	This operation searches for terms in the term store by their term label. It will add unfound terms to the default keywords term store if the protocol client requests this behavior. This operation is defined as follows.
GetTermSets	This operation retrieves term sets from the term store and is defined as follows.

3.1.4.1 AddTerms

This operation adds new terms to a term set and specifies the hierarchical relations of the new terms to each other and to existing terms. This operation is defined as follows.

```
<wsdl:operation name="AddTerms">
  <wsdl:input message="tns:AddTermsSoapIn"/>
  <wsdl:output message="tns:AddTermsSoapOut"/>
</wsdl:operation>
```

3.1.4.1.1 Messages

The following **WSDL message** definitions are specific to this operation.

3.1.4.1.1.1 AddTermsSoapIn

The request WSDL message for the **AddTerms** WSDL operation.

The **SOAP action** value is:

```
http://schemas.microsoft.com/sharepoint/taxonomy/soap/AddTerms
```

The **SOAP body** contains the **AddTerms** element.

3.1.4.1.1.2 AddTermsSoapOut

The response WSDL message for the **AddTerms** WSDL operation.

The SOAP body contains the **AddTermsResponse** element.

3.1.4.1.2 Elements

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

3.1.4.1.2.1 AddTerms

The input data for the **AddTerms** WSDL operation.

```
<xs:element name="AddTerms">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="sharedServiceId"
xmlns:q1="http://microsoft.com/wsdl/types/" type="q1:guid"/>
      <xs:element minOccurs="1" maxOccurs="1" name="termSetId"
xmlns:q2="http://microsoft.com/wsdl/types/" type="q2:guid"/>
      <xs:element minOccurs="1" maxOccurs="1" name="lcid" type="xs:int"/>
      <xs:element minOccurs="0" maxOccurs="1" name="newTerms" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sharedServiceId: The identifier of the term store to which the new terms will be added. If the empty GUID is specified, the terms MUST be added to the default keywords term store.<1>

termSetId: The identifier of the term set to which the new terms will be added. If the empty GUID is specified, the terms MUST be added to the keywords term set.<2>

lcid: The language in which the term labels will be saved, if the language exists in the term store. If it does not exist the term store default language MUST be used.

newTerms: Specifies the new terms to be added, which is an **XML encoded** string that MUST conform to the schema of the **NewTermCollectionForAddTermsClientService** complex type (section 2.2.4.5). No new terms will be added if this node is absent. The protocol client MUST assign a unique **clientId** value to each new term specified by the **NewTermForAddTermsClientServiceParent** (section 2.2.4.7) and **NewTermForAddTermsClientServiceChild** elements (section 2.2.4.6) within a single enclosing object of type **NewTermCollectionForAddTermsClientService**. These **clientId** values MUST lie in the range 1 to n where n is the number of new terms being specified by the **NewTermCollectionForAddTermsClientService** object.

3.1.4.1.2.2 AddTermsResponse

The result data for the **AddTerms** WSDL operation.

```
<xs:element name="AddTermsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="AddTermsResult" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

AddTermsResult: The specification of the terms that were added based on the request, which is an XML encoded string that MUST conform to the schema of the **AddTermsTermStoreResponseObject** complex type (section 2.2.4.1). Identifiers are assigned to the terms when they are created in the term store. The specification of the terms is retrieved after the terms are added to the term store.

3.1.4.1.3 Complex Types

None.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 GetChildTermsInTerm

This operation retrieves the immediate child terms for a specific term and is defined as follows.

```
<wsdl:operation name="GetChildTermsInTerm">
  <wsdl:input message="tns:GetChildTermsInTermSoapIn"/>
  <wsdl:output message="tns:GetChildTermsInTermSoapOut"/>
</wsdl:operation>
```

3.1.4.2.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.2.1.1 GetChildTermsInTermSoapIn

The request WSDL message for the **GetChildTermsInTerm** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetChildTermsInTerm
```

The SOAP body contains the **GetChildTermsInTerm** element.

3.1.4.2.1.2 GetChildTermsInTermSoapOut

The response WSDL message for the **GetChildTermsInTerm** WSDL operation.

The SOAP body contains the **GetChildTermsInTermResponse** element.

3.1.4.2.2 Elements

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

3.1.4.2.2.1 GetChildTermsInTerm

The input data for the **GetChildTermsInTerm** WSDL operation.

