[MS-SITEDATS]:
Site Data Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

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

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
10/06/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
07/13/2009	1.03	Major	Revised and edited the technical content
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Minor	Clarified the meaning of the technical content.
07/23/2010	2.05	Minor	Clarified the meaning of the technical content.
09/27/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.06	Major	Significantly changed the technical content.
12/17/2010	2.06	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.06	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	3.0	Major	Significantly changed the technical content.
01/20/2012	4.0	Major	Significantly changed the technical content.
04/11/2012	4.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	4.0	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	4.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	4.1	Minor	Clarified the meaning of the technical content.
02/11/2013	4.2	Minor	Clarified the meaning of the technical content.

Table of Contents

1   Introduction

The Site Data Web Service Protocol consists of a set of server extensions used to augment a basic Hypertext Transfer Protocol (HTTP) server so that it supports full and incremental indexing. Indexing, in this context, is defined as the process of exploring Web site content and building an index to be used for search, systematic cataloging, content auditing, or similar purposes. This protocol is intended to be used by an indexing service application, not by a user directly through a Web browser.

This protocol provides such an indexing service with a set of operations for systematic, full traversal and incremental change tracking of site content. This protocol assumes that:

▪ The site content conforms to a well-defined hierarchical structure (specified in this document) where sites consist of lists, lists consist of items and items have standard properties.

▪ The protocol server keeps track of all changes to Web site content. The indexing service can periodically query for all changes that have occurred since the index was last updated, ensuring that the indexing service need not start anew every time.

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

1.1   Glossary

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

Coordinated Universal Time (UTC)
GUID
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
language code identifier (LCID)
principal
security identifier (SID)

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

absolute URL
alert
attachment
base type
bucket web
content database
content type
context site
current user
default list view
display form
document
document library
external security provider
field
file
folder

group
indexing service
list
list identifier
list item
list item identifier
login name
permission
personal site
role assignment
role definition
server-relative URL
Simple Object Access Protocol (SOAP)
site
site collection
site collection administrator
site membership
site subscription
site-relative URL
SOAP action
SOAP body
SOAP fault
SOAP message
store-relative URL
subsite
Uniform Resource Locator (URL)
user identifier
view
web application
Web Services Description Language (WSDL)
website
Welcome page
XML namespace
XML node
XML schema
XML schema definition (XSD)

The following terms are specific to this document:

change tracking space: A collection of one or more tokens that contains data about the granularity of and context in which each incremental change occurred for a specific content database or site collection.

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

1.2 References

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

1.2.1 Normative References

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

[ISO-8601] International Organization for Standardization, "Data Elements and Interchange Formats - Information Interchange - Representation of Dates and Times", ISO/IEC 8601:2004, December 2004, http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874&ICS1=1&ICS2=140&ICS3=30

Note There is a charge to download the specification.

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

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

[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

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, http://www.ietf.org/rfc/rfc2396.txt

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

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, http://www.rfc-editor.org/rfc/rfc5234.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.S., Beech, D., Maloney, M., Eds., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

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

1.2.2   Informative References

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

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

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

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

1.3   Protocol Overview (Synopsis)

This protocol supports **site (2)** indexing by external **indexing services**. Indexing is the process of building an external index of the **Web site (2)**, facilitating search, auditing, or cataloging of the site content. The purpose, techniques, and internal data structures used by an indexing service are beyond the scope of this protocol; however the protocol does make certain assumptions about the structure of the site content and the behavior of the protocol server, such as the permissions required for the Site Data client.

1.3.1   Site Structure
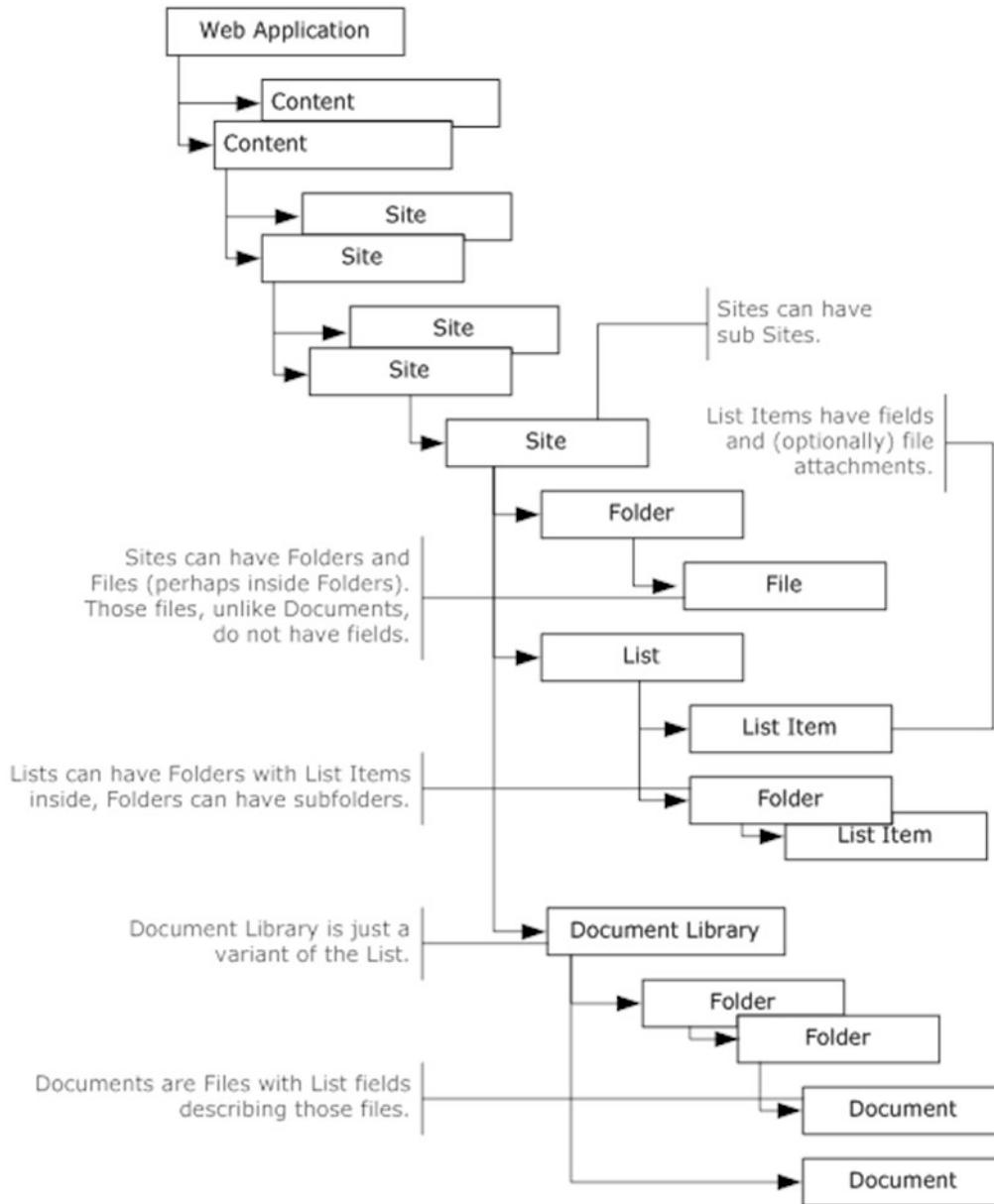
This protocol supports indexing services or similar client applications that follow the site (2) indexing process described in the diagram in section 1.3.2 to traverse a site (2) where the content conforms to the site (2) structure described in the following diagram.

The site structure described in the following diagram is specified in detail in section 3.1.1. The indexing process described in the diagram in section 1.3.2 is specified in detail in section 3.1.4.

This protocol requires the site to present content in a hierarchical structure as follows.

```
Web application
   content database
      site collection
         site
            subsite
               list
                  list item elements
```

This is described in the following diagram. It requires the protocol server to keep track of details about common types of content such as pages, **lists (1)**, **list items**, **documents**, and **document libraries**. The protocol server also tracks changes to content elements.
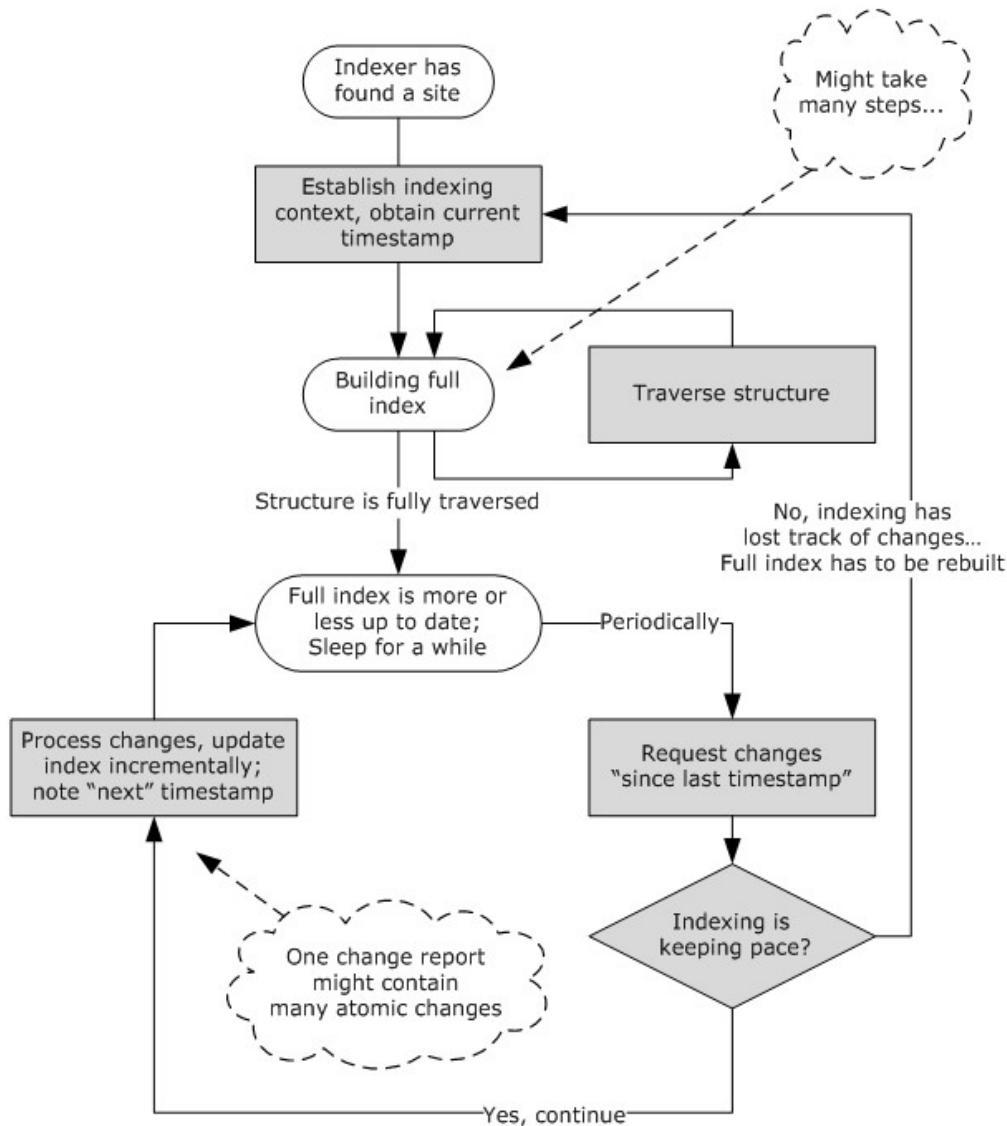
Figure 1: Hierarchical structure of a site

The indexing service or any other client uses the Site Data Web Service Protocol methods to traverse the site structure, explore **fields (2)** of list items, or retrieve the document library items, or list item **attachments**.

1.3.2   Site Indexing Process

This protocol is designed to support a client application (an indexing service) that conducts a full or incremental scan of the site (2) content following the recommended site (2) indexing process described in the following diagram. This process is described in detail in section 3.1.4.

Figure 2: Site indexing process

The site (2) indexing process assumes that the indexing service is configured with the **URL** of the site (2) to index, and that the site (2) supports this protocol.

First, the indexing service establishes the indexing context. This requires identifying the **content database** used by the site (2) to track changes, and obtaining a token (referred to as the StartToken) to be referred to in the incremental indexing stage. Details of context establishment are shown in the first figure in section 3.1.4.

Note that the timestamp-like tokens (formally called ChangeIds) used in this protocol create serialized entries in the server's change log and are opaque for the client applications. These tokens cannot be interpreted by the protocol client as moments in time.

After establishing context, the indexing service proceeds with a full traversal of the site (2). Traversal requires identifying all **subsites**, for each subsite identifying its lists and document libraries, and then scanning all list items and documents to peruse their content. The operations for this process are described in detail in section 3.1.4.

Note that a site (2) can store the documents in a variety of proprietary formats, as well as in many international languages. This protocol does not help the indexing service understand or interpret those formats and languages; it just allows the indexing service to retrieve them.

