[MS-SITED3S]:

Site Data 2003 Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation 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 might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
6/27/2008	1.0	Major	Revised and edited the technical content
12/14/2008	1.01	Editorial	Revised and edited the technical content
7/13/2009	1.02	Major	Revised and edited the technical content
8/28/2009	1.03	Editorial	Revised and edited the technical content
11/6/2009	1.04	Editorial	Revised and edited the technical content
2/19/2010	2.0	Editorial	Revised and edited the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Major	Updated and revised the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	2.5	Minor	Clarified the meaning of the technical content.
4/11/2012	2.5	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.5	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	2.5	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.5	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	2.5	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	2.5	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
11/18/2013	2.5	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.5	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.5	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	2.5	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	2.5	None	No changes to the meaning, language, or formatting of the technical content.
6/23/2016	2.5	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Site Data 2003 Web Service Protocol. This protocol enables a set of server extensions that are used to augment a basic HTTP server so that it supports full and incremental crawling. This protocol is intended to be used by an index server application and not by a user directly through a Web browser.

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

1.1 Glossary

This document uses the following terms:

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

attachment: An external file that is included with an Internet message or associated with an item in a SharePoint list.

authentication: The act of proving an identity to a server while providing key material that binds the identity to subsequent communications.

Boolean: An operation or expression that can be evaluated only as either true or false.

bucket web: A site that is used to store content for a specific category (1).

complex type: An element that can contain other elements or attributes and appears as <complexType> in an XML document. See also **simple type**.

content type: A named and uniquely identifiable collection of settings and fields that store metadata for individual items in a SharePoint list. One or more content types can be associated with a list, which restricts the contents to items of those types.

context: A collection of context properties that describe an execution environment.

context site: A site that corresponds to the context of the current request.

Coordinated Universal Time (UTC): A high-precision atomic time standard that approximately tracks Universal Time (UT). It is the basis for legal, civil time all over the Earth. Time zones around the world are expressed as positive and negative offsets from UTC. In this role, it is also referred to as Zulu time (Z) and Greenwich Mean Time (GMT). In these specifications, all references to UTC refer to the time at UTC-0 (or GMT).

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

data source: A database, web service, disk, file, or other collection of information from which data is queried or submitted. Supported data sources vary based on application and data provider.

datetime: A data type that represents the date and time when a document can be normalized and indexed as a numeric value by a search application. The range and degree of granularity varies according to search application and implementation.

default view: The layout and organization of a document or list that appears automatically when users open that document or display that list.

discussion board: A list in which users can read, post, and reply to messages from other users who are members of the same discussion board.

document: An object in a content database such as a file, folder, **list**, or **site**. Each object is identified by a URI.

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

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

external security provider: An external object that manages permissions on a site.

file: A single, discrete unit of content.

flags: A set of values used to configure or report options or settings.

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

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

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

index server: A server that is assigned the task of crawling.

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

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

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

list template: An XML-based definition of list settings, including fields and views, and optionally list items. List templates are stored in .stp files in the content database.

localization: The process of adapting an application or documentation, including text and non-text elements, to meet the language, cultural, and political expectations and requirements of a specific geographic country or region.

parent site: The site that is above the current site in the hierarchy of the site collection.

permission: A rule that is associated with an object and that regulates which users can gain access to the object and in what manner. See also rights.

personal site: A type of SharePoint site that is used by an individual user for personal productivity. The site appears to the user as My Site.

picture library: A type of **document library** that is optimized for storing digital pictures or graphics.

role definition: A named set of permissions for a SharePoint site. See also permission level.

root folder: The folder at the top of a hierarchy of folders in a list.

server-relative URL: A relative URL that does not specify a scheme or host, and assumes a base URI of the root of the host, as described in [RFC3986].

simple type: An element that can contain only text and appears as <simpleType> in an XML document or any attribute of an element. Attributes are considered simple types because they contain only text. See also **complex type**.

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

site collection: A set of **websites** that are in the same content database, have the same owner, and share administration settings. A site collection can be identified by a **GUID** or the **URL** of the top-level site for the site collection. Each site collection contains a top-level site, can contain one or more subsites, and can have a shared navigational structure.

site membership: The status of being a member of a site and having a defined set of user rights for accessing or managing content on that site.

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

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

SOAP envelope: A container for **SOAP message** information and the root element of a SOAP document. See [SOAP1.2-1/2007] section 5.1 for more information.

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

SOAP message: An **XML** document consisting of a mandatory **SOAP envelope**, an optional SOAP header, and a mandatory **SOAP body**. See [SOAP1.2-1/2007] section 5 for more information.

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

subsite: A complete website that is stored in a named subdirectory of another website. The parent website can be the top-level site of a site collection or another subsite. Also referred to as subweb.

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

view: See form view (Microsoft InfoPath), list view (SharePoint Products and Technologies), or View (Microsoft Business Connectivity Services).

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

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

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

XML: The Extensible Markup Language, as described in [XML1.0].

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

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [XML].

XML node: The smallest unit of a valid, complete structure in an XML document. For example, a node can represent an element, an attribute (1), or a text string.

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

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

1.2 References

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

1.2.1 Normative References

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

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification".

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

[MS-SITEDATS] Microsoft Corporation, "Site Data Web Service Protocol".

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

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

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

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

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

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

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

[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

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

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

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

[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

[KNUTH] Knuth, Donald E., "The Art of Computer Programming Vol 1. 3rd ed.", Boston: Addison-Wesley, (1997), ISBN 0-201-89683-4.

1.3 Protocol Overview (Synopsis)

The purpose of this protocol is to support **site** crawling by an external **index server**. The purpose, techniques, and internal data structures used by an index server are beyond the scope of this protocol. However, this protocol does depend on the structure of the site content and the behavior of the protocol server.

This protocol supports an index server or similar client applications that follow the site crawling process to traverse a site where the content conforms to the site structure.

Site Structure

This protocol requires the site to present content in a hierarchical structure. It requires the protocol server to track the details about common **content types** such as pages, **lists**, **list items**, **documents**, and document libraries.

The index server or any other client can use the protocol's methods to traverse the site structure, explore the list item fields, or retrieve **document library items** or list item **attachments**.

Site Crawling Process

This protocol is designed to support an index server or similar client application that conducts a scan of the site content following the recommended site crawling process. This process is described in detail in section 3.1.4.

The site crawling process assumes that the index server is configured with the **URL** of the site to index and that the site supports this protocol.

First, the index server establishes the crawling **context**. Details of context establishment are discussed in section 3.1.4.

Next, the index server proceeds with a traversal of the site. Traversal requires identifying all **subsites**, identifying all lists and document libraries, and scanning all list items and documents to peruse their content. This protocol exposes many operations supporting traversal including **GetSite**, **GetWeb**, and **GetListCollection**. These operations are described in detail in section 3.1.4.

Note that the site can store the documents in a variety of proprietary formats as well as in many languages. This protocol itself allows the index server to enumerate site objects or retrieve the document content itself.

When the crawling process completes the traversal of all site content, it reaches the "complete" state.

The site content might undergo changes before the index server completes a traversal of the site content. In this case, the index can be already out of date with respect to the content.

1.4 Relationship to Other Protocols

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

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

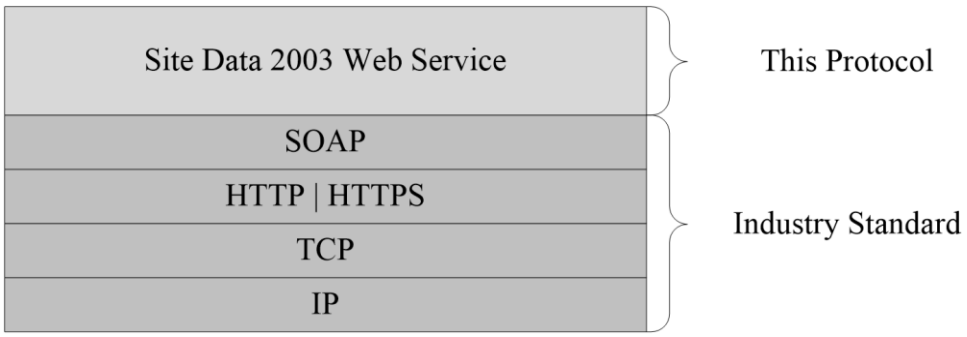

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a site that is identified by a URL that is known by protocol clients. The protocol server **endpoint** is formed by appending "_vti_bin/sitedata.asmx" to the URL of the site. For example, http://www.contoso.com/Repository/_vti_bin/SiteData.asmx.

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

1.6 Applicability Statement

This protocol supports the crawling of **Web sites**, which follow a hierarchical pattern of content organization defined in section 1.3. The protocol also supports tracking the changes of the structural elements.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported transports:** This protocol uses multiple transports with SOAP as described in section 2.1.
- **Localization:** This protocol includes text strings in various messages. **Localization** considerations for such strings are described in sections 2.2 and 3.1.4.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

This protocol does not use any standards assignments other than those of HTTP 1.1, as described in [RFC2616].

2 Messages

2.1 Transport

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

Protocol messages MUST be formatted as specified either in [SOAP1.1] section 4 or in [SOAP1.2/1] section 5. Protocol server faults MUST be returned either using HTTP Status Codes as specified in [RFC2616] section 10 or using **SOAP faults** as specified either in [SOAP1.1] section 4.4 or in [SOAP1.2/1] section 5.4.

2.2 Common Message Syntax

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

2.2.1 Namespaces

This protocol specifies and references **XML namespaces** using the mechanisms specified in [XMLNS]. Although this document associates an **XML namespace prefix** for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability. The following table describes these namespaces.

Prefix	Namespace URI	Reference
Soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
Tns	http://schemas.microsoft.com/sharepoint/soap/	
S	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
(none)	http://schemas.microsoft.com/sharepoint/soap/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
rs	urn:schemas-microsoft-com:rowset	[MS-PRSTFR]
xdr	uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882	[MS-PRSTFR]
dt	uuid:C2F41010-65B3-11d1-A29F-00AA00C14882	[MS-PRSTFR]
core	http://schema.microsoft.com/sharepoint/wire/	[MS-WSSCAML]

2.2.2 Messages

None.

2.2.3 Elements

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

2.2.4 Complex Types

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

Complex type	Description
_sFPUrl	The details about a direct subfolder, document, or page in a **folder**.
_sWebWithTime	The URL and last modified information about a site.
ArrayOf_sFPUrl	An array of **_sFPUrl** elements.
ArrayOf_sWebWithTime	An array of **_sWebWithTime** elements.
ArrayOfString	An array of arbitrary strings (their meaning and format is dependent on the operation).
GroupDescription	The identifier, name, and owner of the **group**.
GroupMembership	For each group, this element contains the group description and the users who belong to this group.
Groups	Specifies an array of group information.
Permission	The rights for a user or group defined by using a data pair in the format of {memberId, **permissions** granted to this member} where **memberId** represents the user or group.
UserDescription	The details about a User.
Users	An array of **UserDescription** elements

2.2.4.1 _sFPUrl

This complex type is identical to [MS-SITEDATS] section 2.2.4.1.

2.2.4.2 _sWebWithTime

This complex type is identical to [MS-SITEDATS] section 2.2.4.2.

2.2.4.3 ArrayOf_sFPUrl

This complex type is identical to [MS-SITEDATS] section 2.2.4.4.

2.2.4.4 ArrayOf_sWebWithTime

This complex type is identical to [MS-SITEDATS] section 2.2.4.5.