```
<xs:element name="GetChildTermsInTerm">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="sspId"
xmlns:q5="http://microsoft.com/wsdl/types/" type="q5:guid"/>
      <xs:element minOccurs="1" maxOccurs="1" name="lcid" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="termId"
xmlns:q6="http://microsoft.com/wsdl/types/" type="q6:guid"/>
      <xs:element minOccurs="1" maxOccurs="1" name="termSetId"
xmlns:q7="http://microsoft.com/wsdl/types/" type="q7:guid"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sspId: Identifier for the term store.<3>

lcid: The **LCID** of the requested language. If the requested language is not available in the term store, the default term store language is used.

termId: The term identifier of the term whose child terms are to be retrieved.<4>

termSetId: The term set identifier of the term set from which to retrieve the terms.<5>

3.1.4.2.2.2 GetChildTermsInTermResponse

The result data for the **GetChildTermsInTerm** WSDL operation.

```
<xs:element name="GetChildTermsInTermResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetChildTermsInTermResult"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetChildTermsInTermResult: The terms that are child terms of the requested term, which is an XML encoded string that MUST conform to the schema of the **AddTermsTermStoreResponseObject** complex type (section 2.2.4.1).

3.1.4.2.3 Complex Types

None.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 GetChildTermsInTermSet

This operation retrieves the immediate child terms for a specific term set and is defined as follows.

```
<wsdl:operation name="GetChildTermsInTermSet">
  <wsdl:input message="tns:GetChildTermsInTermSetSoapIn"/>
  <wsdl:output message="tns:GetChildTermsInTermSetSoapOut"/>
</wsdl:operation>
```

3.1.4.3.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.3.1.1 GetChildTermsInTermSetSoapIn

The request WSDL message for the **GetChildTermsInTermSet** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetChildTermsInTermSet
```

The SOAP body contains the **GetChildTermsInTermSet** element.

3.1.4.3.1.2 GetChildTermsInTermSetSoapOut

The response WSDL message for the **GetChildTermsInTermSet** WSDL operation.

The SOAP body contains the **GetChildTermsInTermSetResponse** element.

3.1.4.3.2 Elements

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

3.1.4.3.2.1 GetChildTermsInTermSet

The input data for the **GetChildTermsInTermSet** WSDL operation.

```
<xs:element name="GetChildTermsInTermSet">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="sspId"
xmlns:q3="http://microsoft.com/wsdl/types/" type="q3:guid"/>
      <xs:element minOccurs="1" maxOccurs="1" name="lcid" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="termSetId"
xmlns:q4="http://microsoft.com/wsdl/types/" type="q4:guid"/>
    </xs:sequence>
```

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

sspId: Identifier for the term store.<6>

lcid: The **LCID** of the requested language. If the requested language is not available in the term store, the default term store language is used.

termSetId: The term set identifier of the term set from which to retrieve the terms.<7>

3.1.4.3.2.2   GetChildTermsInTermSetResponse

The result data for the **GetChildTermsInTermSet** WSDL operation.

```
<xs:element name="GetChildTermsInTermSetResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetChildTermsInTermSetResult"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetChildTermsInTermSetResult: The terms that are child terms of the requested term set, which is an XML encoded string that MUST conform to the schema of the **AddTermsTermStoreResponseObject** complex type (section 2.2.4.1).

3.1.4.3.3   Complex Types

None.

3.1.4.3.4   Simple Types

None.

3.1.4.3.5   Attributes

None.

3.1.4.3.6   Groups

None.

3.1.4.3.7   Attribute Groups

None.

3.1.4.4   GetKeywordTermsByGuids

This operation allows the protocol client to retrieve full data on a specified set of terms from the protocol server. The operation is defined as follows.

```
<wsdl:operation name="GetKeywordTermsByGuids">
```

```
  <wsdl:input message="tns:GetKeywordTermsByGuidsSoapIn"/>
  <wsdl:output message="tns:GetKeywordTermsByGuidsSoapOut"/>
</wsdl:operation>
```

This operation accepts a set of term identifiers and returns the term data for each requested term that exists in the term store and that is available for tagging and is not deprecated.

3.1.4.4.1   Messages

The following WSDL message definitions are specific to this operation.

3.1.4.4.1.1   GetKeywordTermsByGuidsSoapIn

The request WSDL message for the **GetKeywordTermsByGuids** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetKeywordTermsByGuids
```

The SOAP body contains the **GetKeywordTermsByGuids** element.

3.1.4.4.1.2   GetKeywordTermsByGuidsSoapOut

The response WSDL message for the **GetKeywordTermsByGuids** WSDL operation.

The SOAP body contains the **GetKeywordTermsByGuidsResponse** element.

3.1.4.4.2   Elements

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

3.1.4.4.2.1   GetKeywordTermsByGuids

The input data for the **GetKeywordTermsByGuids** WSDL operation.

```
<xs:element name="GetKeywordTermsByGuids">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="termIds" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="lcid" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

termIds: Specifies the identifiers of the requested terms. This field MUST be formatted as an XML encoded string that conforms to the **TermIdsForGetKeywordsClientService** complex type (section 2.2.4.14) schema. The terms are retrieved from all the term stores accessible to the protocol server. If all the term stores are online, any requested terms that cannot be found will not be included in the response. If none of the terms can be found, the response will contain no identifiers. However if any of the term stores are offline, and any of the requested terms cannot be found in the remaining term stores, the protocol server MUST return a SOAP fault message stating that a term store operation exception occurred.

lcid: The requested term label language. If this language does not exist in the term store, the protocol server MUST return term labels using the default term store language.

3.1.4.4.2.2   GetKeywordTermsByGuidsResponse

The result data for the **GetKeywordTermsByGuids** WSDL operation.