When the indexing process is done traversing the site (2) content, it reaches the "Full index more or less up to date; Sleep for a while" state indicated in Figure 2. The reason for the "more or less" qualifier is that the site (2) content might undergo changes before the indexing service completes full traversal of the site (2) content. In this case the full index can be already out-of-date with respect to the content, yet the basis for incremental change tracking has been built.
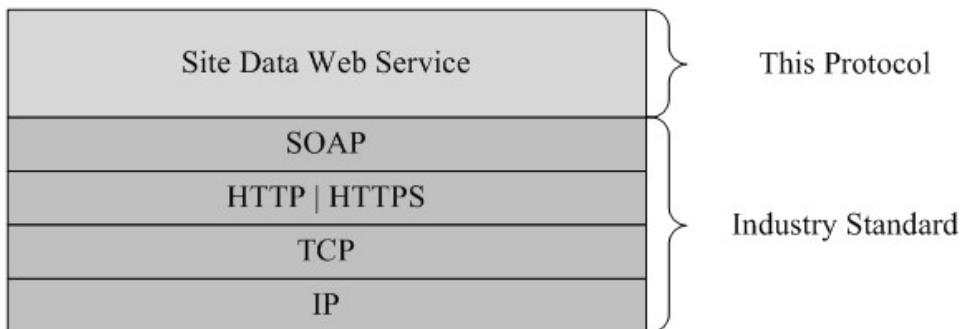
The indexing service then switches to incremental change requests. It uses a token to query the protocol server for all changes since full indexing started. The protocol server returns a change report, always starting with the oldest changes since the timestamp indicated by the request token. To accommodate large numbers of changes, the report always contains a "next change" token for indicating that all changes, from the timestamp of the requested token to the timestamp of the returned token, are included in the report, though there might be additional changes. During the next incremental index process, the indexing agent can use the "next change" token as the "since" token in the request.

While this protocol does not prescribe the frequency of change requests, change requests need to be done often enough to keep pace with the frequency of site (2) content changes. Because there is a limit to the amount of time the protocol server can maintain change tracking records, it is possible for the indexing service to fall so far behind that it receives a "token is too old, and change records are lost" response from the protocol server. This response causes the indexing service to abandon the incremental indexing process and restart full traversal anew. Details of this incremental indexing process are shown in the third figure in section 3.1.4.

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 3: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a site (2) that is identified by a URL that is known by protocol clients. The protocol server endpoint is formed by appending "/_vti_bin/sitedata.asmx" to the URL of the site (2), for example, http://www.contoso.com/Repository/_vti_bin/sitedata.asmx.

For operations related to incremental change tracking, the client network account requires **Full Read** permissions as described in section 5.1.

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

1.6 Applicability Statement

This protocol supports full and incremental indexing of Web sites (2) that follow a site → subsite → list → list item hierarchical pattern of content organization and keep track of the changes of their structural elements.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in one area. This protocol uses multiple transports with SOAP as described in section 2.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

Protocol messages MUST be formatted as specified either in [SOAP1.1] 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. These namespaces are described in the following table.

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]

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

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

2.2.4 Complex Types

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

Complex type	Description
_sFPUrl	Specifies information about a **folder**, page, or document.
_sWebWithTime	Specifies URL and last modified information about a site (2).
ACL	Contains **permissions** granted to certain users and **groups (2)** for access to certain structural elements.
ArrayOf_sFPUrl	An array of **_sFPUrl** elements.
ArrayOf_sWebWithTime	An array of **_sWebWithTime** elements.
ArrayOfString	An array of arbitrary strings (their meaning and format depends on the operation).
ContentDatabaseMetadata	Specifies **GUID** and ChangeId of the content database.
Field	Specifies name and type of a field (2) of a list (1).
GroupDescription	Specifies detailed information about a group.
GroupMembership	For each group (2), contains a group description and the users who belong to this group (2).
Groups	Specifies an array of group information.
IndexedProperties	Contains an array of Property items.<1>
List	A structural schema of a **List** element – its fields (2), **views**, and permissions.
ListMetadata	Contains general details about a list (1).
Lists	An array of list (1) information.
permission	A pair consisting of a **memberId** and the permissions granted to this member. The **memberId** identifies a user or a group.
permissionsForACL	An array of **Permission** elements.
Property	Describes a property that is designated to be indexed.<2>
Schema	Defines list schema.
SOAPFaultDetails	Specifies the details of a SOAP fault.
SPWebAppMetadata	Contains metadata information for a **Web application (2)**.
SPSiteMetadata	Contains metadata information about a **site collection**.
UserDescription	Contains details about a user.
Users	An array of **UserDescription** elements.
View	Provides details about a view.
Views	An array of **View** elements.
WebMetadata	Contains metadata information about a site (2).

Complex type	Description
Webs	Specifies the URL, GUID, and last modified information about a collection of sites (2).

2.2.4.1   _sFPUrl

The **_sFPUrl** complex type contains information about a folder, document, or page.

```
<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>
```

Url: Specifies the **site-relative URL** of a folder, document, or page.

LastModified: Specifies the date and time as a locale-dependent short date value and longtime value, separated by a space in **Coordinated Universal Time (UTC)**, when the folder, document, or page was last modified.

IsFolder: Specifies whether the item is a folder. A value of **true** indicates it is a folder; a value of **false** indicates it is a document or a page.

2.2.4.2   _sWebWithTime

The **_sWebWithTime** type contains URL and last modified information of a site (2).

```
<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>
```

Url: Specifies the **absolute URL** of a site (2).

LastModified: Specifies the date and time, as a locale-dependent short date value and long time value, separated by a space in **UTC**, when the contents of the site (2) were last modified.

2.2.4.3   ACL

The **ACL** complex type contains information related to permissions; which user or group (2) has certain permissions about the element containing this ACL.

```
<s:complexType name="ACL">
  <s:sequence>
    <s:element name="permissions" type="tns:permissionsForACL" />
  </s:sequence>
</s:complexType>
```

permissions: Specifies a collection of permission information, as defined in section 2.2.4.17.

2.2.4.4 ArrayOf_sFPUrl

The **ArrayOf_sFPUrl** complex type is an array of **_sFPUrl** elements. The definition of **ArrayOf_sFPUrl** is as follows.

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

_sFPUrl: Specifies information about a folder, page, or document, as defined in section 2.2.4.1.

2.2.4.5 ArrayOf_sWebWithTime

The **ArrayOf_sWebWithTime** complex type contains an array of **_sWebWithTime** elements.

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

_sWebWithTime: Specifies information for a site (2), as defined in section 2.2.4.2.

2.2.4.6 ArrayOfString

ArrayOfString complex type is an array of strings.

```
<s:complexType name="ArrayOfString">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
              type="s:string" />
  </s:sequence>
</s:complexType>
```

string: One of the strings in the array.

2.2.4.7 ContentDatabaseMetadata

The **ContentDatabaseMetadata** complex type contains change information about the content database.

```
<s:complexType name="ContentDatabaseMetadata" >
  <s:attribute name="ChangeId" type="s:string" use="required" />
  <s:attribute name="ID" type="s:string" use="required" />
</s:complexType>
```

ChangeId: Specifies the latest change token associated with the content database. The structure of this change token is specific to the implementation of the protocol server.

ID: Specifies the GUID of the content database. This GUID MUST be enclosed in curly braces ({}).

2.2.4.8 Field

The **Field** complex type defines a list field (2), it has the following schema:

```
<s:complexType name="Field">
  <s:attribute name="Name" type="s:string" use="required"/>
  <s:attribute name="Type" type="s:string" use="required"/>
  <s:attribute name="TypeAsString" type="s:string" use="optional"/>
  <s:attribute name="Title" type="s:string" use="required"/>
  <s:attribute name="IsSurveyQuestion" type="tns:TrueFalseType" use="optional"/>
  <s:attribute name="ID" type="s:string" use="required"/>
  <s:attribute name="SourceId" type="s:string" use="required"/>
  <s:attribute name="Indexed" type="s:boolean" use="required"/>
  <s:attribute name="IsMultiValued" type="s:boolean" use="required"/>
</s:complexType>
```

Name: Specifies the name of the field (2).

Type: Specifies the type of the field (2); see [MS-WSSTS] section 2.5, for possible values.

TypeAsString: Specifies the type of the field as a simple string when the formal **Type** is not valid, or cannot be retrieved. If the **Type** of the field is valid, this attribute MUST be omitted.

Title: Specifies the title of the field (2).

IsSurveyQuestion: Specifies whether **base type** of the list (1) is **Survey** and whether the field (2) is contained in the base list (1) type. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4. If the **Type** of the field is not valid, this attribute MUST be omitted.

ID: Specifies the GUID of the field (2): MUST NOT be enclosed in curly brackets ({}).

SourceId: Specifies either the namespace that defines a build-in field (2) or, if it is a custom field, the GUID that identifies the list (1) or site (2) where it was created.

Indexed: Specifies a Boolean value that MUST be **true** if, and only if, the list field (2) is indexed in the content database.

IsMultiValued: Specifies a Boolean value that MUST be **true** if and only if the field (2) can contain multiple values.<3>

2.2.4.9 GroupDescription

The **GroupDescription** complex type contains details about a group. It has the following schema:

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

```
<s:simpleContent>
  <s:extension base="s:string">
    <s:attribute name="ID" type="s:int" use="required"/>
    <s:attribute name="Name" type="s:string" use="required"/>
    <s:attribute name="Description" type="s:string" use="required"/>
    <s:attribute name="OwnerID" type="s:int" use="required"/>
    <s:attribute name="OwnerIsUser" type="tns:TrueFalseType" use="required"/>
  </s:extension>
</s:simpleContent>
</s:complexType>
```

ID: Specifies the group identifier in **site membership**.

Name: Specifies the name of the group (2).

Description: Specifies the description of the group.

OwnerID: Specifies the unique identifier of the user (or the group) controlling this group membership.

OwnerIsUser: true if the owner is an individual, and **false** if the owner is a group.

2.2.4.10   GroupMembership

The **GroupMembership** complex type identifies the groups (2) and users in the site membership.

The **GroupMembership** element contains of one or more **Group** elements. Each **Group** element contains exactly one **GroupDescription** element and one **Users** element. Each **Users** element is a collection of zero or more **User** elements, one for each member of that **Group**.

The schema of **GroupMembership** is:

```
<s:complexType name='GroupMembership'>
  <s:sequence>
    <s:element name='Group' maxOccurs='unbounded'>
      <s:complexType>
        <s:sequence>
          <s:element name='Group' type='tns:GroupDescription'/>
          <s:element name='Users' type='tns:Users'/>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

GroupDescription: See section 2.2.4.9.

Users: See section 2.2.4.24.

2.2.4.10.1   Example of GroupMembership

```
<Groups >
  <Group >
    <Group  ID='3'
            Name='Home Owners'
```

```
            Description='Use this group to give people full control permissions to the
SharePoint site: Home'
            OwnerID='3'
            OwnerIsUser='False'>
    </Group>
    <Users >
      <User  ID='1'
             Sid='S-1-5-21-1721254763-462695806-1538882281-2307823'
             Name='Nancy Anderson'
             LoginName='fabrikam\nancy.anderson'
             Email='nancy.anderson@fabrikam.com'
             Notes=''
             IsSiteAdmin='True'
             IsDomainGroup='False'>
      </User>
      <User  ID='14'
             Sid='S-1-5-21-124525095-708259637-1543119021-565407'
             Name='Don Hall'
             LoginName='fabrikam\don.hall'
             Email='don.hall@fabrikam.com'
             Notes=''
             IsSiteAdmin='False'
             IsDomainGroup='False'>
      </User>
    </Users>
  </Group>
</Groups>
```

2.2.4.11   Groups

The **Groups** complex type contains an array of **Group** elements.

```
<s:complexType name="Groups">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="Group"
               type=" tns:GroupDescription" />
  </s:sequence>
</s:complexType>
```

Group: Specifies information about a group (2) as defined in section 2.2.4.9.

2.2.4.12   IndexedProperties

The **IndexedProperties** complex type contains an array of Property items.<4> It has the following schema.