2.2.4.5 ArrayOfString

This complex type is identical to [MS-SITEDATS] section 2.2.4.6.

2.2.4.6 GroupDescription

This complex type is identical to [MS-SITEDATS] section 2.2.4.9.

2.2.4.7 GroupMembership

This complex type is identical to [MS-SITEDATS] section 2.2.4.10.

2.2.4.8 Groups

This complex type is identical to [MS-SITEDATS] section 2.2.4.11.

2.2.4.9 Permission

This complex type is identical to [MS-SITEDATS] section 2.2.4.16.

2.2.4.10 UserDescription

This complex type is identical to [MS-SITEDATS] section 2.2.4.23.

2.2.4.11 Users

This complex type is identical to [MS-SITEDATS] section 2.2.4.24.

2.2.5 Simple Types

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

Simple type	Description
ListBaseType	The type of list.
ListBaseTemplate	The type of list templates.
TrueFalseType	A replacement for the **Boolean** type, having values **true** and **false**.

2.2.5.1 ListBaseType

This simple type is identical to [MS-SITEDATS] section 2.2.5.1.

2.2.5.2 ListBaseTemplate

The **ListBaseTemplate** defines an enumeration of the templates of a list. Its schema is as follows. Note that this enumeration is not exhaustive, and may vary depending on installed customizations.

```
<s:simpleType name="ListBaseTemplate">
  <s:restriction base="s:string">
    <s:enumeration value="InvalidType"/>
    <s:enumeration value="GenericList"/>
    <s:enumeration value="DocumentLibrary"/>
    <s:enumeration value="Survey"/>
    <s:enumeration value="Links"/>
```

```
      <s:enumeration value="Announcements"/>
      <s:enumeration value="Contacts"/>
      <s:enumeration value="Events"/>
      <s:enumeration value="Tasks"/>
      <s:enumeration value="DiscussionBoard"/>
      <s:enumeration value="PictureLibrary"/>
      <s:enumeration value="DataSources"/>
      <s:enumeration value="WebTemplateCatalog"/>
      <s:enumeration value="WebPartCatalog"/>
      <s:enumeration value="ListTemplateCatalog"/>
      <s:enumeration value="XMLForm"/>
      <s:enumeration value="CustomGrid"/>
      <s:enumeration value="IssueTracking"/>
    </s:restriction>
  </s:simpleType>
```

The following table defines the allowable values for **ListBaseTemplate**.

Value	Meaning
InvalidType	An invalid list template.
GenericList	A generic list template.
DocumentLibrary	A document library list template.
Survey	A survey list template.
Links	A links list template.
Announcements	An announcements list template.
Contacts	A contacts list template.
Events	An events list template.
Tasks	A tasks list template.
DiscussionBoard	A **discussion board** list template.
PictureLibrary	A **picture library** list template.
DataSources	A **data sources** list template.
WebTemplateCatalog	A Web template catalog list template.
WebPartCatalog	A **Web Part** gallery list template.
ListTemplateCatalog	A **list template** catalog list template.
XMLForm	An **XML** form list template.
CustomGrid	A custom grid list template.
IssueTracking	An issue tracking list template.

2.2.5.3 TrueFalseType

This simple type is identical to [MS-SITEDATS] section 2.2.5.4.

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 attribute definitions.

2.2.8 Attribute Groups

This specification does not define any common XML Schema attribute 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 Site Data Soap Server Details

3.1.1 Abstract Data Model

This protocol supports crawling sites that conform to a hierarchical pattern of content organization. This section specifies the hierarchical pattern in detail.

The elements of this hierarchy include Web pages, list items inside of various lists, and annotated documents inside of document libraries. The elements should conform to the following organization:

- A root site, also called a **site collection**, consists of subsites or sites that are sometimes called "Webs." Both terms denote the same conceptual entity, but "site" emphasizes the unit of administrative management and User community while "Web" is the unit of content organization. Each site corresponds to one Web called the "top-level Web site."
- A site can contain other subsites.
- Sites contain both predefined and custom lists. Each list has a schema that prescribes the fields in every list item.
- Lists contain list items. Each list item has the same fields (shown as "Columns") prescribed by the list schema. Optionally, list items can have any number of attachments. There are predefined fields in every list. For example, ID and Last Modified Time. Those fields obtain their values automatically and have predefined meanings.
- Document libraries are a special form of list. They contain documents annotated with fields prescribed by the document library schema.
- Lists can contain folders and folders can contain subfolders. Folders help end users to visually organize contained material.

All elements of a site are identifiable either by a URL or by their identifiers, which are typically **GUIDs**. Identifiers might not be human-readable, but they are immutable for the whole life-cycle of an element. In contrast, the URL might change over time. For example, a list or document can be renamed and that would, in turn, change the URL.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

This protocol has the operations described in the following table.

Operation	Description
EnumerateFolder	Returns items and subfolders of a given folder.
GetAttachments	Returns attachments to a given list item.
GetList	Returns general information, the schema of the fields, and access right permissions for a given list.
GetListCollection	Returns general information about all lists of a given site.
GetListItems	Returns all or the desired fields of all list items satisfying certain criteria in a given list.
GetSite	Returns information about the site collection of the **context site**.
GetSiteAndWeb	Extracts the fragments of a URL which refer to the site as a whole and the particular Web where the URL belongs.
GetURLSegments	Given a URL of some structural element, extracts the identifiers, URLs, or both elements of the containing structural elements.
GetWeb	Returns information about the context site.

Each method is a **SOAP operation**, as specified in [SOAP1.1], that sends parameters that are organized into an input XML message and returns a set of values that are combined in a response or output XML message. The protocol server never initiates any communication with the protocol client. All communication is transported over HTTP or HTTPS as specified in [RFC2616], section 9.1.

Method calls are sent as HTTP posts with **SOAP action** headers denoting the method and the body containing the input message that are wrapped in a **SOAP envelope**. The responses are sent as an output message in the body of an HTTP response and have the content type: text/xml and are wrapped in a SOAP envelope. All posts are made to a well-known URL on the protocol server. For example, http://root/_vti_bin/SiteData.asmx, where the root denotes a root URL of a site or subsite.

Crawling begins with the establishment of context. Assume that all the crawling agent knows is the URL of a page from the site content. For example:

```
http://www.fabricam.com/Subsite/Shared%20Documents/Forms/AllItems.aspx
```

The following sequence diagram depicts the sequence of the required operations for full traversal. A brief explanation of each message follows. Note that the following figure explains only one branch of depth-first [KNUTH] traversal of a site. For example, having found a list among others returned by the **GetListCollection** response, the protocol explores its list items. Complete depth-first traversal enumerates all the lists first, and then explores the list items of each list.

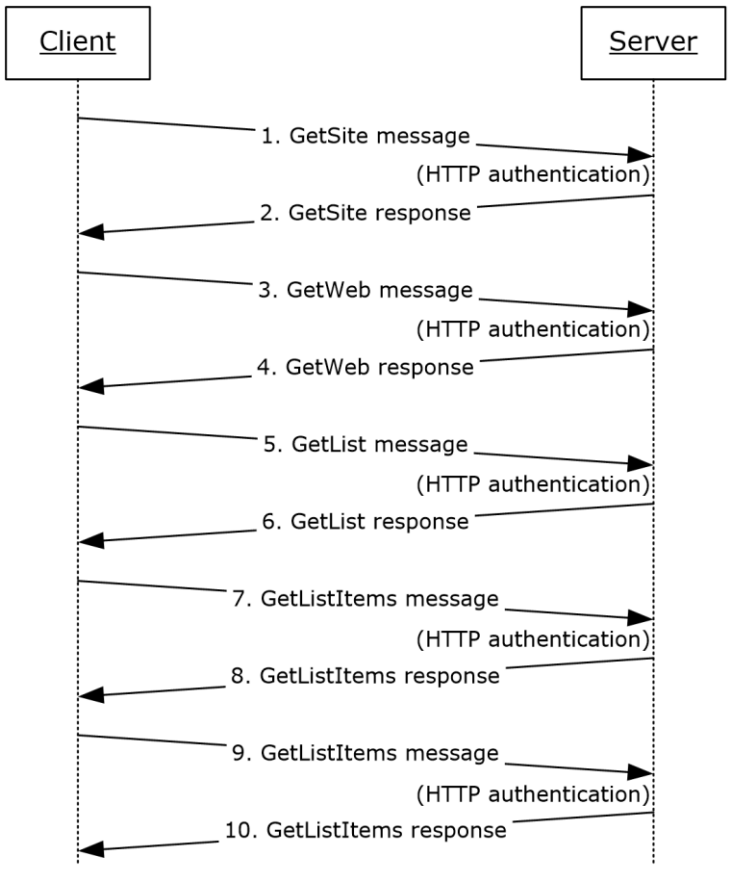

Figure 2: Outline of site traversal scenario

The following occurs:

1. The protocol client sends a **GetSite** input message with any site-referring URL as a parameter to get the URL for the site collection and the site referred to by the URL.

2. The protocol server sends a response message containing those URLs.

3. The protocol client sends a **GetWeb** request message targeting a web site of interest in the site collection. For example, http://www.fabricam.com.

4. The protocol server sends a response message that contains the list of all subsites (child objects) and lists of the web site targeted by the request. In this example, http://www.fabricam.com/subsite.

5. To explore the first list in the depth-first traversal, the protocol client sends a **GetList** message, passing this subsite URL and list identifier as parameters.

6. The protocol server sends the **GetList** response message containing information about the list, including when it was created, when it was last modified, the permission settings, and the list schema.

7. To obtain information about all the list items, the protocol client sends a **GetListItems** input message with an empty *sQuery* parameter (meaning "all items ").

8. The protocol server sends a **GetListItems** response message, enumerating all list items.

9. To obtain information about each list item, the protocol client sends a **GetListItems** input message again to get the list item properties.

10. The protocol server sends a **GetListItems** response message for list item properties.

The protocol client can now inspect all fields of those list items to build the index. The details of building the index are outside of the scope of this protocol.

Full details of individual Site Data Protocol operations are specified in sections 3.1.4.1 through 3.1.4.9.

3.1.4.1 EnumerateFolder

The **EnumerateFolder** operation is used to enumerate the contents of a folder. This operation provides information about the immediate child folders, documents, and pages in a folder.

```
<wsdl:operation name="EnumerateFolder">
    <wsdl:input message="EnumerateFolderSoapIn" />
    <wsdl:output message="EnumerateFolderSoapOut" />
</wsdl:operation>
```

The protocol client sends an **EnumerateFolderSoapIn** request message and the protocol server responds with an **EnumerateFolderSoapOut** response message.

3.1.4.1.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.1.1.1 EnumerateFolderSoapIn

The **EnumerateFolderSoapIn** request message enumerates the contents of the folder.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/EnumerateFolder
```

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

3.1.4.1.1.2 EnumerateFolderSoapOut

The **EnumerateFolderSoapOut** response message enumerates the contents of the folder.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/EnumerateFolder
```

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

3.1.4.1.2 Elements

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

3.1.4.1.2.1 EnumerateFolder

The **EnumerateFolder** element specifies the information of the folder the client wants to enumerate.

```
<s:element name="EnumerateFolder">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="strFolderUrl" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

strFolderUrl: A string that specifies the URL of the folder. The URL MUST be one of the following:

- An absolute URL of the folder in context site.
- A relative URL of the folder with respect to the context site.
- Empty. If **strFolderUrl** is empty, the protocol server MUST consider the requested folder to be the **root folder** of the context site.

3.1.4.1.2.2 EnumerateFolderResponse

The **EnumerateFolderResponse** element specifies the result of the EnumerateFolder operation.