```
<xs:element name="GetKeywordTermsByGuidsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetKeywordTermsByGuidsResult"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetKeywordTermsByGuidsResult: This is the serialized term data of the terms which were requested, which is an XML encoded string that MUST conform to the schema of the **AddTermsTermStoreResponseObject** complex type (section 2.2.4.1).

3.1.4.4.3   Complex Types

None.

3.1.4.4.4   Simple Types

None.

3.1.4.4.5   Attributes

None.

3.1.4.4.6   Groups

None.

3.1.4.4.7   Attribute Groups

None.

3.1.4.5   GetTermsByLabel

This operation searches for terms in the term store by their term label. It will add unfound terms to the default keywords term store if the protocol client requests this behavior. This operation is defined as follows.

```
<wsdl:operation name="GetTermsByLabel">
  <wsdl:input message="tns:GetTermsByLabelSoapIn"/>
  <wsdl:output message="tns:GetTermsByLabelSoapOut"/>
</wsdl:operation>
```

3.1.4.5.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.5.1.1 GetTermsByLabelSoapIn

The request WSDL message for the **GetTermsByLabel** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetTermsByLabel
```

The SOAP body contains the **GetTermsByLabel** element.

3.1.4.5.1.2 GetTermsByLabelSoapOut

The response WSDL message for the **GetTermsByLabel** WSDL operation.

The SOAP body contains the **GetTermsByLabelResponse** element.

3.1.4.5.2 Elements

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

3.1.4.5.2.1 GetTermsByLabel

The input data for the **GetTermsByLabel** WSDL operation.

```
<xs:element name="GetTermsByLabel">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="label" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="lcid" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="matchOption" type="MatchOptions"/>
      <xs:element minOccurs="1" maxOccurs="1" name="resultCollectionSize" type="xs:int"/>
      <xs:element minOccurs="0" maxOccurs="1" name="termIds" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="addIfNotFound" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

label: The set of term labels to be searched for. Each individual label within this string MUST be separated from the preceding label (if any) by a semicolon (;). The following characters MUST NOT be used in a term label:

- Opening square bracket ([)

- Semicolon (;)

- Quotation mark (")

- Angle brackets (<>)

- Vertical line (|)

- Ampersand (&)

The label MUST be a string with minimum length 1 character and maximum length 255 characters after spaces are trimmed.

lcid: The LCID of the term label language to which the search MUST be restricted. If the requested language is not available in the term store, the default term store language is used.

matchOption: The type of search to be performed.

resultCollectionSize: The number of terms to be returned.

termIds: Term identifiers to be validated for existence, which is an XML encoded string that MUST conform to the schema of the **TermIdsForGetTermsByLabelClientService** complex type (section 2.2.4.15). If this node is absent, no term identifiers will be validated for existence.

addIfNotFound: If **true**, the protocol server MUST add unfound terms to the default keywords term store and MUST place these terms in the keywords term set. In this situation, the **lcid** parameter specifies the language of the new terms.<8> If **addIfNotFound** is **false**, the protocol server MUST NOT add unfound terms to the default keywords term store.

3.1.4.5.2.2 GetTermsByLabelResponse

The result data for the **GetTermsByLabel** WSDL operation.

```
<xs:element name="GetTermsByLabelResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetTermsByLabelResult" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetTermsByLabelResult: Serialized data for each term found that matches request, which is an XML encoded string that MUST conform to the schema of the **SerializedSharedTerm** complex type (section 2.2.4.9).

3.1.4.5.3 Complex Types

None.

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.4.6 GetTermSets

This operation retrieves term sets from the term store and is defined as follows.

```
<wsdl:operation name="GetTermSets">
  <wsdl:input message="tns:GetTermSetsSoapIn"/>
  <wsdl:output message="tns:GetTermSetsSoapOut"/>
</wsdl:operation>
```

The parameters to this operation include the identifiers of the requested term sets and a time stamp for each. If a requested term set has been updated since the specified time stamp, the protocol server MUST return the entire term set. Otherwise, the protocol server MUST return an empty node.<9>

3.1.4.6.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.6.1.1 GetTermSetsSoapIn

The request WSDL message for the **GetTermSets** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetTermSets
```

The SOAP body contains the **GetTermSets** element.

3.1.4.6.1.2 GetTermSetsSoapOut

The response WSDL message for the **GetTermSets** WSDL operation.

The SOAP body contains the **GetTermSetsResponse** element.