```
<s:complexType name='IndexedProperties'>
  <s:sequence >
    <s:element  name='Property' type='tns:Property' maxOccurs='unbounded' />
  </s:sequence>
</s:complexType>
```

Property: Specifies Property information as defined in section 2.2.4.18.

2.2.4.13 List

The **List** complex type contains structural schema of a list (1): its fields, views, and permissions. It has the following schema:

```
<s:complexType name='List'>
  <s:sequence>
    <s:element name='Metadata' type='tns:ListMetadata'/>
    <s:element name='ACL' type='tns:ACL'/>
    <s:element name='Views' type='tns:Views'/>
    <s:element name='Schema' type='tns:Schema'/>
  </s:sequence>
</s:complexType>
```

Metadata: Specifies metadata information about a list (1), as defined in section 2.2.4.14.

ACL: If the site (2) containing the list (1) does not have an **external security provider** then ACL specifies a list (1) of permissions, as defined in section 2.2.4.3. Otherwise it MUST be empty.

Views: Specifies the list (1) of the **Views**, as defined in section 2.2.4.26.

Schema: An array of fields (2). It MUST be of type **Schema**, as defined in section 2.2.4.19.

2.2.4.14 ListMetadata

The **ListMetadata** complex type contains details about a list (1). It has the following schema:

```
<s:complexType name="ListMetadata">
  <s:attribute name="ID" type="s:string" use="required"/>
  <s:attribute name="Title" type="s:string" use="required"/>
  <s:attribute name="Description" type="s:string" use="required"/>
  <s:attribute name="DefaultViewUrl" type="s:string" use="required"/>
  <s:attribute name="DefaultViewItemUrl" type="s:string" use="required"/>
  <s:attribute name="DefaultTitle" type="s:string"  use="required"/>
  <s:attribute name="ScopeID" type="s:string" />
  <s:attribute name="RootFolder" type="s:string"  use="required"/>
  <s:attribute name="ReadSecurity" type="s:int" use="required"/>
  <s:attribute name="NoIndex" type="tns:TrueFalseType"  use="required"/>
  <s:attribute name="UseFormsForDisplay" type="tns:TrueFalseType" use="optional"/>
  <s:attribute name="BaseType" type="tns:ListBaseType"  use="required"/>
  <s:attribute name="BaseTemplate"  type="tns:ListBaseTemplate"   use="required"/>
  <s:attribute name="Author"  type="s:string"  use="required"/>
  <s:attribute name="AnonymousViewListItems" type="tns:TrueFalseType"
            use="required"/>
  <s:attribute name="AnonymousPermMask" type="s:unsignedLong" use="required"/>
  <s:attribute name="AnonymousPermMaskRaw" type="s:unsignedLong" use="required"/>
  <s:attribute name="AllowAnonymousAccess" type="tns:TrueFalseType"
            use="required"/>
  <s:attribute name="LastModified" type="s:string" use="required"/>
  <s:attribute name="ItemCount" type="s:int"  use="required"/>
  <s:attribute name="CRC" type="s:unsignedInt"  use="required"/>
  <s:sequence>
    <s:element name="IndexedProperties" minOccurs="1" type="tns: IndexedProperties"/>
  </s:sequence>
</s:complexType>
```

ID: Specifies the GUID of the list. This GUID MUST be enclosed in curly braces ({}).

Title: Specifies the title of the list.

Description: Specifies the description of the list.

DefaultViewUrl: Specifies the **server-relative URL** of the page containing the **default list view** of the list.

DefaultViewItemUrl: Specifies the server-relative URL of the **display form** of the list**.**

DefaultTitle: Specifies whether the **Title** is the default title of the list.

ScopeID: The **ScopeID** of this list, see section 2.2.6.1. It MUST be enclosed in curly braces ({}). The protocol server MUST return it only if the site (2) containing the list does not have an external security provider.

RootFolder: Specifies the name of the root folder of the list.

ReadSecurity: Specifies the read security setting. Possible values include the following:

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

NoIndex: A Boolean value that MUST be **true** if, and only if, list cannot be crawled for indexing. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

UseFormsForDisplay: If **true**, forms will be used for display. Otherwise, forms will not be used for display.<5>

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

BaseTemplate: Specifies the name of the base template (end-user look-and-feel) of the list. It MUST be of type **ListBaseTemplate**, as defined in section 2.2.5.2.

Author: Specifies the name of the user who has created this list.

AnonymousViewListItems: A Boolean value that MUST be **true** if, and only if, anonymous users can view the list items of the list. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

AnonymousPermMask: Specifies the permissions granted to anonymous users for the list, a combination of WSS Rights bitmask flags as defined in [MS-WSSFO2] section 2.2.2.14.

AnonymousPermMaskRaw: Specifies the permissions granted to anonymous users for the list, a combination of WSS Rights bitmask flags as defined in [MS-WSSFO2] section 2.2.2.14.

AllowAnonymousAccess: A Boolean value that MUST be true if, and only if, anonymous users have access permissions for the list. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

LastModified: Specifies the date and time, as a [ISO-8601] separate date and time UTC value, when the list was last modified.

ItemCount: Specifies the number of list items in the list.

CRC: Specifies the hash code of the fields (2) excluded from indexing in the list. If the CRC changes, it means that some fields are included (excluded from indexing).

IndexedProperties: Specifies indexed properties for this list as described in section 2.2.4.12.

2.2.4.15 Lists

The **Lists** complex type contains an array of list (1) information.

```
<s:complexType name="Lists">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="List" >
     <s:complexType>
      <s:attribute name="ID" type="s:string" use="required" />
      <s:attribute name="LastModified" type="s:string" use="required" />
      <s:attribute name="DefaultViewUrl" type="s:string" use="required" />
     </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

List.ID: Specifies the GUID of the list (1). This GUID MUST be enclosed in curly braces ({}).

List.LastModifed: Specifies the date and time, as a [ISO-8601] separate date and time UTC value, when the list (1) was last modified.

List.DefaltViewUrl: Specifies the server-relative URL of the page containing the default list view of the list (1).

2.2.4.16 permission

The **permission** complex type describes permissions granted to some site (2) membership member (user or group (2)). It has the following schema.

```
<s:complexType name="permission">
  <s:attribute name="memberid" type="s:int" use="required" />
  <s:attribute name="mask" type="s:unsignedLong" use="required" />
</s:complexType>
```

memberid: Specifies the unique identifier of a member of site (2) membership. This specifies either a user or a group (2).

mask: Specifies a mask permission flag as defined in the WSS Rights bitmask flags in [MS-WSSFO2] section 2.2.2.14.

2.2.4.17 permissionsForACL

The **permissionsForACL** complex type contains a collection of permissions defined for a structural element (site collection, site (2), list (1), list item). It has the following schema.

```
<s:complexType name='permissionsForACL'>
  <s:sequence >
    <s:element  name='permission' type='tns:permission'   maxOccurs='unbounded' />
```

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

permission: Specifies permission information, as defined in section 2.2.4.16.

2.2.4.18 Property

The **Property** complex type describes a property that is designated to be indexed.<6> It has the following schema.

```
<s:complexType name="Property">
  <s:attribute name="Name" type="s:string" use="required" />
  <s:attribute name="Type" type="s:string" use="required" />
  <s:attribute name="Value" type="s:string" use="required" />
</s:complexType>
```

Name: Name of the property.

Type: Type of the property. MUST be either empty string or one of the following values **System.String**, **System.DateTime**, **System.Int32**.

Value: The value of the property.

2.2.4.19 Schema

The **Schema** complex type defines list schema. It has the following schema.

```
<s:complexType name='Schema'>
  <s:sequence >
    <s:element  name='Field' type='tns:Field' maxOccurs='unbounded' />
  </s:sequence>
</s:complexType>
```

Field: Specifies information about a field (2) in the list (1), as defined in section 2.2.4.8.

2.2.4.20 SOAPFaultDetails

SOAPFaultDetails complex type specifies the details of a SOAP fault.

```
<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema" targetNamespace="
http://schemas.microsoft.com/sharepoint/soap">
   <s:complexType name="SOAPFaultDetails">
      <s:sequence>
         <s:element name="errorstring" type="s:string"/>
         <s:element name="errorcode" type="s:string" minOccurs="0" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:schema>
```

errorstring: Specifies a human-readable text explaining the application-level fault.

errorcode: Specifies the hexadecimal representation of a 4-byte result code.

2.2.4.21 SPSiteMetadata

The **SPSiteMetadata** complex type contains metadata information for a site collection.

```
<s:complexType name="SPSiteMetadata">
  <s:attribute name="SiteSubscriptionId" type="s:string" />
  <s:attribute name="URL" type="s:string" use="required" />
  <s:attribute name="ID" type="s:string" use="required" />
  <s:attribute name="LastModified" type="s:string" use="required" />
  <s:attribute name="PortalURL" type="s:string" use="required" />
  <s:attribute name="UserProfileGUID" type="s:string" use="required" />
  <s:attribute name="RootWebId" type="s:string" use="required" />
  <s:attribute name="ChangeId" type="s:string" use="required" />
  <s:attribute name="ContentDatabaseId" type="s:string" use="required" />
</s:complexType>
```

SiteSubscriptionId: Specifies the **site subscription** GUID of the site collection<7>.

URL: Specifies the absolute URL of the site collection.

ID: Specifies the GUID of the site collection. This GUID MUST be enclosed in curly braces ({}).

LastModified: Specifies the date and time, as an [ISO-8601] separate date and time UTC value, when the site collection was last modified.

PortalURL: Specifies the URL of the portal associated with the site collection.

UserProfileGUID: If the site (2) is a **personal site**<8> it MUST specify the GUID of the user profile who owns the site (2). Otherwise, it MUST be empty.

RootWebId: Specifies the GUID of the site (2) of the site collection.

ChangeId: Specifies the latest change token of the site collection.

ContentDatabaseId: Specifies the GUID of the content database in which this site collection is contained. This GUID MUST be enclosed in curly braces ({}).

2.2.4.22 SPWebAppMetadata

The **SPWebAppMetadata** complex type contains metadata information for a Web application (2).

```
<s:complexType name="SPWebAppMetadata">
  <s:attribute name="ID" type="s:string" use="optional" />
  <s:attribute name="Version" type="s:string" use="optional" />
  <s:attribute name="URL" type="s:string" use="required" />
  <s:attribute name="URLZone" type="s:string" use="optional" />
  <s:attribute name="URLIsHostHeader" type="tns:TrueFalseType" use="optional" />
</s:complexType>
```

ID: Specifies the GUID of the Web application (2). This GUID MUST be enclosed in curly braces ({})<9>.

Version: Specifies the version of the Web application (2). This value MUST be a version string, formatted as a 64-bit unsigned integer.<10>

URL: Specifies the absolute URL of the Web application (2).

URLZone: Specifies the zone of the Web application (2).<11>

URLIsHostHeader: Specifies whether the Web application (2) **URL** is a host-header URL.<12>

2.2.4.23   UserDescription

The **UserDescription** complex type contains details about a user who is a member of a certain group (2). The **UserDescription** type is used to relate the **principal (1)** with a site membership entry. It has the following schema:

```
<s:complexType name='UserDescription'>
  <s:attribute name='Sid' type='s:string' use='required'/>
  <s:attribute  name='Name' type='s:string' use='required' />
  <s:attribute  name='LoginName' type='s:string' use='required' />
  <s:attribute  name='IsDomainGroup' type= 'tns:TrueFalseType' use='required'/>
  <s:attribute  name='Email' type='s:string' use='required'/>
  <s:attribute  name='Notes'  type='s:string' use='required' />
  <s:attribute  name='ID' type='s:int' use='required' />
  <s:attribute  name='IsSiteAdmin' type= 'tns:TrueFalseType' use='required'/>
  <s:attribute  name='Flags' type='s:int' use='required'/>
  <s:attribute  name='UserKey' type='s:string' use='optional' />
</s:complexType>
```

Sid: Specifies the **security identifier (SID)** of the user.

Name: Specifies the name of the principal (2).

LoginName: Specifies the **login name** of the user.

IsDomainGroup: A Boolean value that specifies whether the principal (2) is a user or a group (2). A value of **false** indicates a user, while a value of **true** indicates a group (2).

Email: Contains an e-mail address in standard SMTP format, for example, nancy.anderson@fabrikam.com. If no e-mail address is known to the server (for example, if the principal (2) is a group), **Email** contains an empty string.

Notes: Specifies arbitrary notes about that principal (1).

ID: Specifies the **user identifier** in site membership.

IsSiteAdmin: A Boolean value that is **true** when this user is the **site collection administrator**, and **false** otherwise.

Flags: An integer that specifies the user's options, as defined in [MS-WSSFO2] section 2.2.2.11.<13>

UserKey: A string that specifies the Provider User Key for this user.

2.2.4.24   Users

The **Users** complex type contains an array of elements of type **UserDescription**.

```
<s:complexType name="Users">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="User"
```

```
            type="tns:UserDescription" />
    </s:sequence>
  </s:complexType>
```

User: Specifies information about a user as defined in section 2.2.4.23.

2.2.4.25 View

The **View** complex type contains minimal details about a view. It has the following schema:

```
<s:complexType name="View">
  <s:attribute name="URL" type="s:string" use="required"/>
  <s:attribute name="Title" type="s:string"  use="required"/>
  <s:attribute name="ID" type="s:string" use="required"/>
  <s:attribute name="AllowedInRoot" type="tns:TrueFalseType" use="optional"/>
</s:complexType>
```

URL: Specifies a site-relative URL of the view, for example, "Shared Documents/Forms/AllItems.aspx". This URL MUST NOT be URL-encoded.

Title: Specifies the title of the view, for example, "All Documents".

ID: Specifies the GUID of the view. This GUID MUST be enclosed in curly braces ({}).

AllowedInRoot: If **true**, this view is allowed in root; otherwise, this view is not allowed in root.<14>

2.2.4.26 Views

The **Views** complex type contains an array of **View** elements.

```
<s:complexType name='Views'>
  <s:sequence >
    <s:element  name='View' type='tns:View' maxOccurs='unbounded'/>
  </s:sequence>
</s:complexType>
```

View: Specifies information about a view, as defined in section 2.2.4.25.