```
<s:element name="EnumerateFolderResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="EnumerateFolderResult"
                 type="s:unsignedInt" />
      <s:element minOccurs="0" maxOccurs="1" name="vUrls"
                 type="tns:ArrayOf_sFPUrl" />
    </s:sequence>
  </s:complexType>
</s:element>
```

EnumerateFolderResult: An unsigned integer that specifies the status of the operation. On successful completion of the operation, the protocol server MUST set it to 0.

vUrls: An **ArrayOf_sFPUrl** (section 2.2.4.3) complex types that specify URL information about immediate subfolders, documents, and pages of a folder. If there are no subfolders, this element MUST not be present.

If the protocol server encounters an error during the execution of this operation, a SOAP fault MUST be returned. If the prefix of the absolute URL specified in **strFolderUrl** does not contain the context site, the protocol server MUST return a SOAP fault with the error message, "The folder that would hold URL strFolderUrl does not exist on the server".

3.1.4.2 GetAttachments

The **GetAttachments** operation is used to get information about all the attachments of a list item.

```
<wsdl:operation name="GetAttachments">
    <wsdl:input message="GetAttachmentsSoapIn" />
    <wsdl:output message="GetAttachmentsSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetAttachmentsSoapIn** request message and the protocol server responds with a **GetAttachmentsSoapOut** response message.

3.1.4.2.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.2.1.1 GetAttachmentsSoapIn

The **GetAttachmentSoapIn** request message gets all the attachments belonging to a list item.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/GetAttachments
```

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

3.1.4.2.1.2 GetAttachmentsSoapOut

The **GetAttachmentSoapOut** response message gets all the attachments belonging to a list item.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/GetAttachments
```

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

3.1.4.2.2 Elements

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

3.1.4.2.2.1 GetAttachments

The **GetAttachments** element specifies the list item from which the client is retrieving the attachments information.

```
<s:element name="GetAttachments">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="strListName" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="strItemId" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

strListName: A string that specifies the name of the list or the GUID of the list. The GUID of the list MUST be enclosed in curly braces<1> ({}).

strItemId: A string that specifies the index of the list item in the list, where 1 identifies the first item in the list.

3.1.4.2.2.2 GetAttachmentsResponse

The **GetAttachmentsResponse** element specifies the results from the **GetAttachments** operation.

```
<s:element name="GetAttachmentsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetAttachmentsResult"
                 type="s:unsignedInt" />
      <s:element minOccurs="0" maxOccurs="1" name="vAttachments"
                 type="tns:ArrayOfString" />
```

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

GetAttachmentsResult: An unsigned integer that specifies the status of the operation. On successful completion of the operation the protocol server MUST set it to 0.

vAttachments: An **ArrayOfString** complex type (section 2.2.4.5) that specifies an array of URLs of all the attachments belonging to a list item. Each string in the array contains an absolute URL of an attachment. If there are no attachments, this element MUST not be present.

If the protocol server encounters an error during the execution of this operation, a SOAP fault MUST be returned that contains one of the error messages described in the following table.

Error code	Error message	Error condition
0x80090007	The GUID should contain 32 digits with 4 dashes (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx).	The specified **strListName** is ill-formatted.
0x80090007	The GUID should contain 32 digits with 4 dashes (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx).	The specified **strItemId** is empty or contains illegal values.
0x80004003	The value cannot be null.	The specified **strListName** is empty.
0x80131502	The value does not fall within the expected range.	The specified list item does not exist.
0x80070002	The system cannot find the **file** specified. (Exception from HRESULT: 0x80070002)	The specified list does not exist.

3.1.4.3 GetList

The **GetList** operation is used to obtain general information about a list, its schema (field names and their types), and the permissions granted to various users and groups.

```
<wsdl:operation name="GetList">
      <wsdl:input message="GetListSoapIn" />
      <wsdl:output message="GetListSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetListSoapIn** request message and the protocol server responds with a **GetListSoapOut** message.

3.1.4.3.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.3.1.1 GetListSoapIn

The **GetListSoapIn** message contains the **GetList** element that contains a single parameter: *strListName*. This parameter specifies either the list name, such as "Documents", or the GUID of the

list with or without the curly braces around it. The **GetListSoapIn** message is used to request general information about the list, its schema and access permissions.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/GetList
```

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

3.1.4.3.1.2 GetListSoapOut

The **GetListSoapOut** message contains detailed information about the list, including general information, detailed list schema, and access permissions.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/GetList
```

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

3.1.4.3.2 Elements

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

3.1.4.3.2.1 GetList

The **GetList** element specifies the list from which the client tries to get information:

```
<s:element name="GetList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="strListName" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

strListName: A string that contains either a list name, such as "Documents", or a GUID of the list, in the following format:

```
{318B9E8F-1EF4-4D49-9773-1BD2976772B6}
```

The string can be delimited with or without curly braces.

3.1.4.3.2.2 GetListResponse

The **GetListResponse** element specifies the result from the GetList operation:

```
<s:element name="GetListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetListResult"
                 type="s:unsignedInt" />
      <s:element minOccurs="1" maxOccurs="1" name="sListMetadata"
                 type="tns: sListMetadata" />
      <s:element minOccurs="1" maxOccurs="1" name="vProperties"
                 type="tns:ArrayOf_sProperty" />
```

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

GetListResult: An unsigned integer that specifies the status of the operation. On successful execution of the operation, the protocol server MUST set it to 0.

sListMetadata: An **_sListMetadata** complex type (section 3.1.4.3.3.1) that specifies metadata information about the list.

vProperties: An array of **_sProperty** elements, as specified in section 3.1.4.3.3.2.

If the list name or the GUID supplied by the protocol client is not well formed or does not correspond to any list that the target site contains, the protocol server MUST return a SOAP fault.

This protocol assumes that the list collection of any site is non-hierarchical, that is, lists do not have sublists. Therefore, list names are not composite names and MUST NOT contain slashes (/). One example of an ill-formed name is "Lists/Contacts".

3.1.4.3.3 Complex Types

3.1.4.3.3.1 _sListMetadata

The **_sListMetadata** complex type contains general information about the list, which has the following definition:

```
<s:complexType name="_sListMetadata">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Title" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Description" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="BaseType"
               type="tns:ListBaseType" />
    <s:element minOccurs="1" maxOccurs="1" name="BaseTemplate"
               type="tns:ListBaseTemplate" />
    <s:element minOccurs="1" maxOccurs="1" name="DefaultViewUrl" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModified" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModifiedForceRecrawl"
               type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="Author" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ValidSecurityInfo"
               type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="InheritedSecurity"
               type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="AllowAnonymousAccess"
               type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="AnonymousViewListItems"
               type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="ReadSecurity" type="s:int" />
    <s:element minOccurs="0" maxOccurs="1" name="Permissions" type="s:string" />
  </s:sequence>
</s:complexType>
```

Title: A string that specifies the title of the list.

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

BaseType: Specifies the name of the base type of the list. It MUST be of type **ListBaseType** as specified in section 2.2.5.1.

BaseTemplate: Specifies the name of the base template of the list. It MUST be of type **ListBaseTemplate** as specified in section 2.2.5.2.

DefaultViewUrl: A string that specifies the **server-relative URL** of the page containing the **default view** for the list.

LastModified: A dateTime element that specifies the time of last modification (for example, item added) in **Coordinated Universal Time (UTC)** format.

LastModifiedForceRecrawl: Not used. The protocol server MUST set it to **DateTime.MinValue**.

Author: A string that specifies the full name of the user who created the List.

ValidSecurityInfo: A Boolean value that is set to **false** if the requester has no right to view permissions or **true** if the user has the right to view permissions.

InheritedSecurity: A Boolean value that is set to **true** if the list inherits security from its parent site or **false** if its permissions are individually configured.

AllowAnonymousAccess: A Boolean value that is set to **true** if non-authenticated users can access the list or **false** if non-authenticated users cannot access the list.

AnonymousViewListItems: A Boolean value that is set to **true** if non-authenticated users can view list items or **false** if non-authenticated users cannot view list items.

ReadSecurity: An integer that specifies the read security setting. Possible values include the those described in the following table.

Value	Meaning
1	All users have read access to all items.
2	Users have read access only to the Items that they created.

Permissions: If the **current user** does not have read permissions, the field is empty (see also **ValidSecurityInfo**). If **InheritedSecurity** is true, **Permissions** is empty. Otherwise (when the list has individually configured permissions), the **Permissions** string contains well-formed XML described in section 3.1.4.3.3.4. If the list inherits permissions from its parent, this element MUST not be present.

3.1.4.3.3.2 ArrayOf_sProperty

The **ArrayOf_sProperty** complex type is a list of **_sProperty** elements that specify the schema of the list.

```
<s:complexType name="ArrayOf_sProperty">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name=" sProperty"
               type="tns:_sProperty" />
  </s:sequence>
</s:complexType>
```

_sProperty: Field name and type, as defined in section 3.1.4.3.3.3.

3.1.4.3.3.3 _sProperty

The **_sProperty** complex type defines the name and type of a list field.

```
<s:complexType name="_sProperty">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Name" type="s:string" />
```

```
    <s:element minOccurs="1" maxOccurs="1" name="Title" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Type" type="s:string" />
  </s:sequence>
</s:complexType>
```

Name: A string specifying the name of the field, which can be used to construct a query or to interpret **GetListItems** responses (see section 3.1.4.5.1.2).

Title: A string specifying the title shown in the site pages.

Type: A string specifying the type of the field, as specified in [MS-WSSTS] section 2.1.

3.1.4.3.3.4 ListPermissions

The **ListPermissions** complex type is an array of **Permission** elements, as per the following schema:

```
<s:complexType name="ListPermissions">
  <s:sequence>
    <s:element name="Permission" type="tns:ListPermission"  maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

Permission: See section 3.1.4.3.3.5.

3.1.4.3.3.5 ListPermission

The **ListPermission** complex type specifies information about the permissions granted to certain users or groups in the **site membership**. It has the following schema:

```
<s:complexType name="ListPermission">
   <s:attribute name="MemberID" type="s:int" use="required"/>
   <s:attribute name="Mask" type="tns:unsignedLong" use="required"/>
   <s:attribute name="MemberIsUser" type="tns:TrueFalseType" use="required"/>
   <s:attribute name="MemberGlobal" type="tns:TrueFalseType" use="required"/>
   <s:attribute name="UserLogin" type="s:string" />
   <s:attribute name="GroupName" type="s:string "/>
 </s:complexType>
```

ListPermission attributes have the following meanings:

MemberID: An integer that specifies the identifier of the user or group in the site membership.

Mask: An unsigned long integer that specifies a mask of permission **flags** as defined in [MS-WSSFO2] section 2.2.2.14.

MemberIsUser: A Boolean value that is set to **true** if **MemberID** specifies a user in the site membership, or **false** if **MemberID** specifies a group in the site membership. Note that the user in site membership can be an Active Directory Group, see [MS-ADTS]. Each member of the Active Directory Group is then treated as the same logical user and has the same permissions as far as this protocol is concerned.

MemberGlobal: A Boolean value that is set to **true** if **MemberID** specifies a group, or **false** otherwise. **MemberGlobal** is the inverse of **MemberIsUser**. This attribute is deprecated.