3.1.4.6.2 Elements

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

3.1.4.6.2.1 GetTermSets

The input data for the **GetTermSets** WSDL operation.

```
<xs:element name="GetTermSets">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sharedServiceIds" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="termSetIds" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="lcid" type="xs:int"/>
      <xs:element minOccurs="0" maxOccurs="1" name="clientTimeStamps" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="clientVersions" type="xs:string"/>
```

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

sharedServiceIds: The identifiers of the term stores in which the requested term sets reside. The number of **sharedServiceIds** child nodes MUST equal the number of **termSetIds** child nodes and the two sets of child nodes MUST be in one-to-one correspondence so that each **sharedServiceIds** child represents the term store identifier of the corresponding term set in the **termSetIds,** which is an XML encoded string that MUST conform to the schema of the **SspIdsForGetTermSetsClientService** complex type (section 2.2.4.13). An empty string with no term set information MUST be returned if this node is absent.

termSetIds: The term set identifiers of the term sets to be retrieved, which is an XML encoded string that MUST conform to the schema of the **TermSetIdsForGetTermSetsClientService** complex type (section 2.2.4.16). An empty string with no term set information MUST be returned if this node is absent.

lcid: The LCID of the requested language.

clientTimeStamps: The time in ticks used to determine whether or not to return the term set data. This is an XML encoded string that MUST conform to the schema of the **ClientTimeStampsForGetTermSetsClientService** complex type (section 2.2.4.2). The number of **clientTimeStamps** child nodes MUST equal the number of **termSetIds** child nodes and these two sets of child nodes MUST be in one-to-one correspondence. For each term set specified in **termSetIds**, if the term set has been updated since the time specified by the corresponding time stamp in **clientTimeStamps**, the protocol server MUST return the data for the term set. Otherwise, the protocol server MUST return an empty node for the term set. An empty string with no term set information MUST be returned if this node is absent. A protocol client can request that term set data be returned regardless of last update time by specifying the text string "1900-01-01T00:00:00" instead of the integer number for the time stamp (since unparseable values are treated as 0).

clientVersions: The server version numbers that the term set data stored on the client was retrieved from. The number of **clientVersions** child nodes MUST equal the number of termSetIds child nodes and these two sets of child nodes MUST be in one-to-one correspondence. For each term set specified in **termSetIds**, if the term set is already stored on the client it MUST send 1, otherwise it MUST send 0. This is an XML encoded string that MUST conform to the schema of the **ClientVersionsForGetTermSetsClientService** complex type (section 2.2.4.3). An empty string with no term set information MUST be returned if this node is absent.

3.1.4.6.2.2 GetTermSetsResponse

The result data for the **GetTermSets** WSDL operation.

```
<xs:element name="GetTermSetsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetTermSetsResult"
type="GetTermSetsResponseObject"/>
      <xs:element minOccurs="0" maxOccurs="1" name="serverTermSetTimeStampXml"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetTermSetsResult: The data for the requested term sets. As described in the definition of the **clientTimeStamps** element of **GetTermSets**, the returned data for some term sets are empty if the term set has not been updated since the time specified by the corresponding timestamp in clientTimeStamps. The complex type is defined in 2.2.4.4.

serverTermSetTimeStampXml: The time stamps specifying the most recent update times for the requested term sets, which is an XML encoded string that MUST conform to the schema of the **ServerTermSetTimeStampsForGetTermSetsClientService** complex type (section 2.2.4.12).

3.1.4.6.3   Complex Types

None.

3.1.4.6.4   Simple Types

None.

3.1.4.6.5   Attributes

None.

3.1.4.6.6   Groups

None.

3.1.4.6.7   Attribute Groups

None.

3.1.5   Timer Events

None.

3.1.6   Other Local Events

None.

4 Protocol Examples

The following is an example of how a request for a term set that has never been requested before is made.

```
<?xml version="1.0" encoding="utf-8"?>
    <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
        <soap:Body>
            <GetTermSets xmlns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/">
                <sharedServiceIds>&lt;sspIds&gt;&lt;sspId&gt;1b070419-b5a2-4e10-bed8-
a8449b977eac&lt;/sspId&gt;&lt;/sspIds&gt;
                </sharedServiceIds><termSetIds>&lt;termSetIds&gt;&lt;termSetId&gt;755938ac-177c-
4a74-8ef6-45fc2caf8470&lt;/termSetId&gt;&lt;/termSetIds&gt;</termSetIds><lcid>1033</lcid>
                <clientTimeStamps>&lt;dateTimes&gt;&lt;dateTime&gt;1900-01-
01T00:00:00&lt;/dateTime&gt;&lt;/dateTimes&gt;</clientTimeStamps><clientVersions>&lt;versions
&gt;&lt;version&gt;0&lt;/version&gt;&lt;/versions&gt;</clientVersions></GetTermSets>
        </soap:Body>
    </soap:Envelope>
```

This is the response of that request. The term set is serialized and returned because the old update time was passed.