2.2.4.27 WebMetadata

The **WebMetadata** complex type contains metadata information about the site (2).

```
<s:complexType name="WebMetadata">
  <s:attribute name="URL" type="s:string" use="required" />
  <s:attribute name="LastModified" type="s:string" use="required" />
  <s:attribute name="Created" type="s:string" use="required" />
  <s:attribute name="ID" type="s:string" use="required" />
  <s:attribute name="Title" type="s:string" use="required" />
  <s:attribute name="Description" type="s:string" use="required" />
  <s:attribute name="LogoURL" type="s:string" />
  <s:attribute name="LogoDescription" type="s:string" />
  <s:attribute name="Author" type="s:string" use="required" />
  <s:attribute name="Language" type="s:unsignedShort" use="required" />
```

```
        <s:attribute name="CRC" type="s:unsignedInt" use="required" />
        <s:attribute name="NoIndex" type="s:string" use="required" />
        <s:attribute name="DefaultHomePage" type="s:string" use="required" />
        <s:attribute name="DefaultHomePageListId" type="s:string" use="optional" />
        <s:attribute name="DefaultHomePageItemId" type="s:integer" use="optional" />
        <s:attribute name="ExternalSecurity" type="tns:TrueFalseType" use="required" />
        <s:attribute name="ScopeID" type="s:string" />
        <s:attribute name="AllowAnonymousAccess" type="tns:TrueFalseType"
                    use="required" />
        <s:attribute name="AnonymousViewListItems" type="tns:TrueFalseType"
                    use="required" />
        <s:attribute name="AnonymousPermMask" type="s:unsignedLong" use="required" />
        <s:attribute name="AnonymousPermMaskRaw" type="s:unsignedLong" use="required"/>
        <s:sequence>
          <s:element name="IndexedProperties" minOccurs="1" type="tns: IndexedProperties"/>
        </s:sequence>
        <s:attribute name="UIVersion" type="s:int" use="required" />
  </s:complexType>
```

URL: Specifies the absolute URL of the site (2).

LastModified: Specifies the date and time, as an [ISO-8601] separate date and time UTC value, when the site (2) was last modified.

Created: Specifies the date and time, as an [ISO-8601] separate date and time UTC value, when the site (2) was created.

ID: Specifies the GUID of the site (2). The GUID MUST be enclosed in curly braces ({}).

Title: Specifies the title of the site (2).

Description: Specifies the description of the site (2).

LogoURL: Specifies the server-relative URL of a picture that represents a site (2).<15>

LogoDescription: Specifies a text description of the **LogoURL.**<16>

Author: Specifies the name of the user who created the site (2).

Language: Specifies the **language code identifier (LCID)** for the language that is used on the site (2).

CRC: Specifies the hash code of the crawling information of the site (2). If the **CRC** changes, it means the crawling properties of the site (2) have changed.

NoIndex: A Boolean value that MUST be **true** if, and only if, the site (2) cannot be crawled for indexing. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

DefaultHomePage: Specifies the site-relative URL of the **Welcome page** of the site (2).

DefaultHomePageListId: Specifies the GUID of the list (1) that contains the Welcome page of the site (2). If the Welcome page is not contained in a list or cannot be retrieved, this attribute MUST be omitted.

DefaultHomePageItemId: Specifies the **list item identifier** of the list item that represents the Welcome page of the site (2). If the Welcome page is not a list item or cannot be retrieved, this attribute MUST be omitted.

ExternalSecurity: A Boolean value that MUST be **true** if, and only if, the site (2) has an external security provider. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

ScopeID: Specifies the **ScopeID** of this site (2), as defined in section 2.2.6.1. It MUST be enclosed in curly braces ({}). The protocol server MUST return this only if the site (2) does not have an external security provider.

AllowAnonymousAccess: A Boolean value that MUST be **true** if, and only if, anonymous users have access permissions for the site (2). It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

AnonymousViewListItems: A Boolean value that MUST be **true** if, and only if, anonymous users can view the list items of the site (2) lists. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

AnonymousPermMask: Specifies the mask of the permissions for anonymous users of the site (2). See [MS-WSSFO2] section 2.2.2.14.

IndexedProperties: Specifies indexed properties for this list, as described in section 2.2.4.12.

UIVersion: Specifies the user interface version for the site.<17>

2.2.4.28 Webs

The **Webs** complex type contains an array of information about all the subsites of the site (2).

```
<s:complexType name="Webs">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="Web">
      <s:complexType>
        <s:attribute name="URL" type="s:string" use="required" />
        <s:attribute name="ID" type="s:string" use="required" />
        <s:attribute name="LastModified" type="s:string" use="required" />
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

Web.URL: Specifies the absolute URL of the subsite.

Web.ID: Specifies the GUID of the subsite. The GUID MUST be enclosed in curly braces ({}).

Web.LastModified: Specifies the data and time, as a [ISO-8601] separate date and time UTC value, when the subsite was last modified.

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	Type of list (1).

Simple type	Description
ListBaseTemplate	Type of list (1) templates.
ObjectType	Type of the structural element in **GetContent**, **GetChange** operations.
TrueFalseType	Replacement of Boolean type, having values **true** and **false**.

2.2.5.1 ListBaseType

The **ListBaseType** defines an enumeration of the base types of a list (1). Its schema is:

```
<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>
```

The following table defines the allowable values for **ListBaseType**:

Value	Meaning
UnspecifiedBaseType	Unspecified list type.
GenericList	Generic list.
DocumentLibrary	Document library list.
Unused	Unused.
DiscussionBoard	Discussion board list.
Survey	Survey list.
Issue	Issue list.

2.2.5.2 ListBaseTemplate

The **ListBaseTemplate** defines an enumeration of the templates of a list. Its schema is:

```
<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="UserInformation"/>
        <s:enumeration value="WebPartCatalog"/>
        <s:enumeration value="ListTemplateCatalog"/>
        <s:enumeration value="XMLForm"/>
        <s:enumeration value="MasterPageCatalog"/>
        <s:enumeration value="NoCodeWorkflows"/>
        <s:enumeration value="WorkflowProcess"/>
        <s:enumeration value="WebPageLibrary"/>
        <s:enumeration value="CustomGrid"/>
        <s:enumeration value="DataConnectionLibrary"/>
        <s:enumeration value="WorkflowHistory"/>
        <s:enumeration value="GanttTasks"/>
        <s:enumeration value="Meetings"/>
        <s:enumeration value="Agenda"/>
        <s:enumeration value="MeetingUser"/>
        <s:enumeration value="Decision"/>
        <s:enumeration value="MeetingObjective"/>
        <s:enumeration value="TextBox"/>
        <s:enumeration value="ThingsToBring"/>
        <s:enumeration value="HomePageLibrary"/>
        <s:enumeration value="Posts"/>
        <s:enumeration value="Comments"/>
        <s:enumeration value="Categories"/>
        <s:enumeration value="IssueTracking"/>
        <s:enumeration value="AdminTasks"/>
      </s:restriction>
    </s:simpleType>
```

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

Value	Meaning
InvalidType	List template that is not valid.
GenericList	Generic list template.
DocumentLibrary	Document library list template.
Survey	Survey list template.
Links	Links list template.
Announcements	Announcements list template.
Contacts	Contacts list template.
Events	Events list template.
Tasks	Tasks list template.
DiscussionBoard	Discussion board list template.

Value	Meaning
PictureLibrary	Picture library list template.
DataSources	Data sources list template.
WebTemplateCatalog	Web template catalog list template.
UserInformation	User information list template.
WebPartCatalog	Web part gallery list template.
ListTemplateCatalog	List template catalog list template.
XMLForm	XML form list template.
MasterPageCatalog	Master page catalog list template.
NoCodeWorkflows	No code workflows list template.
WorkflowProcess	Workflow process list template.
WebPageLibrary	Web page library list template.
CustomGrid	Custom grid list template.
DataConnectionLibrary	Data connection library list template.
WorkflowHistory	Workflow history list template.
GanttTasks	Gantt tasks list template.
Meetings	Meetings list template.
Agenda	Agenda list template.
MeetingUser	Meeting user list template.
Decision	Decision list template.
MeetingObjective	Meeting objective list template.
TextBox	TextBox list template.
ThingsToBring	Things to bring list template.
HomePageLibrary	Home page library list template.
Posts	Posts list template.
Comments	Comments list template.
Categories	Categories list template.
IssueTracking	Issue tracking list template.
AdminTasks	Admin tasks list template.

2.2.5.3 ObjectType

The **ObjectType** simple type defines the type of the object whose content is requested by **GetContent** or **GetChanges** operations. Its schema is:

```
<s:simpleType name="ObjectType">
  <s:restriction base="s:string">
    <s:enumeration value="VirtualServer"/>
    <s:enumeration value="ContentDatabase"/>
    <s:enumeration value="SiteCollection"/>
    <s:enumeration value="Site"/>
    <s:enumeration value="List"/>
    <s:enumeration value="Folder"/>
    <s:enumeration value="ListItem"/>
    <s:enumeration value="ListItemAttachments"/>
  </s:restriction>
</s:simpleType>
```

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

Value	Meaning
VirtualServer	Web application
ContentDatabase	content database
SiteCollection	site collection
Site	site (2)
List	list (1)
Folder	folder
ListItem	list item
ListItemAttachments	attachments

2.2.5.4 TrueFalseType

The **TrueFalseType** simple type is used to represent **true** or **false** Boolean values.