UserLogin: A string that specifies the login name of the user. **UserLogin** MUST be present when **MemberIsUser** is **true**.

GroupName: A string that contains the group name as defined in the site membership, for example, "Home Owners". **GroupName** MUST be present when **MemberIsUser** is **false**.

3.1.4.4 GetListCollection

The **GetListCollection** operation is used to get general information about all the lists in the context site.

```
<wsdl:operation name="GetListCollection">
    <wsdl:input message="GetListCollectionSoapIn" />
    <wsdl:output message="GetListCollectionSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetListCollectionSoapIn** request message, and the protocol server responds with a **GetListCollectionSoapOut** response message.

3.1.4.4.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.4.1.1 GetListCollectionSoapIn

The **GetListCollectionSoapIn** message is the request message to get information about all the lists in the context site.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetListCollection
```

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

3.1.4.4.1.2 GetListCollectionSoapOut

The **GetListCollectionSoapOut** message is the response message to get information about all the lists in the context site.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetListCollection
```

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

3.1.4.4.2 Elements

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

3.1.4.4.2.1 GetListCollection

The **GetListCollection** element is the request message used to get information about all the lists in the context site.

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

3.1.4.4.2.2 GetListCollectionResponse

The **GetListCollectionResponse** element specifies the result from **GetListCollection** operation:

```
<s:element name="GetListCollectionResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetListCollectionResult"
                 type="s:unsignedInt" />
      <s:element minOccurs="1" maxOccurs="1" name="vLists"
                 type="tns:ArrayOf sList" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetListCollectionResult: An unsigned integer that specifies the status of the operation. On successful completion of the operation, the protocol server MUST set it to 0.

vLists: An **ArrayOf_sList** complex type (section 3.1.4.4.3.2) that specifies an array of list information about all the lists in the context site.

If the protocol server encounters an error during the execution of this operation, a SOAP fault MUST be returned. There are no error(s) specific to this operation.

3.1.4.4.3 Complex Types

3.1.4.4.3.1 _sList

The **_sList** complex type contains metadata information about a list.

```
<s:complexType name="_sList">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="InternalName" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Title" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Description" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="BaseType"
               type=" tns:ListBaseType " />
    <s:element minOccurs="1" maxOccurs="1" name="BaseTemplate"
               type="tns:ListBaseTemplate" />
    <s:element minOccurs="1" maxOccurs="1" name="DefaultViewUrl" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModified" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="PermId" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="InheritedSecurity"
               type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="AllowAnonymousAccess"
               type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="AnonymousViewListItems"
               type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="ReadSecurity" type="s:int" />
  </s:sequence>
</s:complexType>
```

InternalName: A string that specifies the GUID of the list. This GUID MUST be enclosed in curly braces ({}).

Title: A string that specifies the title of the list.

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

BaseType: A **ListBaseType** (section 2.2.5.1) that specifies the name of the base type of the list. It MUST be of type **ListBaseType**.

BaseTemplate: A **ListBaseTemplate** (section 2.2.5.2) that specifies the name of the base template of the list. It MUST be of type **ListBaseTemplate**.

DefaultViewUrl: A string that specifies the server-relative URL of the page containing the default **view** for the list.

LastModified: A string that specifies the date and time when the list was last modified.

PermId: Not used. This element MUST NOT be included in the **_sList**.

InheritedSecurity: A Boolean value that specifies whether this list inherits security from its parent site. **true** indicates that this list inherits security from its **parent site**, and **false** indicates that permissions of the list are individually configured.

AllowAnonymousAccess: A Boolean value that is set to **true** if non-authenticated users can access the list; otherwise, **false**.

AnonymousViewListItems: A Boolean value that is set to **true** if non-authenticated users can view list items of the list; otherwise, **false**.

ReadSecurity: An integer that specifies what items users have read access to. Possible values include the those described in the following table.

Value	Meaning
1	All users have read access to all items.
2	Users have read access only to the items that they created.

3.1.4.4.3.2 ArrayOf_sList

The **ArrayOf_sList** complex type contains an array of list information.

```
<s:complexType name="ArrayOf_sList">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name=" sList"
               type="tns: sList" />
  </s:sequence>
</s:complexType>
```

_sList: A complex type that contains metadata information about a list, as defined in (section 3.1.4.4.3.1).

3.1.4.5 GetListItems

The **GetListItems** operation is used to retrieve details of all list items satisfying certain criteria.

The WSDL schema for the **GetListItems** operation is as follows:

```
<wsdl:operation name="GetListItems">
    <wsdl:input message="GetListItemsSoapIn" />
    <wsdl:output message="GetListItemsSoapOut" />
</wsdl:operation>
```

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

The protocol client specifies the following inputs in the **GetListItems** element of the **GetListItemsSoapIn** message:

- The list identifier.
- The query that defines the selection criteria and the ordering of list items in the result.
- The maximum number of list items to be included in the result.

The protocol server responds by sending a **GetListItemsSoapOut** message that consists of a single **GetListItemsResponse** element. This **GetListItemsResponse** element contains a single **GetListItemsResult** element. The value of this **GetListItemsResult** element MUST be a well-formed XML string in ADO XML Persistence format, as defined in [MS-PRSTFR], (see also section 7), with one row for each list item that matches the selection criteria.

Note 1: The protocol client limits the number of list items to be returned by the *uRowLimit* parameter of the **GetListItems** element.

To achieve pagination, that is, to be able to sequentially retrieve a large number of list Items by fixed-size chunks, the protocol client must include the Greater Than (**Gt**) predicate on some field (see section 3.1.4.5.3.1). The protocol client may optionally include that field in the **OrderBy** clause (see section 3.1.4.5.3.2), and modify the **Gt** criterion in each subsequent call using the maximum field value returned by the previous call. The **ID** field is a good candidate for pagination because it has a well-known minimum value of 1, hence the protocol client may specify the criterion **ID** > 0 to begin the pagination.

Note 2: In the response, row attribute names are not the same as the field names prescribed by the schema returned by **GetList**. To relate row attributes to field names, the protocol client has to look up the **Xdr** schema of the response, and then look up the schema returned by **GetList** as specified in GetList (section 3.1.4.3).

The following example shows how the protocol client can look up the field names using the **Xdr** schema of the response.

Assume that the protocol client performed the **GetListItems** operation and has obtained a response row with the **ows_ContentTypeId** attribute, as follows:

```
<z:row
    ows_ContentTypeId='0x010007CCE586A00F8F408754CA2BA1A39C02'>
<!-- other attributes … -->
</z:row>
```

The **Xdr** schema associates **ows_ContentTypeId** with the underlying **rs:name** "Content Type ID". This is actually the **Title**, rather than the name:

```
<xdr:AttributeType  name='ows_ContentTypeId'
                    rs:name='Content Type ID'
                    rs:number='1'>
   <xdr:datatype  dt:type='int'
      dt:maxLength='512'>
   </xdr:datatype>
</xdr:AttributeType>
```

To get the valid **Name** of the field, the protocol client has to look up the schema returned by **GetList**, seeking for **Title** "Content Type ID"

```
<_sProperty >
   <Name >ContentTypeId</Name>
   <Title >Content Type ID</Title>
   <Type >ContentTypeId</Type>
```

```
</_sProperty>
```

The **ContentTypeId** is the **Name** to be used in **Where** and **OrderBy** clauses, and to be used for answer interpretation.

3.1.4.5.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.5.1.1 GetListItemsSoapIn

The **GetListItemsSoapIn** message is the request message to retrieve details of all list items satisfying certain criteria.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetListItems
```

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

3.1.4.5.1.2 GetListItemsSoapOut

The **GetListItemsSoapOut** message is the response message to retrieve details of all list items satisfying certain criteria.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/GetListItems
```

The SOAP body contains a single **GetListItemsResponse** element.

3.1.4.5.2 Elements

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

3.1.4.5.2.1 GetListItems

The **GetListItems** element specifies the criteria which determine which list items are retrieved:

```
<s:element name="GetListItems">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="strListName" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="strQuery" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="strViewFields"
                 type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="uRowLimit"
                 type="s:unsignedInt" />
    </s:sequence>
  </s:complexType>
</s:element>
```

strListName: A string parameter that MUST contain the GUID of the list. This GUID MUST be enclosed in curly braces<2> ({}).

strQuery: A string that, unless empty, MUST contain well-formed Xml elements **Where**, as described in section 3.1.4.5.3.1, and **OrderBy**, as described in 3.1.4.5.3.2. This **strQuery**, if not empty, MUST contain at most one **Where** clause and at most one **OrderBy** clause, in that order. An empty **strQuery** indicates that all list items are to be returned in any order.

strViewFields: Not used and MUST be empty.

uRowLimit: An unsigned integer that, unless set to 0, indicates that the protocol server MUST NOT return any more list items in the response than **uRowLimit** indicates. A value of 0 for **uRowLimit** indicates that all items that satisfy the criteria are to be returned.

3.1.4.5.2.2 GetListItemsResponse

The **GetListItemsResponse** element contains a response to a **GetListItems** request.

```
<s:element name="GetListItemsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetListItemsResult"
                type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetListItemsResult: MUST be a well-formed XML string compliant with the ADO XML Persistence Format schema as specified in [MS-PRSTFR]. **GetListItemsResult** consists of XDR schema for rows (all fields with their returned and original names, data types, and maximum length) and row data, one row for each list item returned. The number of row elements returned is limited to the top **uRowLimit** items matching the criteria defined in **GetListItems** element *strQuery* parameter. Each row element contains as many attributes as the XDR schema defines, and their values are given in the syntax prescribed by the corresponding data type. For example, if the data type is **datetime,** then the format for the value is UTC format.

If the protocol server encounters an error during the execution of this operation, a SOAP fault MUST be returned.

3.1.4.5.3 Complex Types

3.1.4.5.3.1 Where Format

The **Where** clause specifies the criteria governing which list items are returned by the **GetListItems** method. The **Where** clause MUST be well-formed XML and MUST comply with the **XML schema definition (XSD)** specified in [MS-WSSCAML].

An example of a well-formed **Where** clause follows:

```
<Where >
    <And >
        <Contains >
            <FieldRef Name='Title'>
            </FieldRef>
            <Value Type='Text'>Red</Value>
        </Contains>
        <Gt >
            <FieldRef Name='ID'>
            </FieldRef>
            <Value Type='Counter'>0</Value>
        </Gt>
    </And>
```

```
</Where>
```

The ANSI-SQL language equivalent of these criteria would be as follows:

```
WHERE Title LIKE ''%Red%" AND ID > 0
```

The **Where** clause can contain an arbitrary number of conditions, joined into And/Or groups if more than one condition is defined. Each And/Or group MUST contain two elements – each being either condition or nested And/Or group.

Each condition consists of a **Predicate**, a **FieldRef** element, and a **Value** of the appropriate type, where:

- **Predicate** specifies the comparison, such as **Eq**, **Gt**, or **Le**, as specified in the complete XSD in [MS-WSSCAML].
- **FieldRef** element is a reference to one of the fields. The **FieldRef** name attribute MUST be a valid field name. Valid field names are provided in the **Name** attribute of the **_sProperty** element (section 3.1.4.3.3.3) returned from the **GetList** operation.
- The **Value** of the appropriate type is suitable to be compared using the **Predicate** with the field in question. Two predicates (**IsNull** and **IsNotNull**) do not require a **Value** for comparison.

3.1.4.5.3.2 OrderBy Format

The **OrderBy** clause MUST be well-formed XML as defined by the XSD in [MS-WSSCAML]. The **OrderBy** clause specifies the order of the list items in the response by identifying the names of certain fields.

An example of an **OrderBy** clause is as follows:

```
<OrderBy >
    <FieldRef  Name='ID'>
    </FieldRef>
    <FieldRef  Name='Title'
               Ascending='False'>
    </FieldRef>
</OrderBy>
```

The **OrderBy** clause contains a **FieldRef** element which MUST refer to a valid field name. Valid field names are provided in the **Name** attribute of the **_sProperty** element (section 3.1.4.3.3.3) returned from the **GetList** operation. The **Ascending** attribute is optional and indicates ascending order if value is **true**, or descending otherwise. By default, **Ascending** is **true**.

3.1.4.6 GetSite

The **GetSite** operation is used to retrieve information about the site collection of the context site. This information includes the following:

- Metadata about the site collection.
- All the subsites in the site collection.
- All the users of the site collection.

- All the groups of the site collection.