```
<?xml version="1.0" encoding="utf-8"?>
    <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <soap:Body><GetTermSetsResponse
xmlns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/">
            <GetTermSetsResult>&lt;Container&gt;&lt;TermStore&gt;&lt;TS a9="c6baf284-1e99-4650-
b84d-e28794856d21" a17="true" a11="" a16="false" a12="Deleted TermSet" a68="" /&gt;&lt;T
a9="c7c0785f-9c5a-41d9-a1bd-5611f4480e21" a21="false" a61="0"&gt;&lt;LS&gt;&lt;TL a32="Bar"
a31="true" /&gt;&lt;/LS&gt;&lt;DS /&gt;&lt;TMS&gt;&lt;TM a24="c6baf284-1e99-4650-b84d-
e28794856d21" a12="Deleted TermSet" a40="" a17="true" a67="" a45="c7c0785f-9c5a-41d9-a1bd-
5611f4480e21" /&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;T a9="b90e03bd-7e0f-4f27-b960-0d9e4d2ae5af"
a21="false" a61="0"&gt;&lt;LS&gt;&lt;TL a32="Baz" a31="true" /&gt;&lt;/LS&gt;&lt;DS
/&gt;&lt;TMS&gt;&lt;TM a24="c6baf284-1e99-4650-b84d-e28794856d21" a12="Deleted TermSet"
a40="" a17="true" a67="" a45="b90e03bd-7e0f-4f27-b960-0d9e4d2ae5af"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;T a9="1b34a55b-a2c0-46a8-8b1f-77805a72edfa" a21="false"
a61="0"&gt;&lt;LS&gt;&lt;TL a32="myValue" a31="true" /&gt;&lt;/LS&gt;&lt;DS
/&gt;&lt;TMS&gt;&lt;TM a24="c6baf284-1e99-4650-b84d-e28794856d21" a12="Deleted TermSet"
a40="" a17="true" a67="" a45="1b34a55b-a2c0-46a8-8b1f-77805a72edfa"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;/TermStore&gt;&lt;/Container&gt;</GetTermSetsResult>
            <serverTermSetTimeStampXml>&lt;Container&gt;&lt;Node Time="633767110910230000"
TermId="c6baf284-1e99-4650-b84d-e28794856d21"
/&gt;&lt;/Container&gt;</serverTermSetTimeStampXml>
        </GetTermSetsResponse>
        </soap:Body>
    </soap:Envelope>
```

The following is a request/response for adding new terms to a term set. It adds three terms, one directly under the root, one under that new term, and one under a term that already exists.

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

```
        <soap:Body>
            <AddTerms xmlns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/">
                <sharedServiceId>ff579fe0-0f63-41c7-b9f7-0ceba78d2e5f</sharedServiceId>
                <termSetId>97ea1a2d-0eb4-4ac0-acfd-862d8fec7607</termSetId>
                <lcid>1033</lcid>
                <newTerms>&lt;newTerms&gt;&lt;newTerm label=&quot;Under Root&quot;
clientId=&quot;1&quot; parentTermId=&quot;00000000-0000-0000-0000-
000000000000&quot;&gt;&lt;newTerm label=&quot;Under new term&quot; clientId=&quot;2&quot;
parentTermId=&quot;1&quot;/&gt;&lt;/newTerm&gt;&lt;newTerm label=&quot;Under existing
term&quot; clientId=&quot;3&quot; parentTermId=&quot;9884bef8-17e3-4e56-ac3b-
5b86d20a8d4b&quot;/&gt;&lt;/newTerms&gt;</newTerms>
            </AddTerms>
        </soap:Body>
    </soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
    <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <soap:Body>
        <AddTermsResponse xmlns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/">
            <AddTermsResult>&lt;?xml version="1.0" encoding="utf-
16"?&gt;&lt;TermStore&gt;&lt;T a9="3f5dc4ad-9ca2-489c-9444-a84bce1312e1" a21="false"
a61="1"&gt;&lt;LS&gt;&lt;TL a32="Under Root" a31="true" /&gt;&lt;/LS&gt;&lt;DS
/&gt;&lt;TMS&gt;&lt;TM a24="97ea1a2d-0eb4-4ac0-acfd-862d8fec7607" a12="Open" a40=""
a17="true" a67="" a45="3f5dc4ad-9ca2-489c-9444-a84bce1312e1" a69="true"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;T a9="5add558b-10ba-41dc-8b7e-473b807e9044" a21="false"
a61="2"&gt;&lt;LS&gt;&lt;TL a32="Under new term" a31="true" /&gt;&lt;/LS&gt;&lt;DS
/&gt;&lt;TMS&gt;&lt;TM a24="97ea1a2d-0eb4-4ac0-acfd-862d8fec7607" a12="Open" a25="3f5dc4ad-
9ca2-489c-9444-a84bce1312e1" a40="Under Root" a17="true" a67="" a45="3f5dc4ad-9ca2-489c-9444-
a84bce1312e1;5add558b-10ba-41dc-8b7e-473b807e9044" /&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;T
a9="39f10c1a-b8d5-4546-aee8-4d692d5f29ae" a21="false" a61="3"&gt;&lt;LS&gt;&lt;TL a32="Under
existing term" a31="true" /&gt;&lt;/LS&gt;&lt;DS /&gt;&lt;TMS&gt;&lt;TM a24="97ea1a2d-0eb4-
4ac0-acfd-862d8fec7607" a12="Open" a25="9884bef8-17e3-4e56-ac3b-5b86d20a8d4b" a40="Bar"
a17="true" a67="" a45="9884bef8-17e3-4e56-ac3b-5b86d20a8d4b;39f10c1a-b8d5-4546-aee8-
4d692d5f29ae" /&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;/TermStore&gt;</AddTermsResult>
        </AddTermsResponse>
      </soap:Body>
    </soap:Envelope>
```

The following is a request/response for querying for terms that begin with "un".