```
<s:simpleType name="TrueFalseType">
  <s:restriction base="s:string">
    <s:enumeration value="True"/>
    <s:enumeration value="False"/>
  </s:restriction>
</s:simpleType>
```

Value	Description
True	Boolean value **true**.
False	Boolean value **false**.

2.2.6 Attributes

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

Value	Description
ScopeID	A GUID used to identify a unique permission list..

2.2.6.1 ScopeID

A GUID used to identify a unique permission list. Protocol clients use **ScopeID** to improve the efficiency of managing the permission lists that governs what users have access to what object on the site (2).

Normally, protocol clients save unique permission lists in a structure where they are indexed by this **ScopeID**. Protocol clients compute permission lists from ACL information provided by the protocol server. Because this computation tends to be costly, and the same permission lists often apply to more than one element on the site (2), protocol clients save the permission lists in a structure where they can be indexed (looked up) using the **ScopeID**. Subsequently, protocol clients obtain the **ScopeID** from the protocol server and use it to look up the operative permission list. If a the protocol client finds a permission list corresponding to the **ScopeID**, the protocol client uses the permission list directly, saving the work of computing the permission list from ACL information.

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3   Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

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

This protocol allows protocol servers to notify protocol clients of application-level faults using SOAP faults. This protocol allows protocol servers to provide additional details for SOAP faults by including either a **detail** element as specified in [SOAP1.1] section 4.4, or a **Detail** element as specified in [SOAP1.2/1] section 5.4.5, which conforms to the XML schema of the **SOAPFaultDetails** complex type specified in section 2.2.4.20. Except where specified, these SOAP faults are not significant for interoperability, and protocol clients can interpret them in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults either using HTTP status codes or using SOAP faults as specified previously in this section.

3.1   Site Data Web Service Details

3.1.1   Abstract Data Model

This protocol supports indexing of sites (2) which conform to a hierarchical pattern of content organization. This section specifies the required 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. These elements conform to the following organization:

- A site collection, sometimes called a root site, consists of subsites, or simply sites (2), sometimes called Webs.

- A site (2) can contain other subsites.

- There is a content database used by a site collection, to record site (2) structure and keep track of changes. All subsites share the same content database as used by the site collection.

- sites (2) contain lists (1), both predefined (for example, Users) and custom lists (1). Each list (1) has a schema which prescribes the **list item** fields.

- List (1) contain **list items**. Each **list item** has the same fields (2) (shown as columns) prescribed by **list schema**. Optionally, **list items** can have any number of attachments (arbitrary files, perhaps in proprietary vendor-specific formats). There are predefined fields (2) that every list (1) has, for example, ID, and Modified - last modification time. Those fields (2) obtain their values automatically and have predefined meanings.

- Document libraries are special form of lists (1). They contain documents. Documents are arbitrary files, sometimes in proprietary vendor-specific formats, annotated with fields (2) prescribed by this document library schema.

- List (1) can contain folders and folders can contain subfolders.

- All elements of a site (2) are identifiable either by URL or by their identifiers, typically **GUIDs**. Identifiers might be opaque, however they are immutable for the whole lifecycle of an element,

be it site collection, site (2), list (1), folder, or list item, while the URL might change over time. For example, a list (1) or document can be renamed, thus changing URLs.

3.1.2 Timers

None.

3.1.3 Initialization

The protocol server endpoint is formed by appending "/_vti_bin/sitedata.asmx" to the URL of the site (2), for example, http://www.contoso.com/Repository/_vti_bin/sitedata.asmx.<18>

3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
EnumerateFolder	Returns information about the immediate child folders, documents, and pages of the specified folder.
GetAttachments	Returns information about all the attachments of the specified list item.
GetChanges	Returns the changes that occurred on site collection content from one moment in time to another.
GetChangesEx	Returns the changes that occurred on site collection content from one moment in time to another, as well as additional metadata related to the site collection.
GetContent	Returns change information (**ChangeId**) about the content database and site collection, as well as information about the Web application (2), site collection, site (2), list (1), list item, folder and attachment.
GetContentEx	Returns change information (**ChangeId**) about the content database and site collection, as well as information about the Web application (2), site collection, site (2), list (1), list item, folder, and attachment, and enables better control of the protocol with parameters.
GetList	Returns general information, schema of the fields (2) and access right permissions for a given list.
GetListCollection	Returns general information about all lists of the **context site**.
GetListItems	Returns all or 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	Returns the URL of the site collection and the site (2) to which the specified URL belongs.
GetSiteUrl	Returns the GUID and URL of the site collection to which the specified URL belongs.
GetURLSegments	Returns the identifiers of the object pointed by the specified URL.
GetWeb	Returns information about the context site.

Operations **GetSite**, **GetWeb**, **EnumerateFolder**, **GetListCollection**, **GetList**, **GetListItems**, **GetAttachments** are used for full content traversal and indexing. They can be called specific traversal operations.

Operations **GetSiteAndWeb**, **GetSiteUrl**, **GetURLSegments** are auxiliary; they are used to parse URLs.

Operations **GetContent**, **GetContentEx**, **GetChanges** and **GetChangesEx** are used for full and then incremental content indexing.

Note that **GetContent** operation is polymorphic; it can be used (when called with proper parameters) instead of specific traversal operations: **GetSite**, **GetWeb**, **EnumerateFolder**, **GetListCollection**, **GetList**, **GetListItems**, **GetAttachments**. **GetContent** fully covers specific traversal operations functionality, albeit details of input parameters and syntax of the response might differ.
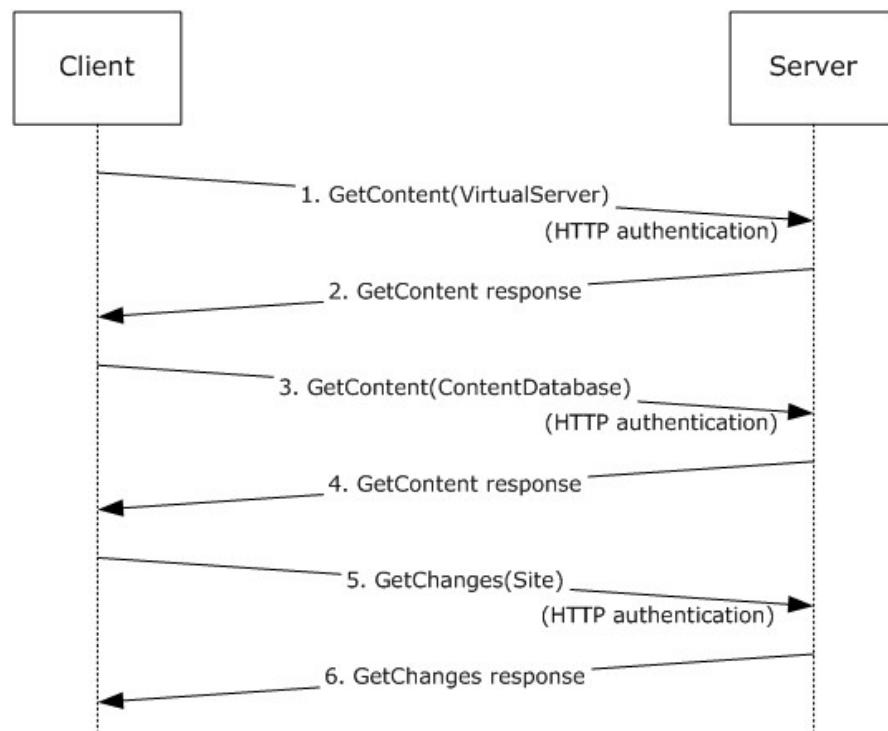
Recommended sequencing of the operations, briefly outlined in section 1.3 is explained here in detail.

Sequence for Establishing Indexing Context

Indexing starts with the indexing service establishing context. All the indexing service needs to know is the URL of a page from the site collection. For example:

```
http://www.Fabrikam.com:8080/Subsite/Shared%20Documents/Forms/AllItems.aspx
```

The indexing service identifies the site collection of interest, and obtains details about its content database and current change timestamp, using the steps as shown in the following diagram.

Figure 4: Establishment of indexing context

1. Indexing service extracts the host name with port number from the URL (for example www.Fabrikam.com:8080) and sends the **GetContent** request to that host, passing "VirtualServer" as the value of the *objectType* parameter.

2. The protocol server responds with a message that contains GUIDs of all the content databases servicing that Web application (2).

3. The indexing service sends a **GetContent** request, passing "ContentDatabase" as the value of the *objectType* parameter, and passing one of the identifiers obtained in step 2 as the value the of the *objectId* parameter.

4. The protocol server sends a response message, enumerating all site collections serviced by that content database: their URLs and identifiers (GUIDs), as well as the current **ChangeId**, (in the context of full content database). Assume that the response contains an entry such as the following:

   ```
   <Site URL=" http://www.Fabrikam.com:8080"
   ID="f945dc15-825e-4c16-8d12-bed6413dc61a" />
   ```

   Also, assume that this site collection is exactly the one that the indexing service wants to index. Then proceed to step 5. Otherwise, if the site collection is not serviced by this content database, repeat from step 3 until the URL of interest is found.
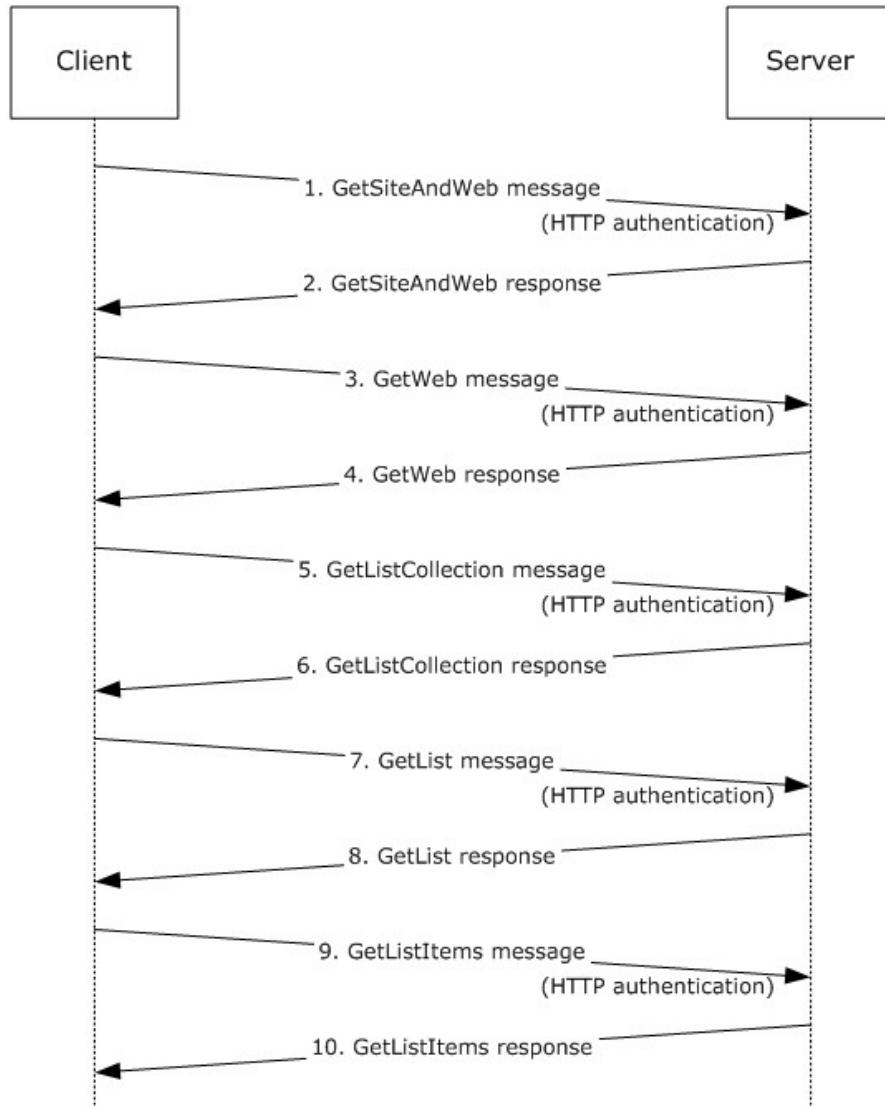
5. The indexing service sends the request **GetChanges** to the site collection with URL http://www.Fabrikam.com:8080, passing "Site" as the value of the *objectType* parameter, optionally the identifier of the content database obtained in step 2 as the value of the *contentDatabaseId* parameter (it can be left blank), and the current **ChangeId** (in the context of content database) obtained in step 4 as the value of the *LastChangeId* parameter, leaving the *CurrentChangeId* parameter blank.

6. The protocol server sends a response message containing the **LastChangeID** (meaning the last known change) in the context of the site collection, namely http://www.Fabrikam.com:8080. This **ChangeId** (hereafter referred to as the **StartChangeId**) is required to perform incremental change processing in the future.

Now that the context is established, indexing starts with full traversal, when the indexing service attempts to snapshot the whole structure of the site collection and its current content.

Sequence of full traversal

The following diagram depicts the sequence of this protocol's operations required for full traversal. A brief explanation of each message follows. Note that the following diagram explains only one branch of depth-first [KNUTH] full traversal scenario. For example, having found some lists among others returned by the **GetListCollection** response, first explore its list items. Complete depth-first traversal explores all the remaining lists then, which is not shown.

Note that this protocol does not prescribe depth-first traversal to build a snapshot of the site (2) content. Any systematic traversal sequence, for example, breadth-first [KNUTH], will do, however depth-first traversal is simpler to explain.

Figure 5: Outline of full site traversal scenario

1. The protocol client sends **GetSiteAndWeb** input message (with any site (2) – using referring URL as parameter) to get URL for site collection and the site (2) that this URL refers to.

2. In response, the protocol server sends a message containing those URLs.

3. The protocol client sends a **GetWeb** request message, targeting the site collection of interest, for example, http://www.fabrikam.com:8080.

4. The protocol server sends a response message, which among other things contains the list of all subsites (direct child objects) of the site collection targeted by the request. For example, one of them can be http://www.fabrikam.com:8080/subsite.

5. To perform a depth-first exploration of that subsite, the protocol client sends a **GetListCollection** request message targeting the first subsite.

6. The protocol server sends the **GetListCollection** response message, containing information about all lists, and in particular, their identifiers.

7. To perform a depth-first exploration of the first list, the protocol client sends a **GetList** message passing this subsite URL and **list identifier** as parameters.

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

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

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

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 itself.

Recommended sequence of incremental indexing

This section specifies the recommended sequence of steps to keep the index up-to-date.

The **GetChanges** and the **GetContent** operations of this protocol provide the ability to obtain incremental change reports, rather than repeating full traversal of the site (2).

This protocol presumes that the protocol server keeps track of all changes to site (2) content in a content database. The content database can service multiple site collections, however a site collection is always supported by a single content database. Although a site collection can be comprised of multiple sites (2) and subsites, all are considered part of the site collection. The protocol server uses the content database to keep track of all content changes to all sites and subsites in the site collection.

This protocol provides protocol clients with two levels of granularity when obtaining incremental changes: the whole content database level, and the single site collection level. This flexibility means that protocol clients can obtain incremental changes to the entire content database, or incremental changes to just a single site collection.

Protocol clients typically run multiple processes that obtain incremental changes to multiple site collections in parallel. To facilitate this capability, this protocol requires a mechanism to identify the context of any given incremental change. This document and this protocol define the term, **change tracking space** to identify the context of any incremental change. The change tracking space identifies the context—both the site collection and the granularity—of any incremental change.

This protocol associates every incremental change with a specific change tracking space. A given change tracking space can specify the whole content database or it can specify a single, particular site collection. Subsites do not have their own change spaces, their changes belong to the site collection.

The protocol server MUST capture every change to site collection content in an atomic change record and mark each record with an opaque token, called a **ChangeId.** This **ChangeId** has the following properties:

▪ Every **ChangeId** is unambiguously associated with a particular change tracking space.

▪ All **ChangeId** tokens that belong to the same change tracking space MUST be ordered in the sequence of change occurrence time. **ChangeId** tokens that belong to different change tracking spaces might not be comparable.

- Protocol clients (indexing services) MUST treat **ChangeId** tokens as opaque strings and SHOULD NOT interpret the **ChangeId** token structure.

- Protocol clients (indexing services) MUST pass to the protocol server only **ChangeId** tokens previously received from the same protocol server.

- Protocol clients (indexing services) use **ChangeId** tokens to request the set of changes that have occurred in a particular change tracking space because a specified **ChangeId** token. **ChangeId** tokens serve no other purpose.

- The retention time for change records (how long the change records remain in the content database) is an implementation detail. If the protocol server receives a request for changes it has not retained, it MUST indicate that a **ChangeId** specified in a request for incremental changes is too old, meaning the change records are no longer retained. In this case, the protocol client SHOULD abandon incremental indexing using any previously-obtained **ChangeId** tokens in that change tracking space, and SHOULD restart with full traversal.

- The protocol does not limit the number of changes that can occur after a certain **ChangeId**. To make the dialog round-trip time predictable, the protocol client sets the **Timeout** value in each request, limiting the time the protocol server can spend preparing the answer. The protocol server begins to build a change report starting from the oldest changes, and once the timeout is over, returns a report that includes all changes from the **LastChangeId** token (as received from the protocol client) and up to the new **LastChangeId** token, provided by the protocol server in the response. The protocol client MUST continue the process by supplying this new **LastChangeId** token. The protocol server MUST guarantee that by following this procedure, eventually the protocol client receives all changes.

Structural elements

From the change reporting point of view, the site collection is a structure consisting of sites (2), subsites and lists. Lists consist of views and list items. Lists can have folders, while list items can have **files** as attachments. This document defines the term, structural elements, to refer to all these components of a site (2). The site collection itself is a structural element of the content database.

Note that document libraries, documents, pages and other **content types** are just special cases of lists and list items respectively and are treated in the same manner.

The protocol server MUST consider a structural element to have changed only if the following conditions are satisfied:

- The structural element has changed, for example, it was added, deleted, directly modified or renamed.

- One or more components that this element consists of have changed.

- Permissions to access this element have changed.

Change notifications

This document uses the term change notification to specify a fragment of a change report, describing the change to a particular structural element. The element that changed is referred to as the subject of this notification.

All change notifications are marked with the following attributes:

- **Change:** Type of the change, as defined in section 3.1.4.3.5.1.

- **ItemCount:** Number of subordinated structural elements (direct and indirect), whose change notifications are included into that same report.

All change reports, full or partial, always contain one, and only one, root change notification. The subject of the root notification is the change tracking space. This change tracking space can specify the whole content database or a particular site collection.

If the root element is marked as "Unchanged", `ItemCount = 0`, it means that no direct or indirect changes are included in that report. Otherwise, root change notification MUST contain one or more change notifications for its direct components. These component change notifications typically contain nested change notifications.

The number of change notifications to include in one change report is an implementation detail of the protocol server. However, the change notifications created by the protocol server MUST conform to the following rules:

- If a change notification about any structural element is included in the report, it is always embedded in a change notification addressing the parent of the element. This is because the parent element is considered changed and MUST have **ItemCount** greater than zero. In other words, change notifications cannot be orphans, they are always embedded in parent change notifications.

- No change notifications addressing unchanged elements can be included in the report, with the exception of the root element.

- Any given changed element, that is, the change notification addressing this element, can only appear in the change report once. All reported changes of its structural components MUST be embedded into that notification.

In other words, the change report MUST constitute a contiguous sub-tree of the tree of all structural elements in the change tracking space.
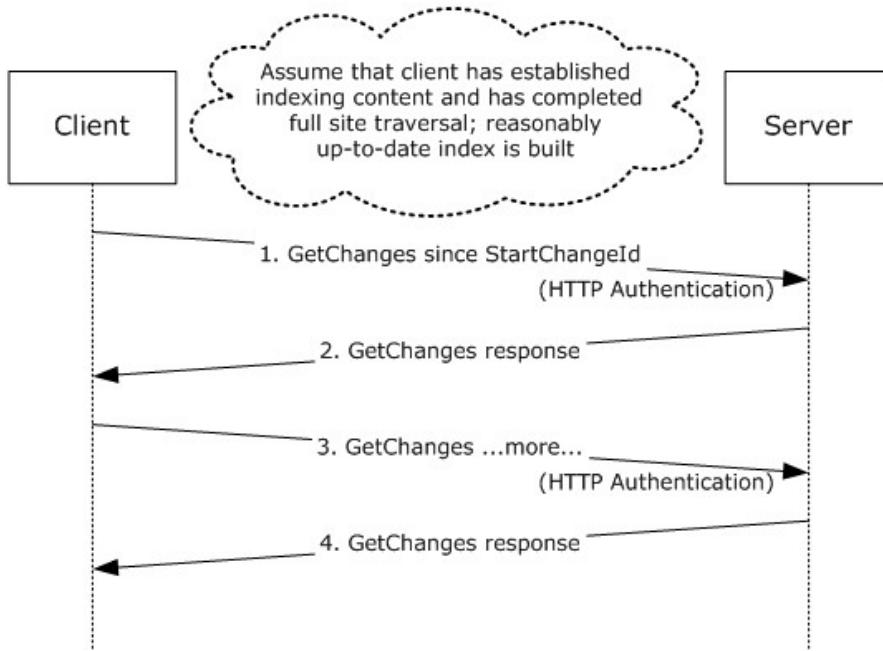
Permissions

The user or principal (1) that performs **GetContent** and **GetChanges** operations MUST have a special permission, called "Full Read".

Full Read permission is not normally granted to any user, not even to administrators. By default, only the service account used by the built-in indexing agent is granted Full Read permission.

Incremental indexing

The following diagram describes the recommended sequence of operations to perform incremental indexing.

Figure 6: Incremental indexing scenario

The following occurs:

1. To obtain incremental changes, the protocol client sends a **GetChanges** message, passing the following input parameters:

   ▪ Content database identifier obtained in step 2 of **Establishment of indexing context** scenario as the value of *contentDatabaseId* parameter.

   ▪ The **StartChangeId** token, obtained in the context establishment phase, as the value of *LastChangeId* parameter.

2. The protocol server sends a **GetChanges** response message that contains a change report. This change report consists of one or more change notifications. Each change notification specifies a particular change location and the changed element itself. This change location specifies a particular site (2), subsite, list, or list item. It also returns the **LastChangeId** token (meaning to get further changes, provide this **LastChangeId** value) and a flag called **moreChanges**, which indicates whether there are more changes to retrieve.

3. If the **moreChanges** flag is **true**, the protocol client sends another **GetChanges** message, passing the value of the **LastChangeId** token obtained in step 2.

4. The protocol server sends a **GetChanges** response message similar to that in step 2, returning second change report. If there are no more changes, the protocol server returns **false** in the **moreChanges** flag.

The protocol client stores the **LastChangeId** obtained in step 4 to be used in the next incremental indexing session by repeating this sequence.

Full details of the individual operations of this protocol are specified in sections 3.1.4.1 to 3.1.4.14.

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

This is the request message to enumerate the contents of the folder.

The **SOAP action** value of the message is defined as follows.

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

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

3.1.4.1.1.2 EnumerateFolderSoapOut

This is the response message to enumerate the contents of the folder.

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