```
<wsdl:operation name="GetSite">
    <wsdl:input message="GetSiteSoapIn" />
    <wsdl:output message="GetSiteSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetSiteSoapIn** request message, and the protocol server responds with a **GetSiteSoapOut** response message.

3.1.4.6.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.6.1.1 GetSiteSoapIn

The **GetSiteSoapIn** message is the request message to get information about the site collection.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetSite
```

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

3.1.4.6.1.2 GetSiteSoapOut

The **GetSiteSoapOut** message is the response to get metadata information about the site collection.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetSite
```

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

3.1.4.6.2 Elements

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

3.1.4.6.2.1 GetSite

The **GetSite** element specifies the body of the SOAP message:

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

3.1.4.6.2.2 GetSiteResponse

The **GetSiteResponse** element specifies the result from **GetSite** operation:

```
<s:element name="GetSiteResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetSiteResult"
                 type="s:unsignedInt" />
      <s:element minOccurs="1" maxOccurs="1" name="sSiteMetadata"
```

```
                type="tns:_sSiteMetadata" />
          <s:element minOccurs="0" maxOccurs="1" name="vWebs"
                type="tns:ArrayOf_sWebWithTime" />
          <s:element minOccurs="0" maxOccurs="1" name="strUsers" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="strGroups" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="vGroups"
                type="tns:ArrayOfString" />
        </s:sequence>
      </s:complexType>
    </s:element>
```

GetSiteResult: An unsigned integer that specifies the status of the operation. On successful execution of the operation, the protocol server MUST set it to 0.

sSiteMetadata: A **_sSiteMetadata** complex type (section 3.1.4.6.3.1) that specifies the metadata information about the site collection.

vWebs: An **ArrayOf_sWebWithTime** complex types (section 2.2.4.4) that specifies an array of information about all the subsites in the site collection. **vWebs** MUST be returned only if the site collection is a small site collection and the current user is the administrator of the site collection.

strUsers: A string that protocol server MUST set to null.

strGroups: A string that contains an **XML node** whose schema is as specified in section 2.2.4.8. **strGroups** specifies information about all the groups in the site collection.

vGroups: An **ArrayOfString** (section 2.2.4.5) that specifies information about the users of all the groups for the site collection. Each string in the array is an XML node containing information about all the users of a group. The number of strings in the array MUST be equal to the number of groups in **strGroups**. The order of user information in **vGroups** MUST correspond with the order of groups in **strGroups**. Each XML node is as defined [MS-SITEDATS] section 2.2.4.24.

If the protocol server encounters an error during the execution of this operation, a SOAP fault MUST be returned. There are no error(s) specific to this operation.

3.1.4.6.2.3 Groups

The **Groups** element contains a list of group information.

```
<s:element name="Groups" type="tns:Groups"/>
```

Groups: Refer to **Groups**, as specified in section 2.2.4.8.

3.1.4.6.2.4 Users

The **Users** element contains a list of user information.

```
<s:element name="Users" type="tns:Users"/>
```

Users: As specified in [MS-SITEDATS] section 2.2.4.24.

3.1.4.6.3 Complex Types

3.1.4.6.3.1 _sSiteMetadata

The **_sSiteMetadata** complex type contains metadata information about a site collection.

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

```
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="LastModified" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModifiedForceRecrawl"
               type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="SmallSite" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="PortalUrl" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="UserProfileGUID" type="s:string" />


    <s:element minOccurs="1" maxOccurs="1" name="ValidSecurityInfo"
               type="s:boolean" />
  </s:sequence>
</s:complexType>
```

LastModified: A datetime that specifies the UTC date and time when the site collection was last modified.

LastModifiedForceRecrawl: A datetime that specifies the UTC date and time when the permissions of any structural element of the site collection were last modified.

SmallSite: A Boolean that specifies whether the site collection is small (has less than 1000 subsites). Set to **true** if the site collection has less than 1000 subsites; otherwise, **false**.

PortalUrl: A string that specifies the URL of the portal associated with the site collection.

UserProfileGUID: A string that specifies whether the site is a personal site. If the site is a **personal site** it MUST specify the GUID of the user who owns the site. Otherwise, it MUST be **null**.

ValidSecurityInfo: A Boolean that specifies whether the current user is the administrator of the site collection. Set to **true** if the current user is the administrator of the site collection; otherwise, **false**.

3.1.4.7 GetSiteAndWeb

The **GetSiteAndWeb** operation is used to get the URL of the site collection and the site to which the specified URL belongs.

```
<wsdl:operation name="GetSiteAndWeb">
    <wsdl:input message="GetSiteAndWebSoapIn" />
    <wsdl:output message="GetSiteAndWebSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetSiteAndWebSoapIn** request message, and the protocol server responds with a **GetSiteAndWebSoapOut** response message.

3.1.4.7.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.7.1.1 GetSiteAndWebSoapIn

The **GetSiteAndWebSoapIn** message is the request message to get the URL of the site collection and the site to which the specified URL belongs.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetSiteAndWeb
```

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

3.1.4.7.1.2 GetSiteAndWebSoapOut

The **GetSiteAndWebSoapOut** message is the response message to get the URL of the site collection and the site to which the specified URL belongs.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetSiteAndWeb
```

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

3.1.4.7.2 Elements

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

3.1.4.7.2.1 GetSiteAndWeb

The **GetSiteAndWeb** element specifies the site from which the client is retrieving information.

```
<s:element name="GetSiteAndWeb">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="strUrl" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

strUrl: A string that specifies the URL of any item on the site.

3.1.4.7.2.2 GetSiteAndWebResponse

The **GetSiteAndWebResponse** element specifies the result from **GetSiteAndWeb** operation.

```
<s:element name="GetSiteAndWebResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetSiteAndWebResult" type="s:unsignedInt"
/>
      <s:element minOccurs="1" maxOccurs="1" name="strSite" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="strWeb" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetSiteAndWebResult: An unsigned integer that contains the status of the operation. On successful completion of the operation the protocol server MUST set it to 0.

strSite: A string that specifies the absolute URL of the site collection that contains the item specified by the *strUrl* parameter.

strWeb: A string that specifies the absolute URL of the site that contains the item specified by the *strUrl* parameter.

If the protocol server encounters an error during the execution of this operation, a SOAP fault MUST be returned. If the input parameter *strUrl* is empty, the protocol server MUST return a SOAP fault with the error message, "Invalid URI: The URI is empty.".

3.1.4.8 GetURLSegments

The **GetURLSegments** operation is used to get the identifiers of a site, list, or list item depending on the specified URL. The behavior of the operation depends on the specified URL.

```
<wsdl:operation name="GetURLSegments">
    <wsdl:input message="GetURLSegmentsSoapIn" />
    <wsdl:output message="GetURLSegmentsSoapOut" />
</wsdl:operation>
```

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

- The protocol client specifies the absolute URL of any item on the site.
- The response message from the protocol server contains different GUIDs and IDs depending on the type of element addressed by the *strURL* parameter, as follows:
  - If the *strURL* parameter is the URL of the context site and if the context site has an **external security provider**, the protocol server MUST return the GUID of the site and the GUID of the external security provider. This GUID MUST be enclosed in curly braces ({}).
  - Otherwise, the *strURL* parameter is a URL of a list item within the context site, the protocol server MUST return:
  - The GUID of the list that contains the list item. This GUID MUST be enclosed in curly braces ({}).
  - The ID of the list item.
- Otherwise, the *strURL* parameter is a URL of a list within the request site, the protocol server MUST return the GUID of the list. This GUID MUST be enclosed in curly braces ({}).
- Otherwise the *strURL* parameter is a URL of a document within the context site, the protocol server MUST return the following:
  - The GUID of the list that contains the document list item. The GUID MUST be enclosed in curly braces ({}).
  - The ID of the document list item.

If the input URL matches any of the preceding scenarios, the protocol server MUST return **true** for the *GetURLSegmentsResult* parameter. Otherwise, the protocol server MUST return **false.**

3.1.4.8.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.8.1.1 GetURLSegmentsSoapIn

The **GetURLSegmentsSoapIn** message is the request message of the **GetURLSegments** operation.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetURLSegments
```

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

3.1.4.8.1.2 GetURLSegmentsSoapOut

The GetURLSegmentsSoapOut message is the response message of the **GetURLSegments** operation.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetURLSegments
```

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

3.1.4.8.2 Elements

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

3.1.4.8.2.1 GetURLSegments

The **GetURLSegments** element specifies the Url of which the client is requesting information.

```
<s:element name="GetURLSegments">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="strURL" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

strURL: A string that specifies an absolute URL for any item in the context site.

3.1.4.8.2.2 GetURLSegmentsResponse

The **GetURLSegmentsResponse** element specifies the result of **GetUrlSegments** operation:

```
<s:element name="GetURLSegmentsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetURLSegmentsResult"
                 type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="strWebID" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="strBucketID" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="strListID" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="strItemID" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetURLSegmentsResult: A Boolean that specifies whether the operation returned a result.

strWebID: A string that specifies the GUID of the site. This GUID MUST be enclosed in curly braces ({}). This value MUST not be null if the input URL matches the URL of the server Web and the Web has external security.

strBucketID: A string that specifies the GUID of the external security provider. This GUID MUST be enclosed in curly braces ({}). This value MUST not be null if **strWebId** is not **null**.

strListID: A string that specifies the GUID of the list. This GUID MUST be enclosed in curly braces ({}). This value MUST not be null if the input URL matches the URL of a **List** or a **ListItem**.

strItemID: A string that specifies the identifier of the list item. This value MUST not be null if the input URL matches the URL of a **ListItem**.

If the protocol server encounters an error during the execution of this operation, a SOAP fault MUST be returned. If the input parameter *strUrl* is empty, the protocol server MUST return a SOAP fault with the error message, "Invalid URI: The URI is empty.".

3.1.4.9 GetWeb

The **GetWeb** operation is used to get metadata information, subsite information, and lists and **role definitions** for the context site.

```
<wsdl:operation name="GetWeb">
    <wsdl:input message="GetWebSoapIn" />
    <wsdl:output message="GetWebSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetWebSoapIn** request message, and the protocol server responds with a **GetWebSoapOut** response message.