```
<?xml version="1.0" encoding="utf-8"?>
    <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
      <soap:Body>
        <GetTermsByLabel xmlns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/">
            <label>un</label>
            <lcid>1033</lcid>
            <matchOption>StartsWith</matchOption>
            <resultCollectionSize>40</resultCollectionSize>
            <addIfNotFound>false</addIfNotFound>
        </GetTermsByLabel>
      </soap:Body>
    </soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
   <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <soap:Body>
         <GetTermsByLabelResponse
xmlns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/">
            <GetTermsByLabelResult>&lt;?xml version="1.0" encoding="utf-
16"?&gt;&lt;TermStore&gt;&lt;T a9="39f10c1a-b8d5-4546-aee8-4d692d5f29ae" a21="false"
a61="0"&gt;&lt;LS&gt;&lt;TL a32="Under existing term" a31="true" /&gt;&lt;/LS&gt;&lt;DS
/&gt;&lt;TMS&gt;&lt;TM a24="97ea1a2d-0eb4-4ac0-acfd-862d8fec7607" a12="Open" a15="0"
a25="9884bef8-17e3-4e56-ac3b-5b86d20a8d4b" a40="Bar" a17="true"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;T a9="5add558b-10ba-41dc-8b7e-473b807e9044" a21="false"
a61="0"&gt;&lt;LS&gt;&lt;TL a32="Under new term" a31="true" /&gt;&lt;/LS&gt;&lt;DS
/&gt;&lt;TMS&gt;&lt;TM a24="97ea1a2d-0eb4-4ac0-acfd-862d8fec7607" a12="Open" a15="0"
a25="3f5dc4ad-9ca2-489c-9444-a84bce1312e1" a40="Under Root" a17="true"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;T a9="3f5dc4ad-9ca2-489c-9444-a84bce1312e1" a21="false"
a61="0"&gt;&lt;LS&gt;&lt;TL a32="Under Root" a31="true" /&gt;&lt;/LS&gt;&lt;DS
/&gt;&lt;TMS&gt;&lt;TM a24="97ea1a2d-0eb4-4ac0-acfd-862d8fec7607" a12="Open" a15="0"
a25="00000000-0000-0000-0000-000000000000" a40="" a17="true"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;/TermStore&gt;</GetTermsByLabelResult>
         </GetTermsByLabelResponse>
      </soap:Body>
   </soap:Envelope>
```

The following is a request/response for querying for keyword terms by GUID:

```
<?xml version="1.0" encoding="utf-8"?>
   <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
      <soap:Body>
         <GetKeywordTermsByGuids
xmlns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/">
<termIds>&lt;termIds&gt;&lt;termId&gt;5add558b-10ba-41dc-8b7e-
473b807e9044&lt;/termId&gt;&lt;/termIds&gt;</termIds>
            <lcid>1033</lcid>
         </GetKeywordTermsByGuids>
      </soap:Body>
   </soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
   <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <soap:Body>
         <GetKeywordTermsByGuidsResponse
xmlns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/">
            <GetKeywordTermsByGuidsResult>&lt;?xml version="1.0" encoding="utf-
16"?&gt;&lt;TermStore&gt;&lt;T a9="5add558b-10ba-41dc-8b7e-473b807e9044" a21="false"
a61="2"&gt;&lt;LS&gt;&lt;TL a32="Under new term" a31="true" /&gt;&lt;/LS&gt;&lt;DS
/&gt;&lt;TMS&gt;&lt;TM a24="97ea1a2d-0eb4-4ac0-acfd-862d8fec7607" a12="Open" a25="3f5dc4ad-
9ca2-489c-9444-a84bce1312e1" a40="Under Root" a17="true" a67="" a45="3f5dc4ad-9ca2-489c-9444-
a84bce1312e1;5add558b-10ba-41dc-8b7e-473b807e9044"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;/TermStore&gt;</GetKeywordTermsByGuidsResult>
```

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

5 Security

5.1 Security Considerations for Implementers

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

5.2 Index of Security Parameters

None.

Release: Sunday, December 19, 2010

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided below.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:s1="http://microsoft.com/wsdl/types/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://schemas.microsoft.com/sharepoint/taxonomy/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">This web service is
designed for the use of client applications.</wsdl:documentation>
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/taxonomy/soap/">
      <s:import namespace="http://microsoft.com/wsdl/types/" />
      <s:element name="GetTermSets">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="sharedServiceIds" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="termSetIds" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="lcid" type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="clientTimeStamps" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="clientVersions" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetTermSetsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetTermSetsResult" type="s:string"
/>
            <s:element minOccurs="0" maxOccurs="1" name="serverTermSetTimeStampXml"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetTermsByLabel">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="label" type="s:string"/>
            <s:element minOccurs="1" maxOccurs="1" name="lcid" type="s:int"/>
            <s:element minOccurs="1" maxOccurs="1" name="matchOption"
type="tns:MatchOptions"/>
            <s:element minOccurs="1" maxOccurs="1" name="resultCollectionSize" type="s:int"/>
            <s:element minOccurs="0" maxOccurs="1" name="termIds" type="s:string"/>
            <s:element minOccurs="1" maxOccurs="1" name="addIfNotFound" type="s:boolean"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:simpleType name="MatchOption">
        <s:restriction base="s:string">
```

```
              <s:enumeration value="StartsWith" />
              <s:enumeration value="ExactMatch" />
            </s:restriction>
        </s:simpleType>
        <s:element name="GetTermsByLabelResponse">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="0" maxOccurs="1" name="GetTermsByLabelResult"
  type="s:string" />
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="AddTerms">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="1" maxOccurs="1" name="sharedServiceId" type="s1:guid" />
              <s:element minOccurs="1" maxOccurs="1" name="termSetId" type="s1:guid" />
              <s:element minOccurs="1" maxOccurs="1" name="lcid" type="s:int" />
              <s:element minOccurs="0" maxOccurs="1" name="newTerms" type="s:string" />
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="AddTermsResponse">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="0" maxOccurs="1" name="AddTermsResult" type="s:string" />
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="GetChildTermsInTerm">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="1" maxOccurs="1" name="sspId" type="s1:guid"/>
              <s:element minOccurs="1" maxOccurs="1" name="lcid" type="s:int"/>
              <s:element minOccurs="1" maxOccurs="1" name="termId" type="s1:guid"/>
              <s:element minOccurs="1" maxOccurs="1" name="termSetId" type="s1:guid"/>
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="GetChildTermsInTermResponse">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="0" maxOccurs="1" name="GetChildTermsInTermResult"
  type="s:string"/>
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="GetChildTermsInTermSet">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="1" maxOccurs="1" name="sspId" type="s1:guid"/>
              <s:element minOccurs="1" maxOccurs="1" name="lcid" type="s:int"/>
              <s:element minOccurs="1" maxOccurs="1" name="termSetId" type="s1:guid"/>
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="GetChildTermsInTermSetResponse">
          <s:complexType>
            <s:sequence>
```

```
            <s:element minOccurs="0" maxOccurs="1" name="GetChildTermsInTermSetResult"
type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetKeywordTermsByGuids">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="termIds" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="lcid" type="s:int" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetKeywordTermsByGuidsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetKeywordTermsByGuidsResult"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/wsdl/types/">
      <s:simpleType name="guid">
        <s:restriction base="s:string">
          <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-
9a-fA-F]{12}" />
        </s:restriction>
      </s:simpleType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="GetTermSetsSoapIn">
    <wsdl:part name="parameters" element="tns:GetTermSets" />
  </wsdl:message>
  <wsdl:message name="GetTermSetsSoapOut">
    <wsdl:part name="parameters" element="tns:GetTermSetsResponse" />
  </wsdl:message>
  <wsdl:message name="GetTermsByLabelSoapIn">
    <wsdl:part name="parameters" element="tns:GetTermsByLabel" />
  </wsdl:message>
  <wsdl:message name="GetTermsByLabelSoapOut">