```
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 definition of the **EnumerateFolder** element is as follows.

```
<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: Specifies the URL of the folder. The URL MUST be one of the following:

- Absolute URL of the folder in context site.

- 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 definition of the **EnumerateFolderResponse** element is as follows.

```
<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: Specifies the status of the operation. On successful completion of the operation the protocol server MUST set it to zero.

vUrls: Specifies URL information about immediate child folders, documents, and pages of a folder. **vUrls** is of **ArrayOf_sFPUrl** complex type, as defined in section 2.2.4.4.

If the protocol server encounters an error during 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 sever MUST return a SOAP fault with the following error message: "The Web application (2) at [Folder URL] could not be found. Verify that you have typed the URL correctly. If the URL should be serving existing content, the system administrator may need to add a new request URL mapping to the intended application." where [Folder URL] is the URL in the request.

3.1.4.1.3   Complex Types

None.

3.1.4.1.4   Simple Types

None.

3.1.4.1.5   Attributes

None.

3.1.4.1.6   Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 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

This is the request message to get all the attachments belonging to a list item.

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

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

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

3.1.4.2.1.2 GetAttachmentsSoapOut

This is the response message to get all the attachments belonging to a list item.

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

```
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 definition of the **GetAttachments** element is as follows.

```
<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: Specifies the name of the list or the GUID of the list. The GUID of the list SHOULD be enclosed in curly braces ({}).

strItemId: 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 definition of the **GetAttachmentsResponse** element is as follows.

```
<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: Specifies the status of the operation. On successful completion of the operation, the protocol server MUST set it to zero.

vAttachments: Specifies an array of URLs of all the attachments belonging to a list item. It is of type **ArrayOfString**, as defined in section 2.2.4.6. Each string in the array contains an absolute URL of an attachment.

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

Error code	Error string	Error condition
None	GUID should contain 32 digits with 4 dashes (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx).<19>	Specified **strListName** is ill-formatted.
None	Input string was not in a correct format.	Specified **strItemId** is empty or contains illegal values.
None	Value cannot be null.	Specified **strListName** element is not present.
None	Value does not fall within the expected range.<20>	The specified list item does not exist.
None	The system cannot find the file specified. (Exception from HRESULT: 0x80070002).<21>	The specified list does not exist.

3.1.4.2.3   Complex Types

None.

Release: February 11, 2013

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 GetChanges

The **GetChanges** operation is used by the protocol client to request the changes that occurred on a site (2) from one moment in time up to another moment in time. These moments in time are specified by opaque **ChangeId** tokens. **ChangeId** tokens are created and used by the protocol server to serialize the records of all atomic changes. **GetChanges** has **SOAPAction** header value:

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

GetChanges has the following schema.

```
<wsdl:operation name="GetChanges">
    <wsdl:input message="GetChangesSoapIn" />
    <wsdl:output message="GetChangesSoapOut" />
</wsdl:operation>
```

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

3.1.4.3.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.3.1.1 GetChangesSoapIn

GetChangesSoapIn is the request message of the **GetChanges** operation.

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

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

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

3.1.4.3.1.2 GetChangesSoapOut

GetChangesSoapOut is the response message to **GetChanges** operation.

The SOAP action value of the message is defined as:

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

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

3.1.4.3.2 Elements

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

3.1.4.3.2.1 GetChanges

The **GetChanges** element is the only element of **GetChangesSoapIn** request message, and has the following schema.

```
<s:element name="GetChanges">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="objectType"
              type="tns:ObjectType" />
      <s:element minOccurs="0" maxOccurs="1" name="contentDatabaseId"
              type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="LastChangeId" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="CurrentChangeId"
              type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="Timeout" type="s:int" />
    </s:sequence>
  </s:complexType>
</s:element>
```

objectType: The change tracking space to report about, either "ContentDatabase" or "SiteCollection". All other objectType values, as defined in section 2.2.5.3, MUST NOT be used. Note that "Site" in the context of this parameter actually means site collection.

contentDatabaseId: GUID of the content database, known in advance or obtained by GetContent request.

LastChangeId: A token specifying the starting point for the requested change report. Normally the protocol client obtains this value from the response to a previous **GetContent** or **GetChanges** operation.

CurrentChangeId: A token specifying the endpoint for the requested change report. If not empty, **CurrentChangeId** MUST be a valid token obtained from the response to a previous **GetChanges** operation. Normally, this element is empty; empty specifies that the protocol client requests all changes starting from the starting point up to the present time.

Timeout: A value that determines how many changes can be fetched in the current operation. This value MUST be greater than zero, and the protocol server MUST only fetch x% of total changes that are fetched by default, where *x* is (**Timeout** divided by 30000).

The protocol client MUST pass tokens that correspond to the change tracking space specified by the **objectType** and the target URL of the SOAP request.

3.1.4.3.2.2   GetChangesResponse

The **GetChangesResponse** element is the sole content of **GetChangesSoapOut** response message. This element has the following schema.

```
<s:element name="GetChangesResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetChangesResult" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="LastChangeId" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="CurrentChangeId" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="MoreChanges" type="s:boolean" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetChangesResult: MUST be a well-formed XML string containing exactly one change notification about the change tracking space defined in the **objectType** element (**ContentDatabase** or **Site**) of the **GetChange** request element. If the change tracking space is the content database, then the root element of the change notification is the **SPContentDatabase** element, as defined in section 3.1.4.3.2.3; if the change tracking space is the site (2), then the root element is the **SPSite** element, as defined in section 3.1.4.3.2.8. In both cases, the root element attributes either indicate that there are no more changes to report about (Change=Unchanged, ItemCount = 0), or, when ItemCount > 0, contain embedded change notifications of structural elements subordinated to the root.

The content of **GetChangesResult** is called the *change report* in this document. This change report consists of change notifications in the change tracking space requested. A change report contains zero or more atomic change notifications, which the protocol client can inspect and use. The protocol server returns as many atomic changes in the change report as it has been able to collect, bounded by the amount of data allowed by the timeout limit. The protocol server MUST always return a contiguous set of change notifications.

Change notifications have the following hierarchy:

```
ContentDatabase
   SiteCollection
      Site
         subsite
         List
            View
            List Item
         Folder
         File
```

All change notifications in the change report are conventionally prefixed with SP. For example, **SPContentDatabase**, **SPSite**, **SPWeb**, **SPFolder**, **SPFile**, **SPList**, **SPView**, or **SPListItem** as defined in section 3.1.4.3.2.