3.1.4.9.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.9.1.1 GetWebSoapIn

The **GetWebSoapIn** message is the request message to get information about a site.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetWeb
```

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

3.1.4.9.1.2 GetWebSoapOut

The GetWebSoapOut message is the response message to get information about a site.

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

```
http://schemas.microsoft.com/sharepoint/soap/GetWeb
```

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

3.1.4.9.2 Elements

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

3.1.4.9.2.1 GetWeb

The **GetWeb** element specifies the body of the SOAP message:

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

3.1.4.9.2.2 GetWebResponse

The definition of **GetWebResponse** specifies the result of **GetWeb** operation:

```
<s:element name="GetWebResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetWebResult" type="s:unsignedInt" />
      <s:element minOccurs="1" maxOccurs="1" name="sWebMetadata" type="tns:_sWebMetadata" />
      <s:element minOccurs="1" maxOccurs="1" name="vWebs" type="tns:ArrayOf_sWebWithTime" />
      <s:element minOccurs="1" maxOccurs="1" name="vLists" type="tns:ArrayOf sListWithTime"
/>
      <s:element minOccurs="0" maxOccurs="1" name="vFPUrls" type="tns:ArrayOf sFPUrl" />
      <s:element minOccurs="0" maxOccurs="1" name="strRoles" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="vRolesUsers" type="tns:ArrayOfString" />
      <s:element minOccurs="0" maxOccurs="1" name="vRolesGroups" type="tns:ArrayOfString" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetWebResult: An unsigned integer that specifies the status of the operation. On successful completion the operation, the protocol server MUST set it to 0.

sWebMetadata: A **_sWebMetadata** (section 3.1.4.9.3.2). that specifies metadata information about the context site.

vWebs: An **ArrayOf_sWebWithTime** (section 2.2.4.4) that specifies information about all the subsites in the context site.

vLists: An **ArrayOf_sListWithTime** (section 3.1.4.9.3.3) that specifies information about all the lists in the context site.

vFPUrls: An **ArrayOf_sFPUrl** (section 2.2.4.3) that the protocol server MUST set to **null**.

strRoles: A string contains a XML node specifying information of all the role definition in the site. The schema of the **strRoles** is as defined in section 3.1.4.9.2.4. If the context site has no external security provider, the protocol server MUST return information about all the role definitions in the context site otherwise it MUST be null.

vRolesUsers: An **ArrayOfString** (section 2.2.4.5) in which each string in the array is a XML node specifying information of all the users in a role definition. Number of XML nodes in the array MUST be equal to the number of role definitions in the context site. Each XML node element in the array MUST have a schema as defined in section 2.2.4.11. If the context site has no external security provider, the protocol server MUST return user information for all the roles definitions of the context site otherwise it MUST be null.

vRolesGroups: An **ArrayOfString** (section 2.2.4.5) in which each string in the array is a XML node specifying information about all the groups in a role definition. Number of XML nodes in the array MUST be equal to the number of role definitions in the context site. Each XML node element in the array MUST have a schema as defined in **Groups**, as specified section 2.2.4.8. If the context site has no external security provider, the protocol server MUST return group information for all the role definitions of the context site in **vRolesGroup** otherwise it MUST be null.

If the protocol server encounters an error during the execution of this operation, a SOAP fault MUST be returned. There are no error(s) specific to this operation.

3.1.4.9.2.3 Permissions

The **Permissions** element contains an array of information about all the role definitions of the context site.

```
<s:element name="Permissions">
  <s:complexType>
    <s:sequence>
      <s:element maxOccurs="unbounded" name="Permission">
        <s:complexType>
          <s:attribute name="MemberID" type="s:int" use="required" />
          <s:attribute name="Mask" type="s:int" use="required" />
          <s:attribute name="MemberIsUser" type="tns:TrueFalseType"   use="required" />
          <s:attribute name="MemberGlobal" type="tns:TrueFalseType"   use="required" />
          <s:attribute name="RoleName" type="s:string" use="required" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

MemberId: An integer that specifies the ID role definition.

Mask: An integer that specifies the 32-bit integer value of the permissions associated with the role definition. Permissions of role definition are computed as defined in [MS-WSSFO2] section 2.2.2.14.

MemberIsUser: A **TrueFalseType** that MUST be set to **false**.

MemberGlobal: A **TrueFalseType** that MUST be set to **false**.

RoleName: A string that specifies the name of the role definition.

3.1.4.9.2.4 Roles

The **Roles** element contains an array of **Role** elements that specify role definition information.

```
<s:element name="Roles">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded" name="Role">
        <s:complexType>
          <s:attribute name="ID" type="s:unsignedInt" use="required" />
          <s:attribute name="Name" type="s:string" use="required" />
          <s:attribute name="Description" type="s:string" use="required" />
          <s:attribute name="Order" type="s:unsignedInt" use="required" />
          <s:attribute name="Hidden" type="tns:TrueFalseType" use="required" />
          <s:attribute name="Type" type="s:string" use="required" />
          <s:attribute name="BasePermissions" type="s:string" use="required" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

ID: An unsigned integer that specifies the identifier of a role definition.

Name: A string that specifies the name for the role definition.

Description: A string that specifies the description for the role definition.

Order: An unsigned integer that specifies the order of the role definition relative to the other role definitions in the array, when displayed in the user interface.

Hidden: A **TrueFalseType** that specifies whether the role definition is displayed in the user interface.

Type: A string that specifies the type of role definition. Possible values include the those described in the following table.

Value	Meaning
None	Has no rights.
Guest	Has limited rights to view pages and specific page elements.
Reader	Has rights to view content, but cannot add content.
Contributor	Has Reader rights, plus rights to add, edit, and delete content in lists and document libraries.
WebDesigner	Has Contributor rights, plus rights to modify the structure of the Web site and to create new lists and document libraries.
Administrator	Has WebDesigner rights, plus the rights to manage roles. Members of the Adminstrator role always have access to, or can grant themselves access to, any item in the Web site.

BasePermissions: A string which specifies the base permissions granted to the role definition. This string contains a list of permission names, separated by commas. The permission names are defined in [MS-WSSFO2] section 2.2.2.14.

3.1.4.9.3 Complex Types

3.1.4.9.3.1 _sListWithTime

The **_sListWithTime** complex type represents information of a list.

```
<s:complexType name="_sListWithTime">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="InternalName" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModified" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="IsEmpty" type="s:boolean" />
  </s:sequence>
</s:complexType>
```

InternalName: A string that specifies the GUID of list. This GUID MUST be enclosed in curly braces ({}).

LastModified: A dateTime that specifies the UTC date and time when the list was last modified.

IsEmpty: A Boolean that specifies whether the list is empty. The list is empty if it has no list items.

3.1.4.9.3.2 _sWebMetadata

The **_sWebMetadata** complex type contains metadata information for a site.

```
<s:complexType name=" sWebMetadata">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="WebID" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Title" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Description" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Author" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Language" type="s:unsignedInt" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModified" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModifiedForceRecrawl"
               type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="NoIndex" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ValidSecurityInfo"
               type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="InheritedSecurity"
```

```
            type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="AllowAnonymousAccess"
            type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="AnonymousViewListItems"
            type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="Permissions" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ExternalSecurity"
            type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="CategoryId" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="CategoryName" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="CategoryIdPath" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="IsBucketWeb" type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="UsedInAutocat" type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="CategoryBucketID"
            type="s:string" />
  </s:sequence>
</s:complexType>
```

WebID: Specifies the GUID of the site. This GUID MUST be enclosed in curly braces.

Title: Specifies the title of the site.

Description: Specifies the description of the site.

Author: Specifies the author name of the site.

Language: Specifies the locale identifier of the site.

LastModified: Specifies the UTC date and time when the site collection containing the site was last modified.

LastModifiedForceRecrawl: If the site has an external security provider **LastModifiedForceRecrawl** specifies date and time when the site was last modified otherwise it is **DateTime.MinValue**. Format of **LastModifiedForceRecrawl** depends on the implementation of the external security provider.

NoIndex: Specifies whether the site can be crawled for indexing. Possible values are those described in the following table.

Value	Meaning
True	Do not crawl the site for indexing.
Enumerate	Crawl the site for indexing.

ValidSecurityInfo: If the site has an external security provider then **ValidSecurityInfo** specifies whether the security provider has permissions information for the site. If the site does not have an external security provider, then **ValidSecurityInfo** MUST be **true**.

InheritedSecurity: Specifies whether the site inherits security from the parent site, **true** if it inherits, **false** if they are individually configured.

AllowAnonymousAccess: Specifies whether anonymous users have access permissions for the site.

AnonymousViewListItems: Specifies whether anonymous users can view list items in the site.

Permissions: If the site inherits security from its parent site, then it specifies the absolute URL of its parent site. Otherwise, if it contains permission information of user roles for the site, its schema is defined in **Permissions**, section 3.1.4.9.2.3.

ExternalSecurity: Specifies whether an external security provider is provided for the site.

CategoryId: Specifies the GUID of the site. This GUID MUST be enclosed in curly braces ({}). The protocol client MUST ignore it. If the site does not have an external security provider, **CategoryId** MUST be omitted.

CategoryName: Specifies the name of the site. The protocol client MUST ignore it. If the site does not have an external security provider, **CategoryName** MUST be omitted.

CategoryIdPath: It is a colon delimited list of the GUIDs of the web application, all the parent sites and the site itself. It MUST begin with a colon. The protocol client MUST ignore it. If the site does not have an external security provider, **CategoryIdPath** MUST be omitted.

For example: `CategoryIdPath = :af377a2f-b65d-439b-85d3-d7d9c9694cfd:144960A2-F7F9-4E03-B4DE-5A4DB793E900:3CB4310E-706F-41A5-A85B-FAAFA01A7A8A:50D959CF-9804-47A3-AE1F-59959B71DFFB`.

Indicates that the web application with the GUID "af377a2f-b65d-439b-85d3-d7d9c9694cfd" contains a site with GUID "144960A2-F7F9-4E03-B4DE-5A4DB793E900". This site contains a subsite with GUID "3CB4310E-706F-41A5-A85B-FAAFA01A7A8A". This subsite contains a site with GUID "50D959CF-9804-47A3-AE1F-59959B71DFFB".

IsBucketWeb: Specifies whether the site is a bucket Web. The protocol client MUST ignore it. If the site does not have an external security provider, **IsBucketWeb** MUST be omitted.

UsedInAutocat: Specifies whether the site has categories. The protocol client MUST ignore it. If the site does not have an external security provider, **UsedInAutocat** MUST be omitted.

CategoryBucketID: Specifies the identifier of the **bucket Web** that contains the site. The protocol client MUST ignore it. If the site does not have an external security provider, **CategoryBucketID** MUST be omitted.

3.1.4.9.3.3 ArrayOf_sListWithTime

The **ArrayOf_sLisWithTime** complex time contains an array of list information.

```
<s:complexType name="ArrayOf_sListWithTime">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="_sListWithTime"
               type="tns:_sListWithTime" />
  </s:sequence>
</s:complexType>
```

_sListWithTime: Contains list information. It MUST be of type **_sListWithTime** (section 3.1.4.9.3.1).

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 Full Indexing

A site collection can be indexed in many ways. This example illustrates a part of one of the sequence of operations required to perform full indexing of a site collection.

In this scenario there is a site collection addressed by the URL http://fabrikam. This site collection contains many subsites. One of these subsites is addressed by the URL http://fabrikam/TestSubSite.

The protocol client calls the **GetSite** operation to get all the information about the site collection. The protocol client sends the following request **SOAP message** to call the **GetSite** operation.

```
<GetSite xmlns="http://schemas.microsoft.com/sharepoint/soap/" />
```

The protocol client sends the HTTP POST to the protocol server http://fabrikam.