    <wsdl:part name="parameters" element="tns:GetTermsByLabelResponse" />
  </wsdl:message>
  <wsdl:message name="AddTermsSoapIn">
    <wsdl:part name="parameters" element="tns:AddTerms" />
  </wsdl:message>
  <wsdl:message name="AddTermsSoapOut">
    <wsdl:part name="parameters" element="tns:AddTermsResponse" />
  </wsdl:message>
  <wsdl:message name="GetChildTermsInTermSoapIn">
    <wsdl:part name="parameters" element="tns:GetChildTermsInTerm" />
  </wsdl:message>
  <wsdl:message name="GetChildTermsInTermSoapOut">
    <wsdl:part name="parameters" element="tns:GetChildTermsInTermResponse" />
  </wsdl:message>
  <wsdl:message name="GetChildTermsInTermSetSoapIn">
    <wsdl:part name="parameters" element="tns:GetChildTermsInTermSet" />
  </wsdl:message>
```

```
<wsdl:message name="GetChildTermsInTermSetSoapOut">
  <wsdl:part name="parameters" element="tns:GetChildTermsInTermSetResponse" />
</wsdl:message>
<wsdl:message name="GetKeywordTermsByGuidsSoapIn">
  <wsdl:part name="parameters" element="tns:GetKeywordTermsByGuids" />
</wsdl:message>
<wsdl:message name="GetKeywordTermsByGuidsSoapOut">
  <wsdl:part name="parameters" element="tns:GetKeywordTermsByGuidsResponse" />
</wsdl:message>
<wsdl:portType name="Taxonomy_x0020_web_x0020_serviceSoap">
  <wsdl:operation name="GetTermSets">
    <wsdl:input message="tns:GetTermSetsSoapIn" />
    <wsdl:output message="tns:GetTermSetsSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetTermsByLabel">
    <wsdl:input message="tns:GetTermsByLabelSoapIn" />
    <wsdl:output message="tns:GetTermsByLabelSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="AddTerms">
    <wsdl:input message="tns:AddTermsSoapIn" />
    <wsdl:output message="tns:AddTermsSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetChildTermsInTerm">
    <wsdl:input message="tns:GetChildTermsInTermSoapIn"/>
    <wsdl:output message="tns:GetChildTermsInTermSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetChildTermsInTermSet">
    <wsdl:input message="tns:GetChildTermsInTermSetSoapIn"/>
    <wsdl:output message="tns:GetChildTermsInTermSetSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetKeywordTermsByGuids">
    <wsdl:input message="tns:GetKeywordTermsByGuidsSoapIn" />
    <wsdl:output message="tns:GetKeywordTermsByGuidsSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="Taxonomy_x0020_web_x0020_serviceSoap"
type="tns:Taxonomy_x0020_web_x0020_serviceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="GetTermSets">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetTermSets"
style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetTermsByLabel">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetTermsByLabel"
style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
```

```
    </wsdl:operation>
    <wsdl:operation name="AddTerms">
     <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/AddTerms" style="document"
/>
     <wsdl:input>
       <soap:body use="literal" />
     </wsdl:input>
     <wsdl:output>
       <soap:body use="literal" />
     </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetChildTermsInTerm">
     <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetChildTermsInTerm"
style="document" />
     <wsdl:input>
       <soap:body use="literal" />
     </wsdl:input>
     <wsdl:output>
       <soap:body use="literal" />
     </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetChildTermsInTermSet">
     <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetChildTermsInTermSet"
style="document" />
     <wsdl:input>
       <soap:body use="literal" />
     </wsdl:input>
     <wsdl:output>
       <soap:body use="literal" />
     </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetKeywordTermsByGuids">
     <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetKeywordTermsByGuids"
style="document" />
     <wsdl:input>
       <soap:body use="literal" />
     </wsdl:input>
     <wsdl:output>
       <soap:body use="literal" />
     </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="Taxonomy_x0020_web_x0020_serviceSoap12"
type="tns:Taxonomy_x0020_web_x0020_serviceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetTermSets">
     <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetTermSets"
style="document" />
     <wsdl:input>
       <soap12:body use="literal" />
     </wsdl:input>
     <wsdl:output>
       <soap12:body use="literal" />
     </wsdl:output>
    </wsdl:operation>
```

```
    <wsdl:operation name="GetTermsByLabel">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetTermsByLabel"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddTerms">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/AddTerms" style="document"
/>
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetKeywordTermsByGuids">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/taxonomy/soap/GetKeywordTermsByGuids"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

7   Appendix B: Product Behavior

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

- Microsoft® SharePoint® Server 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 3.1.4.1.2.1: If an empty GUID is specified but the default keywords term store is undefined or inaccessible, the Office 2010 protocol server raises a null reference exception. There can be multiple term stores associated with a site collection. One of them can be specified as the default keywords term store. If none is specified, the default keywords term store is undefined.

<2> Section 3.1.4.1.2.1: If the term set does not exist the Office 2010 protocol server raises a null reference exception.

<3> Section 3.1.4.2.2.1: If the term store is not reachable, the Office 2010 protocol server raises an argument-out-of-range exception.

<4> Section 3.1.4.2.2.1: If the term does not exist, the Office 2010 protocol raises an argument-out-of-range exception.

<5> Section 3.1.4.2.2.1: If the term set does not exist, the Office 2010 protocol server raises an argument-out-of-range exception.

<6> Section 3.1.4.3.2.1: If the term store is not reachable, the Office 2010 protocol server raises an argument-out-of-range exception.

<7> Section 3.1.4.3.2.1: If the term set does not exist, the Office 2010 protocol server raises a null reference exception.

<8> Section 3.1.4.5.2.1: When attempting to add unfound terms, the Office 2010 protocol server raises a null reference exception, if the default keywords term store is not accessible or not defined.

<9> Section 3.1.4.6: The Office 2010 protocol server raises a null reference exception if it cannot find a requested term set.

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