Normally, the protocol client traverses the **GetChangesResult** data, looking for change notifications expected at each level, for example, **SPListItem** inside of **SPList**. The way the protocol client processes each notification is an implementation detail of the protocol client. The change report does not indicate which field (2) of the list item has changed, which column is added to, or removed from, the view, or which fragment of an attachment file or a document has been modified. This analysis is left to the protocol client.

LastChangeId: A token specifying the actual endpoint for the returned change report. Normally, the protocol client saves this token and passes it as the *LastChangeId* input parameter to subsequent **GetChanges** requests.

CurrentChangeId: If the protocol client specified a blank **CurrentChangeId** in the request, then this is a token specifying the most recent change the protocol server is prepared to report about. Otherwise, if the protocol client specified a non-blank CurrentChangeId in the request, this is the token specifying the requested endpoint for the change report.

MoreChanges: A Boolean value that indicates whether the protocol server is aware of any more changes before the one marked by **CurrentChangeId** token. If the protocol server has successfully collected all change records for the requested increment within the Timeout limit, then the protocol server MUST specify **MoreChanges** = false. Also in this case the protocol server MUST set **LastChangeId** and **CurrentChangeId** to the same value. This tells the protocol client that there are no more changes marked before the **CurrentChangeId** token.

If the protocol server was unable to collect all change records within the timeout or other constraints, the protocol server MUST specify **MoreChanges** = true. This tells the protocol client that it SHOULD repeat the request using the suggested **LastChangeId** and **CurrentChangeId**.

3.1.4.3.2.3 SPContentDatabase

The **SPContentDatabase** element contains notifications of changes (if any) which have occurred in the change tracking space, and zero or more **SPSite** elements detailing the changes in a particular site collection serviced by this content database. This element has the following schema.

```
<s:element name="SPContentDatabase">
 <s:complexType>
  <s:sequence>
    <s:element name="ContentDatabase">
      <s:complexType>
        <s:sequence>
          <s:element name="Metadata" type="tns:ContentDatabaseMetadata" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element ref="tns:SPSite" minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
  <s:attribute name="ItemCount" type="s:int" use="required"/>
  <s:attribute name="Change" type="s:string" use="required"/>
 </s:complexType>
</s:element>
```

An example of a **SPContentDatabase** element follows.

```
<SPContentDatabase Change="Unchanged" ItemCount="7">
    <ContentDatabase>
        <Metadata
ChangeId="1;0;c20305a0-9249-4f91-b83c-1188c75e6cdd;633368042102070000;4057" ID="{c20305a0-
9249-4f91-b83c-1188c75e6cdd}"/>
    </ContentDatabase>
    <!-- zero or more SPSite elements -->

    <SPSite  >
    <!-- … details of site collection change notification -->
    </SPSite>
```

```
        <SPSite  >
        <!-- … details of site collection change notification -->
        </SPSite>

    </SPContentDatabase>
```

ItemCount: Specifies the number of change notifications included in this change report, see section 3.1.4.3.5.8.

Change: Specifies the change type of the content database, see section 3.1.4.3.5.1.

ContentDatabase.Metadata: Specifies the metadata of the content database, as defined in section 2.2.4.7.

SPSite: Specifies the change notification about one of the site collections serviced by this content database. There might be zero or more of **SPSite** elements in **SPContentDatabase**.

3.1.4.3.2.4 SPFile

The **SPFile** element is a change notification about the file (attachment), and has the following schema.

```
  <s:element name='SPFile'>
    <s:attribute  name='ItemCount' type='s:int' use='required'/>
    <s:attribute  name='Change' type='s:string' use='required'/>
    <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" use="optional" />
    <s:attribute name="ServerUrl" type="s:anyURI" use="optional" />
    <s:attribute name="ParentId" type="s:string"  use="optional" />
    <s:attribute name="InternalUrl" type="s:string" use="optional"/>
    <s:attribute name="Id" type="s:string"  use="optional" />
    <s:attribute name="DisplayUrl" type="s:anyUri"  use="optional" />
    <s:attribute name="CRC" type="s:long" use="optional"/>
    <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
    <s:attribute name="Url" type="s:string" use="optional" />
  </s:element>
```

Url: store-relative URL of the file (attachment). **Url** MUST NOT be URL-encoded. The protocol client SHOULD URL-encode [RFC2396] this URL to successfully apply the HTTP **GET** method. For example, replacing spaces with "%20".

ItemCount: See section 3.1.4.3.5.8. It MUST be always be zero, as files do not have child elements.

Change: Specifies the change type of this file, see section 3.1.4.3.5.1.<22>

UpdateSecurity: See section 3.1.4.3.5.11.

Id: See section 3.1.4.3.5.6.

ParentId: See section 3.1.4.3.5.9.

InternalUrl: See section 3.1.4.3.5.7.

DisplayUrl: See section 3.1.4.3.5.5.

ServerUrl: See section 3.1.4.3.5.10.

CRC: See section 3.1.4.3.5.4.

SequenceNumber: See section 3.1.4.3.5.13.

Note that attachments to the list items are included into special folders hidden inside of the list, for example, `Lists/Actors/Attachments/3/bio.doc`. Here, *3* means "attachments to the list item number 3 in the **Actors** list ", and *bio.doc* is the file name. The indexing service is supposed to read the file using the HTTP **GET** method and probably index its content. Attachment folders are not visible to **GetListCollection**, **GetList**, and **EnumerateFolder** operations.

3.1.4.3.2.5 SPFolder

The **SPFolder** element is a change notification about a folder, and can appear inside of a **SPWeb** element. **SPFolder** has the following schema.

```
<s:element name='SPFolder'>
  <s:attribute  name='ItemCount' type='s:int' use='required'/>
  <s:attribute  name='Change' type='s:string' use='required'/>
  <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" use="optional" />
  <s:attribute name="ServerUrl" type="s:anyURI" use="optional" />
  <s:attribute name="ParentId" type="s:string"  use="optional" />
  <s:attribute name="InternalUrl" type="s:string" use="optional"/>
  <s:attribute name="Id" type="s:string"  use="optional" />
  <s:attribute name="DisplayUrl" type="s:anyUri"  use="optional" />
  <s:attribute name="CRC" type="s:long" use="optional"/>
  <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
  <s:attribute name="Url" type="s:string" use="optional" />
</s:element>
```

Url: See section 3.1.4.3.5.12.

ItemCount: See section 3.1.4.3.5.8. It MUST always be zero, as a folder is the ultimate change unit.

Change: Specifies the change type of this folder, see section 3.1.4.3.5.1.

UpdateSecurity: See section 3.1.4.3.5.11.

Id: See section 3.1.4.3.5.6.

ParentId: See section 3.1.4.3.5.9.

InternalUrl: See section 3.1.4.3.5.7.

DisplayUrl: See section 3.1.4.3.5.5.

ServerUrl: See section 3.1.4.3.5.10.

CRC: See section 3.1.4.3.5.4.

SequenceNumber: See section 3.1.4.3.5.13.

Note that attachments to the list items are included in special hidden folders inside of the list itself; for example, `Lists/Actors/Attachments/3`. Here, *3* means "attachments to the list item #3 in the **Actors** list". When such a **SPFolder** change notification is present, the protocol client SHOULD

Release: February 11, 2013

examine the **SPFile** change notifications to read and possibly index file content. Attachment folders are not visible to **GetListCollection**, **GetList**, and **EnumerateFolder** operations.

3.1.4.3.2.6   SPList

The **SPList** element is a change notification that contains information about lists in the context of incremental indexing (see section [1.3](#)), and can appear inside of a **SPWeb** element. **SPList** has the following schema.

```
<s:element name='SPList'>
  <s:sequence >
      <s:element  ref='tns:List' minOccurs='0'/>
      <s:choice minOccurs='0' maxOccurs='unbounded'>
        <s:element  ref='tns:SPView'  minOccurs='0' maxOccurs='unbounded'/>
        <s:element  ref='tns:SPListItem' minOccurs='0' maxOccurs='unbounded'/>
      </s:choice>
  </s:sequence>
  <s:attribute  name='ItemCount' type='s:int' use='required'/>
  <s:attribute  name='Change' type='s:string' use='required'/>
  <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" />
  <s:attribute name="Id" type="s:string" />
  <s:attribute name="ParentId" type="s:string" />
  <s:attribute name="InternalUrl" type="s:string" />
  <s:attribute name="DisplayUrl" type="s:string" />
  <s:attribute name="CRC" type="s:int" />
  <s:attribute name="ServerUrl" type="s:string" />
  <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
</s:element>
```

ItemCount: See section [3.1.4.3.5.8](#). Indicates how many structural elements (for example, list items) contained in this list have been changed (added, deleted, modified).

Change: See section [3.1.4.3.5.1](#). Indicates whether this list itself has been changed.

Note that the combination of attributes `Change="Unchanged" ItemCount="2"` means that the list itself (its name and so on) remains the same, however two underlying elements have been changed.

UpdateSecurity: See section [3.1.4.3.5.11](#).

Id: See section [3.1.4.3.5.6](#).

ParentId: See section [3.1.4.3.5.9](#).

InternalUrl: See section [3.1.4.3.5.7](#).

DisplayUrl: See section [3.1.4.3.5.5](#).

ServerUrl: See section [3.1.4.3.5.10](#).

CRC: See section [3.1.4.3.5.4](#).

SequenceNumber: See section [3.1.4.3.5.13](#).

List: This element contains list metadata – general properties of the list, schema of the list and permissions granted to users and groups (2), as defined in section [2.2.4.13](#). Only the fields (2) that are not excluded from indexing MUST be shown in the list schema (see section [2.2.4.19](#)).

Note that **ListMetadata** contains property CRC, which is a protocol server-generated hash code reflecting fields which are excluded from indexing. If CRC is different than before, that means that the site (2) administration has changed crawling rules. In such case, the client needs to re-index the list using full traversal, because some fields, though unchanged, might have been made visible (or invisible) to indexing. Visible fields are enumerated in **List.Schema**.

SPView: Those elements (there might be any number of them) describe the view of the list (which columns are visible, in which order and so on). Lists can have more than one view, hence multiple **SPView** elements can be present. See section 3.1.4.3.2.10.

SPListItem: Those elements (any number of them) contain details about the list items having been changed. See section 3.1.4.3.2.7.

3.1.4.3.2.7   SPListItem

The **SPListItem** element is a change notification about a list item. It can appear inside of **SPList** element. **SPListItem** has the following schema.

```
<s:element name='SPListItem'>
  <s:sequence>
    <s:element  name='ListItem' >
      <s:complexType >
        <s:sequence >
          <s:element name='row' type='rs:row' />
          <s:element  name='permissions' type='tns:permissionsForACL' />
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
  <s:attribute name="Change" type="s:string"  use="required"/>
  <s:attribute name="ItemCount" type="s:int" use="required" />
  <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" use="optional" />
  <s:attribute name="ServerUrl" type="s:anyURI" use="optional" />
  <s:attribute name="ParentId" type="s:string"  use="optional" />
  <s:attribute name="InternalUrl" type="s:string" use="optional"/>
  <s:attribute name="Id" type="s:string"  use="optional" />
  <s:attribute name="DisplayUrl" type="s:anyUri"  use="optional" />
  <s:attribute name="CRC" type="s:unsignedInt" use="optional"/>
  <s:attribute name="Url" type="s:string" use="optional" />
  <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
</s:element>
```

ListItem.row: Single data row containing all fields (2) of a list item, in ADO XML Persistence format as defined in [MS-PRSTFR], but without the schema. See section 3.1.4.9 for the explanation of how to obtain the list schema.

ListItem.permissions: Permissions granted to this list item, see section 2.2.4.17.

ItemCount: See section 3.1.4.3.5.8. It MUST be zero, because the list item does not have structural components.

Change: See section 3.1.4.3.5.1. Indicates whether this list item itself has been changed.

UpdateSecurity: See section 3.1.4.3.5.11.

Id: See section 3.1.4.3.5.6.

ParentId: See section 3.1.4.3.5.9.

InternalUrl: See section 3.1.4.3.5.7.

DisplayUrl: See section 3.1.4.3.5.5.

ServerUrl: See section 3.1.4.3.5.10.

CRC: See section 3.1.4.3.5.4. The protocol server MUST set it to zero.

Url: See section 3.1.4.3.5.12.

SequenceNumber: See section 3.1.4.3.5.13.

Note that when a document in a document library has been edited and its content has been changed, it is not reflected in field values that the row with list item fields contains. Normally, an indexing agent reads the document file using the **ows_EncodedAbsUrl** field, to see whether the file has changed.