The POST URL is `/_vti_bin/Sitedata.asmx`.

The header specifies the following SOAP action:

```
http://schemas.microsoft.com/sharepoint/soap/GetSite
```

The protocol server returns information about the site collection in a response SOAP message as defined in **GetSiteResponse** (section 3.1.4.6.2.2). The protocol client parses the response and collects the URLs of all the subsites in the site collection.

The protocol client calls the **GetWeb** operation for all the subsites of the site collection and collects information about all the lists, users, and groups of all the subsites.

The request SOAP message to call the **GetWeb** operation to get information about the site TestSubSite is as follows:

```
<GetWeb xmlns="http://schemas.microsoft.com/sharepoint/soap/" />
```

The protocol client sends the HTTP POST to the protocol server: `http://fabrikam/TestSubSite`

The POST URL is `/_vti_bin/Sitedata.asmx`.

The header specifies the following SOAP action:

```
http://schemas.microsoft.com/sharepoint/soap/GetWeb
```

The protocol server sends a response SOAP message as defined in **GetWebResponse** (section 3.1.4.9.2.2). The response contains all the information about of the subsite TestSubSite. The protocol client parses the response to collect information about all the lists, users, and groups of the subsite TestSubSite. The parsed response information can be used to call other operations such as **GetList** to collect additional information about all the lists of the subsite.

4.2 List Crawling

In this scenario, there is site "fabrikam" on the protocol server. The site "fabrikam" contains two lists called "Documents" and "Contacts".

The protocol client calls the **GetList** operation to get detailed information about each list. The request SOAP message to call the **GetList** operation for list with the GUID is shown as follows.

```
<GetList xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <strListName>{1CDC1A48-695E-4D10-BF84-87946607A456}</strListName>
</GetList>
```

The protocol client sends the HTTP POST to protocol server: `http://fabrikam`.

The POST URL is `/_vti_bin/Sitedata.asmx`.

The header specifies the following SOAP action:

```
http://schemas.microsoft.com/sharepoint/soap/GetList
```

The protocol server returns detailed information of the list as a response SOAP message as specified in **GetListResponse** (section 3.1.4.5.2.2). The protocol client parses the response and updates the index.

4.3 GetListItems

In this scenario, there is a site addressed by the URL http://fabrikam on the protocol server. The site fabrikam contains a document library identified by GUID "8AC68D3D-8A09-4403-8860-D0E494BBE894". The document library contains multiple documents with a .pdf file extension. All documents in a document library are also list items. The **DocIcon** attribute contains the file extension of the document. In this case the **DocIcon** attribute value is "pdf".

The protocol client calls the **GetListItems** operation to get all the information about at most 10 documents (list items) whose **DocIcon** attribute is "pdf". The protocol client sends the following SOAP message to the protocol server to call **GetListItems**:

```
<GetListItems xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <strListName>{8AC68D3D-8A09-4403-8860-D0E494BBE894}</strListName>
  <strQuery>&lt;Where&gt;&lt;Eq&gt;&lt;FieldRef Name="DocIcon"/&gt;&lt;Value
Type="Text"&gt;pdf&lt;/Value&gt;&lt;/Eq&gt;&lt;/Where&gt;</strQuery>
  <strViewFields></strViewFields>
  <uRowLimit>10</uRowLimit>
</GetListItems>
```

The protocol client sends the HTTP POST to the protocol server http://fabrikam.

The POST URL is `/_vti_bin/Sitedata.asmx`.

The header specifies the following SOAP action:

```
http://schemas.microsoft.com/sharepoint/soap/GetListItems
```

The protocol server returns the information about the matching pdf documents in a response SOAP message as defined in section **GetListItemsResponse** (section 3.1.4.5.2.2). The protocol client parses the response and collects information about all the documents in the response.

4.4 GetURLSegments

In this scenario, there is a site addressed by the URL http://fabrikam on the protocol server. This site contains a list with name "Documents" that is identified by the following GUID:

```
B59D96DA-59B5-48E9-842B-5F89EE514232
```

The Documents list contains a list item named "Customers" with the identifier 1. The **absolute URL** of the list item is http://fabrikam/Lists/Documents/DispForm.aspx?ID=1.

The protocol client calls the **GetURLSegments** operation to obtain the following information:

The GUID of the list which contains the list item Customers.

The identifier of the list item Customers.

The protocol client sends the following SOAP message to the protocol server to call **GetURLSegments**.

```
<GetURLSegments
xmlns="http://schemas.microsoft.com/sharepoint/soap/"><strURL>http://fabrikam/Lists/Documents
/DispForm.aspx?ID=1</strURL>
</GetURLSegments>
```

The protocol client sends the HTTP POST to protocol server http://fabrikam.

The POST URL is `/_vti_bin/Sitedata.asmx`.

The header specifies the following SOAP action:

```
http://schemas.microsoft.com/sharepoint/soap/GetURLSegments
```

The protocol server sends the following SOAP message response:

```
<GetURLSegmentsResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
   <GetURLSegmentsResult>true</GetURLSegmentsResult>
   <strListID>{B59D96DA-59B5-48E9-842B-5F89EE514232}</strListID>
   <strItemID>1</strItemID>
</GetURLSegmentsResponse>
```

The response message contains the following:

- The GUID of the list that contains the list item Customers.
- The identifier of the list item Customers.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full **WSDL** is provided:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
       targetNamespace="http://schemas.microsoft.com/sharepoint/soap/">
      <s:element name="GetSiteAndWeb">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="strUrl"
                       type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetSiteAndWebResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1"
               name="GetSiteAndWebResult" type="s:unsignedInt" />
            <s:element minOccurs="1" maxOccurs="1" name="strSite"
               type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="strWeb"
               type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetSite">
        <s:complexType />
      </s:element>
      <s:element name="GetSiteResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetSiteResult"
               type="s:unsignedInt" />
            <s:element minOccurs="1" maxOccurs="1" name="sSiteMetadata"
               type="tns:_sSiteMetadata" />
            <s:element minOccurs="0" maxOccurs="1" name="vWebs"
               type="tns:ArrayOf sWebWithTime" />
            <s:element minOccurs="0" maxOccurs="1" name="strUsers"
               type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="strGroups"
               type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="vGroups"
               type="tns:ArrayOfString" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="Groups" type="tns:Groups"/>
      <s:element name="Users" type="tns:Users"/>
      <s:complexType name=" sSiteMetadata">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="LastModified"
               type="s:dateTime" />
          <s:element minOccurs="1" maxOccurs="1"
               name="LastModifiedForceRecrawl" type="s:dateTime" />
          <s:element minOccurs="1" maxOccurs="1" name="SmallSite"
               type="s:boolean" />
          <s:element minOccurs="1" maxOccurs="1" name="PortalUrl"
               type="s:string" />
```

```
      <s:element minOccurs="1" maxOccurs="1" name="UserProfileGUID"
           type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="ValidSecurityInfo"
           type="s:boolean" />
    </s:sequence>
  </s:complexType>
  <s:complexType name="ArrayOf_sWebWithTime">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded"
           name="_sWebWithTime" type="tns:_sWebWithTime" />
    </s:sequence>
  </s:complexType>
  <s:complexType name="_sWebWithTime">
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="Url"
           type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="LastModified"
           type="s:dateTime" />
    </s:sequence>
  </s:complexType>
  <s:complexType name="ArrayOfString">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded" name="string"
           nillable="true" type="s:string" />
    </s:sequence>
  </s:complexType>
  <s:element name="GetWeb">
    <s:complexType />
  </s:element>
  <s:element name="GetWebResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="GetWebResult"
           type="s:unsignedInt" />
        <s:element minOccurs="1" maxOccurs="1" name="sWebMetadata"
           type="tns: sWebMetadata" />
        <s:element minOccurs="1" maxOccurs="1" name="vWebs"
           type="tns:ArrayOf_sWebWithTime" />
        <s:element minOccurs="1" maxOccurs="1" name="vLists"
           type="tns:ArrayOf sListWithTime" />
        <s:element minOccurs="0" maxOccurs="1" name="vFPUrls"
           type="tns:ArrayOf sFPUrl" />
        <s:element minOccurs="0" maxOccurs="1" name="strRoles"
           type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="vRolesUsers"
           type="tns:ArrayOfString" />
        <s:element minOccurs="0" maxOccurs="1" name="vRolesGroups"
           type="tns:ArrayOfString" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="Permissions">
    <s:complexType>
      <s:sequence>
        <s:element maxOccurs="unbounded" name="Permission">
          <s:complexType>
            <s:attribute name="MemberID" type="s:int" use="required" />
            <s:attribute name="Mask" type="s:int" use="required" />
            <s:attribute name="MemberIsUser" type="tns:TrueFalseType"
               use="required" />
            <s:attribute name="MemberGlobal" type="tns:TrueFalseType"
               use="required" />
            <s:attribute name="RoleName" type="s:string" use="required" />
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="Roles">
```

```
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded" name="Role">
        <s:complexType>
          <s:attribute name="ID" type="s:unsignedInt" use="required" />
          <s:attribute name="Name" type="s:string" use="required" />
          <s:attribute name="Description" type="s:string" use="required" />
          <s:attribute name="Order" type="s:unsignedInt" use="required" />
          <s:attribute name="Hidden" type="tns:TrueFalseType"
             use="required" />
          <s:attribute name="Type" type="s:string" use="required" />
          <s:attribute name="BasePermissions" type="s:string"
             use="required" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name=" sWebMetadata">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="WebID"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Title"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Description"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Author"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Language"
         type="s:unsignedInt" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModified"
         type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1"
         name="LastModifiedForceRecrawl" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="NoIndex"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ValidSecurityInfo"
         type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="InheritedSecurity"
         type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1"
         name="AllowAnonymousAccess" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1"
         name="AnonymousViewListItems" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="Permissions"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ExternalSecurity"
         type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="CategoryId"
         type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="CategoryName"
         type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="CategoryIdPath"
         type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="IsBucketWeb"
         type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="UsedInAutocat"
         type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="CategoryBucketID"
         type="s:string" />
  </s:sequence>
</s:complexType>
<s:complexType name="ArrayOf sListWithTime">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded"
         name="_sListWithTime" type="tns:_sListWithTime" />
  </s:sequence>
</s:complexType>
<s:complexType name="_sListWithTime">
```

```
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="InternalName"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModified"
         type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="IsEmpty"
         type="s:boolean" />
  </s:sequence>
</s:complexType>
<s:complexType name="ArrayOf_sFPUrl">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="_sFPUrl"
         type="tns:_sFPUrl" />
  </s:sequence>
</s:complexType>
<s:complexType name="_sFPUrl">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Url"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModified"
         type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="IsFolder"
         type="s:boolean" />
  </s:sequence>
</s:complexType>
<s:element name="GetList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="strListName"
         type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetListResult"
         type="s:unsignedInt" />
      <s:element minOccurs="1" maxOccurs="1" name="sListMetadata"
         type="tns: sListMetadata" />
      <s:element minOccurs="1" maxOccurs="1" name="vProperties"
         type="tns:ArrayOf sProperty" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name=" sListMetadata">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Title"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Description"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="BaseType"
         type="tns:ListBaseType" />
    <s:element minOccurs="1" maxOccurs="1" name="BaseTemplate"
         type="tns:ListBaseTemplate" />
    <s:element minOccurs="1" maxOccurs="1" name="DefaultViewUrl"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModified"
         type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1"
         name="LastModifiedForceRecrawl" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="Author"
         type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ValidSecurityInfo"
         type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="InheritedSecurity"
         type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1"
         name="AllowAnonymousAccess" type="s:boolean" />
```

```
      <s:element minOccurs="1" maxOccurs="1"
           name="AnonymousViewListItems" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="ReadSecurity"
           type="s:int" />
      <s:element minOccurs="0" maxOccurs="1" name="Permissions"
           type="s:string" />
    </s:sequence>
  </s:complexType>
  <s:simpleType name="ListBaseType">
    <s:restriction base="s:string">
      <s:enumeration value="UnspecifiedBaseType"/>
      <s:enumeration value="GenericList"/>
      <s:enumeration value="DocumentLibrary"/>
      <s:enumeration value="Unused"/>
      <s:enumeration value="DiscussionBoard"/>
      <s:enumeration value="Survey"/>
      <s:enumeration value="Issue"/>
    </s:restriction>
  </s:simpleType>
  <s:simpleType name="ListBaseTemplate">
    <s:restriction base="s:string">
      <s:enumeration value="InvalidType"/>
      <s:enumeration value="GenericList"/>
      <s:enumeration value="DocumentLibrary"/>
      <s:enumeration value="Survey"/>
      <s:enumeration value="Links"/>
      <s:enumeration value="Announcements"/>
      <s:enumeration value="Contacts"/>
      <s:enumeration value="Events"/>
      <s:enumeration value="Tasks"/>
      <s:enumeration value="DiscussionBoard"/>
      <s:enumeration value="PictureLibrary"/>
      <s:enumeration value="DataSources"/>
      <s:enumeration value="WebTemplateCatalog"/>
      <s:enumeration value="WebPartCatalog"/>
      <s:enumeration value="ListTemplateCatalog"/>
      <s:enumeration value="XMLForm"/>
      <s:enumeration value="CustomGrid"/>
      <s:enumeration value="IssueTracking"/>
    </s:restriction>
  </s:simpleType>
  <s:complexType name="ListPermissions">
    <s:sequence>
      <s:element name="Permission" type="tns:ListPermission"
           maxOccurs="unbounded"/>
    </s:sequence>
  </s:complexType>
  <s:complexType name="ListPermission">
    <s:attribute name="MemberID" type="s:int" use="required"/>
    <s:attribute name="Mask" type="tns:unsignedLong" use="required"/>
    <s:attribute name="MemberIsUser" type="tns:TrueFalseType" use="required"/>
    <s:attribute name="MemberGlobal" type="tns:TrueFalseType" use="required"/>
    <s:attribute name="UserLogin" type="s:string" />
    <s:attribute name="GroupName" type="s:string "/>
  </s:complexType>
  <s:complexType name="ArrayOf_sProperty">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded" name=" sProperty"
           type="tns: sProperty" />
    </s:sequence>
  </s:complexType>
  <s:complexType name="_sProperty">
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="Name"
           type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="Title"
           type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="Type"
           type="s:string" />
```

```
  </s:sequence>
</s:complexType>
<s:element name="GetListItems">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="strListName"
         type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="strQuery"
         type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="strViewFields"
         type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="uRowLimit"
        type="s:unsignedInt" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetListItemsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1"
         name="GetListItemsResult" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="EnumerateFolder">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="strFolderUrl"
         type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="EnumerateFolderResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1"
         name="EnumerateFolderResult" type="s:unsignedInt" />
      <s:element minOccurs="0" maxOccurs="1" name="vUrls"
         type="tns:ArrayOf_sFPUrl" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetAttachments">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="strListName"
         type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="strItemId"
         type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetAttachmentsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1"
         name="GetAttachmentsResult" type="s:unsignedInt" />
      <s:element minOccurs="0" maxOccurs="1" name="vAttachments"
         type="tns:ArrayOfString" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetURLSegments">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="strURL"
         type="s:string" />
    </s:sequence>
  </s:complexType>
```

```
    </s:element>
    <s:element name="GetURLSegmentsResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1"
             name="GetURLSegmentsResult" type="s:boolean" />
          <s:element minOccurs="0" maxOccurs="1" name="strWebID"
             type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="strBucketID"
            type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="strListID"
             type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="strItemID"
             type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetListCollection">
      <s:complexType />
    </s:element>
    <s:element name="GetListCollectionResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1"
            name="GetListCollectionResult" type="s:unsignedInt" />
          <s:element minOccurs="1" maxOccurs="1" name="vLists"
             type="tns:ArrayOf_sList" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="ArrayOf sList">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="_sList"
             type="tns: sList" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="_sList">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="InternalName"
             type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Title"
             type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Description"
             type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="BaseType"
             type="tns:ListBaseType" />
        <s:element minOccurs="1" maxOccurs="1" name="BaseTemplate"
             type="tns:ListBaseTemplate" />
        <s:element minOccurs="1" maxOccurs="1" name="DefaultViewUrl"
             type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="LastModified"
             type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="PermId"
             type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="InheritedSecurity"
             type="s:boolean" />
        <s:element minOccurs="1" maxOccurs="1"
             name="AllowAnonymousAccess" type="s:boolean" />
        <s:element minOccurs="1" maxOccurs="1"
             name="AnonymousViewListItems" type="s:boolean" />
        <s:element minOccurs="1" maxOccurs="1" name="ReadSecurity"
             type="s:int" />
      </s:sequence>
    </s:complexType>
  </s:schema>
</wsdl:types>
<wsdl:message name="GetSiteAndWebSoapIn">
  <wsdl:part name="parameters" element="tns:GetSiteAndWeb" />
</wsdl:message>
```

```
<wsdl:message name="GetSiteAndWebSoapOut">
  <wsdl:part name="parameters" element="tns:GetSiteAndWebResponse" />
</wsdl:message>
<wsdl:message name="GetSiteSoapIn">
  <wsdl:part name="parameters" element="tns:GetSite" />
</wsdl:message>
<wsdl:message name="GetSiteSoapOut">
  <wsdl:part name="parameters" element="tns:GetSiteResponse" />
</wsdl:message>
<wsdl:message name="GetWebSoapIn">
  <wsdl:part name="parameters" element="tns:GetWeb" />
</wsdl:message>
<wsdl:message name="GetWebSoapOut">
  <wsdl:part name="parameters" element="tns:GetWebResponse" />
</wsdl:message>
<wsdl:message name="GetListSoapIn">
  <wsdl:part name="parameters" element="tns:GetList" />
</wsdl:message>
<wsdl:message name="GetListSoapOut">
  <wsdl:part name="parameters" element="tns:GetListResponse" />
</wsdl:message>
<wsdl:message name="GetListItemsSoapIn">
  <wsdl:part name="parameters" element="tns:GetListItems" />
</wsdl:message>
<wsdl:message name="GetListItemsSoapOut">
  <wsdl:part name="parameters" element="tns:GetListItemsResponse" />
</wsdl:message>
<wsdl:message name="EnumerateFolderSoapIn">
  <wsdl:part name="parameters" element="tns:EnumerateFolder" />
</wsdl:message>
<wsdl:message name="EnumerateFolderSoapOut">
  <wsdl:part name="parameters" element="tns:EnumerateFolderResponse" />
</wsdl:message>
<wsdl:message name="GetAttachmentsSoapIn">
  <wsdl:part name="parameters" element="tns:GetAttachments" />
</wsdl:message>
<wsdl:message name="GetAttachmentsSoapOut">
  <wsdl:part name="parameters" element="tns:GetAttachmentsResponse" />
</wsdl:message>
<wsdl:message name="GetURLSegmentsSoapIn">
  <wsdl:part name="parameters" element="tns:GetURLSegments" />
</wsdl:message>
<wsdl:message name="GetURLSegmentsSoapOut">
  <wsdl:part name="parameters" element="tns:GetURLSegmentsResponse" />
</wsdl:message>
<wsdl:message name="GetListCollectionSoapIn">
  <wsdl:part name="parameters" element="tns:GetListCollection" />
</wsdl:message>
<wsdl:message name="GetListCollectionSoapOut">
  <wsdl:part name="parameters" element="tns:GetListCollectionResponse" />
</wsdl:message>
<wsdl:portType name="SiteDataSoap">
  <wsdl:operation name="GetSiteAndWeb">
    <wsdl:input message="tns:GetSiteAndWebSoapIn" />
    <wsdl:output message="tns:GetSiteAndWebSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetSite">
    <wsdl:input message="tns:GetSiteSoapIn" />
    <wsdl:output message="tns:GetSiteSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetWeb">
    <wsdl:input message="tns:GetWebSoapIn" />
    <wsdl:output message="tns:GetWebSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetList">
    <wsdl:input message="tns:GetListSoapIn" />
    <wsdl:output message="tns:GetListSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetListItems">
```

```
      <wsdl:input message="tns:GetListItemsSoapIn" />
      <wsdl:output message="tns:GetListItemsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="EnumerateFolder">
      <wsdl:input message="tns:EnumerateFolderSoapIn" />
      <wsdl:output message="tns:EnumerateFolderSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetAttachments">
      <wsdl:input message="tns:GetAttachmentsSoapIn" />
      <wsdl:output message="tns:GetAttachmentsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetURLSegments">
      <wsdl:input message="tns:GetURLSegmentsSoapIn" />
      <wsdl:output message="tns:GetURLSegmentsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetListCollection">
      <wsdl:input message="tns:GetListCollectionSoapIn" />
      <wsdl:output message="tns:GetListCollectionSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="SiteDataSoap" type="tns:SiteDataSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetSiteAndWeb">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSiteAndWeb"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSite">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSite"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetWeb">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetWeb"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetList">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetList"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetListItems">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetListItems"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
```

```
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="EnumerateFolder">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EnumerateFolder" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAttachments">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetAttachments" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetURLSegments">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetURLSegments" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetListCollection">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetListCollection" 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="SiteDataSoap12" type="tns:SiteDataSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetSiteAndWeb">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSiteAndWeb" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSite">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSite"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetWeb">
```

```
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetWeb"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetList">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetList"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetListItems">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetListItems" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="EnumerateFolder">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EnumerateFolder" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAttachments">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetAttachments" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetURLSegments">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetURLSegments" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetListCollection">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetListCollection" 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: Permissions Required for the Protocol Client

For the Site Data Protocol, Microsoft SharePoint servers use the port with a standard address location of http://root/_vti_bin/SiteData.asmx where **root** denotes a root URL of a site (or some subsite thereof).

The network account used by the client of Site Data Protocol has to belong to the group of Site Collection Administrators to be able to fully retrieve information about site structure, its users and groups, as well as permissions granted to those entities.

Those permissions can be granted to the client by the Site Collection Administrator using Central Administration > Application Management > Policy for Web Applications menu of the SharePoint Central Administration tool. By default, those permissions are granted only to **NT AUTHORITY\LOCAL SERVICE** and **NT AUTHORITY\NETWORK SERVICE** accounts, which even administrators cannot impersonate.

8 Appendix C: 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.

- Windows SharePoint Services 2.0

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 3.1.4.2.2.1: Windows SharePoint Services 2.0 does not treat a GUID without curly braces as an error, but instead accepts it as a valid GUID.

<2> Section 3.1.4.5.2.1: Windows SharePoint Services 2.0 does not treat a GUID without curly braces as an error, but instead accepts it as a valid GUID.

9 Change Tracking

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

10 Index

P

R

S

T

U

V

W