3.1.4.3.2.8 SPSite

The **SPSite** element is a change notification about certain site collections. **SPSite** has the following schema.

```
<s:element name="SPSite">
 <s:complexType>
  <s:sequence>
    <s:element name="Messages" >
      <s:complexType>
        <s:sequence>
          <s:element name="Message" type="tns:Message" minOccurs="0" maxOccurs="unbounded" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="Site" minOccurs="0" >
      <s:complexType>
        <s:sequence>
          <s:element name="Metadata" type="tns:SPSiteMetadata"/>
          <s:element name="Groups" type="tns:GroupMembership"/>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element ref="tns:SPWeb" minOccurs="0"  maxOccurs="unbounded"/>
  </s:sequence>
  <s:attribute name="ItemCount" type="s:int" use="required" />
  <s:attribute name="Change" type="s:string" use="required" />
  <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" use="optional" />
  <s:attribute name="Id" type="s:string" use="optional" />
  <s:attribute name="ParentId" type="s:string" use="optional" />
  <s:attribute name="InternalUrl" type="s:string" use="optional" />
  <s:attribute name="DisplayUrl" type="s:string" use="optional" />
  <s:attribute name="CRC" type="s:int" use="optional" />
  <s:attribute name="ServerUrl" type="s:string" use="optional" />
  <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
  <s:attribute name="PresentInFarm" type="tns:TrueFalseType" use="optional" />
 </s:complexType>
</s:element>
```

The following is an example of a **SPSite** element (certain attributes and elements are not shown for simplicity):

```
<SPSite Change="Unchanged" ItemCount="7" >
    <Messages>
    <!-- messages about changes that could not be processed -->
        <Message />
        <Message />
    </Messages>
    <Site>
        <Metadata
 URL="http://johnD"
 ID="{e60888c1-7206-462c-8b87-dbc2ebdc97d9}"
 LastModified="2008-01-24 20:43:30Z"
 PortalURL=""
 UserProfileGUID=""
 RootWebId="{8768351c-2dbc-4b3d-9fce-deb3b685a992}"
 ChangeId="1;0;c20305a0-9249-4f91-b83c-1188c75e6cdd;633368042102070000;4057" />
        <Groups/>
    </Site>
    <!-- zero or more SPWeb change notifications -->
    <SPWeb >
        <!-- details of Web change notification -->
    </SPWeb>
    <SPWeb >
        <!-- details of Web change notification -->
    </SPWeb>
</SPSite>
```

Messages: Specifies details about changes that could not be properly processed as part of this **SPSite** change notification.

Message: Specifies detail about a single change that could not be properly processed.

Site.Metadata: Specifies details about this site collection, as defined in section 2.2.4.21.

Site.Groups: Specifies information about groups (2) and users in this site collection membership, see section 2.2.4.10.

SPWeb: Specifies change notification about a subsite, see section 3.1.4.3.2.11. **SPSite** change notification can contain zero or more **SPWeb** change notifications about its direct and indirect subsites.

ItemCount: Specifies the number of change notifications (about site collection child elements) included into this change report. See section 3.1.4.3.5.8.

Change: Specifies the change type of this site collection. See section 3.1.4.3.5.1.

UpdateSecurity: See section 3.1.4.3.5.11.

Id: See section 3.1.4.3.5.6.

ParentId: See section 3.1.4.3.5.9.

InternalUrl: See section 3.1.4.3.5.7.

DisplayUrl: See section 3.1.4.3.5.5.

ServerUrl: See section 3.1.4.3.5.10.

CRC: See section 3.1.4.3.5.4. If being set, the protocol server MUST set it to zero.

SequenceNumber: See section 3.1.4.3.5.13.

PresentInFarm: Specifies whether the site collection still exists in farm. SHOULD<23> be present if the **Change** type is Delete.

3.1.4.3.2.9   Message

The **Message** element contains detailed information about a change as part of the **SPSite** change notification.<24> **Message** has the following schema.<25>

```
<s:complexType name="Message">
  <s:sequence>
    <s:element name="Exception" minOccurs="1" maxOccurs="1" type="s:string">
    </s:element>
    <s:element name="ChangeInfo" minOccurs="1" maxOccurs="1">
     <s:complexType>
      <s:attribute name="SiteId" type="s:string" use="required" />
      <s:attribute name="WebId" type="s:string" use="optional" />
      <s:attribute name="ListId" type="s:string" use="optional" />
      <s:attribute name="AlertId" type="s:string" use="optional" />
      <s:attribute name="Id" type="s:string" use="optional" />
      <s:attribute name="FieldId" type="s:string" use="optional" />
      <s:attribute name="Url" type="s:string" use="optional" />
      <s:attribute name="DocId" type="s:string" use="optional" />
      <s:attribute name="UserId" type="s:string" use="optional" />
      <s:attribute name="Activate" type="s:boolean" use="optional" />
      <s:attribute name="IsSiteAdminChange" type="s:boolean" use="optional" />
     </s:complexType>
    </s:element>
  </s:sequence>
  <s:attribute name="ChangeToken" type="s:string" use="required" />
  <s:attribute name="ChangeObject" type="tns:ChangeObject" use="required" />
  <s:attribute name="ChangeType" type="tns:ChangeType" use="required" />
</s:complexType>
```

ChangeToken: Specifies the token of the change.

ChangeObject: Specifies the type identifier of the change. See section 3.1.4.3.5.2.

ChangeType: Specifies the original type of the change before processing. See section 3.1.4.3.5.3.

Exception: Specifies details about the error that occurred while processing the change.

ChangeInfo: Specifies a core set of information about the change.

ChangeInfo.SiteId: Specifies the GUID of the site collection in which the changed item is located.

ChangeInfo.WebId: Specifies the GUID of the site (2) in which the changed item is located. If the changed item is a site collection, this attribute MUST be omitted.

ChangeInfo.ListId: Specifies the GUID of the list (1) in which the changed item is located. If the changed item is a site (2) or site collection, this attribute MUST be omitted.

ChangeInfo.AlertId: If the changed item is an **alert (2)**, specifies the GUID for the alert (2). If the changed item is not an alert (2), this attribute MUST be omitted.

ChangeInfo.Id: Specifies the GUID of the changed item.

ChangeInfo.FieldId: Specifies the GUID of the field (2) that was changed. If the changed item is not a field (2), this attribute MUST be omitted.

ChangeInfo.Url: Specifies the URL of the changed item.

ChangeInfo.DocId: Specifies the GUID of the document representing the changed item.

ChangeInfo.UserId: Specifies the GUID of the user whose data was changed. If the changed item is not a user, this attribute MUST be omitted.

ChangeInfo.Activate: Specifies if the security change involved giving write permission to a user who previously only had read permission. If the changed item is not a security change, this attribute MUST be omitted.

ChangeInfo.IsSiteAdminChange: Specifies if the security change involved change the current site collection administrator. If the changed item is not a security change, this attribute MUST be omitted.

3.1.4.3.2.10   SPView

The **SPView** element contains detailed information about the view, and can appear inside of **SPList** element. **SPView** has the following schema.

```
<s:element name="SPView">
 <s:complexType>
  <s:attribute name="ItemCount" type="s:int" use="required"/>
  <s:attribute name="Change" type="s:string" use="required"/>
  <s:attribute name="Id" type="s:string" use="optional"/>
  <s:attribute name="ServerUrl" type="s:anyURI" use="optional"/>
  <s:attribute name="DisplayUrl" type="s:string" use="optional" />
  <s:attribute name="ParentId" type="s:string" use="optional" />
  <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" use="optional"/>
  <s:attribute name="InternalUrl" type="tns:InternalUrl" use="optional"/>
  <s:attribute name="CRC" type="s:int" use="optional" />
  <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
 </s:complexType>
</s:element>
```

ItemCount: The protocol server MUST set it to zero, as view is the ultimate change unit.

Change: Indicates change type of this view. See section 3.1.4.3.5.1.

UpdateSecurity: See section 3.1.4.3.5.11.

Id: See section 3.1.4.3.5.6.

ParentId: See section 3.1.4.3.5.9.

InternalUrl: See section 3.1.4.3.5.7.

DisplayUrl: See section 3.1.4.3.5.5.

ServerUrl: See section 3.1.4.3.5.10.

CRC: See section 3.1.4.3.5.4. The protocol server MUST set it to zero.

SequenceNumber: See section 3.1.4.3.5.13.

3.1.4.3.2.11   SPWeb

The **SPWeb** element is a change notification about one of the sites (2) contained in a certain site collections (hence **SPWeb** elements appear inside a **SPSite** element). **SPWeb** has the following schema.

```
<s:element name="SPWeb">
<s:complexType>
 <s:sequence>
   <s:element name="Web" minoccurs="0">
     <s:complexType>
       <s:sequence>
         <s:element name="Metadata" minOccurs="0" type="tns:WebMetadata"/>
         <s:element name="Users" minOccurs="0" type="tns:Users"/>
         <s:element name="ACL" minOccurs="0" type="tns:ACL"/>
         <s:element name="Webs" minOccurs="0" type="tns:Webs"/>
         <s:element name="Lists" minOccurs="0" type="tns:Lists"/>
       </s:sequence>
     </s:complexType>
   </s:element>
   <s:choice maxOccurs="unbounded">
     <s:element ref="tns:SPFolder"/>
     <s:element ref="tns:SPList"/>
     <s:element ref="tns:SPFile"/>
   </s:choice>
 </s:sequence>
 <s:attribute name="ItemCount" type="s:int" use="required"/>
 <s:attribute name="Change" type="s:string" use="required"/>
 <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" use="optional"/>
 <s:attribute name="Id" type="s:string" use="optional"/>
 <s:attribute name="ParentId" type="s:string" use="optional"/>
 <s:attribute name="InternalUrl" type="s:string" use="optional"/>
 <s:attribute name="DisplayUrl" type="s:string" use="optional"/>
 <s:attribute name="CRC" type="s:int" use="optional"/>
 <s:attribute name="ServerUrl" type="s:string" use="optional"/>
 <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
 </s:complexType>
</s:element>
```

The following example shows the **SPWeb** element (some attributes and elements are not shown for the sake of clarity):

```
<SPWeb Change="Unchanged" ItemCount="7">
    <Web>
        <Metadata
        URL="http://johnD"
        LastModified="2008-01-24 20:43:30Z"
         ID="{8768351c-2dbc-4b3d-9fce-deb3b685a992}"
        Title="Home"
        Description=""
        Author="John Doe"
```

```
                Language="1033"
                CRC="2982074663"
                NoIndex="False"
                DefaultHomePage="Pages/Default.aspx"
                ExternalSecurity="False"
                ScopeID="{54219cfc-64dd-40e3-8cf7-2631e9a941e9}"
                AllowAnonymousAccess="False"
                AnonymousViewListItems="False"
                AnonymousPermMask="0" />
            <Users/>
            <ACL/>
            <Webs>
            </Webs>
            <Lists>
            </Lists>
        </Web>
        <SPFolder/>
        <SPList/>
        <SPFile/>
        <SPList/>
    </SPWeb>
```

Web: MUST be present except if **Change** equals "Delete". If **Change** equals "Delete", **Web** MUST be omitted.

Web.Metadata: Specifies details about the Web as defined in section 2.2.4.27.

Web.Users: Specifies the list of the users of the site (2), as defined in site membership, see section 2.2.4.24.

Web.ACL: Specifies the permissions granted to users and groups (2), as defined in site membership. See section 2.2.4.3.

Web.Webs: The protocol server MUST set it to empty.

Web.Lists: The protocol server MUST set it to empty.

SPFolder: Change notification about a certain folder included in the change report, see section 3.1.4.3.2.5.

SPList: Change notification about a certain list included in the change report, see section 3.1.4.3.2.6.

SPFile: Change notification about a certain folder included in the change report, see section 3.1.4.3.2.4.

There is no fixed order for **SPFile**, **SPFolder**, and **SPList** elements inside a **SPWeb**; they can appear in any order.

Change: Specifies the change type of this site (2). See section 3.1.4.3.5.1.

ItemCount: Specifies the number of change notifications (following this site (2)) included in this change report. See section 3.1.4.3.5.8.

UpdateSecurity: See section 3.1.4.3.5.11.

Id: See section 3.1.4.3.5.6.

ParentId: See section 3.1.4.3.5.9.

InternalUrl: See section 3.1.4.3.5.7.

DisplayUrl: See section 3.1.4.3.5.5.

ServerUrl: See section 3.1.4.3.5.10.

CRC: See section 3.1.4.3.5.4. This attribute of **SPWeb** is the hash code of the Web indexing properties configured by site collection administration: whether to index this Web, and whether to index ASPX pages. A change of CRC (as compared with previously known) indicates that indexing properties have changed.

SequenceNumber: See section 3.1.4.3.5.13.

3.1.4.3.3 Complex Types

None.

3.1.4.3.4 Simple Types

None

3.1.4.3.5 Attributes

All change notifications have two mandatory common attributes–**Change** and **ItemCount**–and a number of optional, common attributes. The optional attributes include the following:

- **CRC**

- **DisplayUrl**

- **Id**

- **InternalUrl**

- **ParentId**

- **ServerUrl**

- **UpdateSecurity**

- **Url**

- **SequenceNumber**<26>

This section specifies both the mandatory attributes and these optional attributes.

The protocol server MUST include both mandatory attributes in every change notification. The protocol server includes optional attributes if the structural element that is the subject of the change notification has changed.

3.1.4.3.5.1 Change

The **Change** attribute defines what kind of change this notification is about. It is of type **string** and MUST contain one of the values described in the following table.

Value	Description
Unchanged	The structural element that is the subject of the change notification has not changed, but components of that element might have changed. See the **ItemCount** attribute value.
UpdateShallow	The structural element that is the subject of the change notification has changed, and its components might have changed as well. See the **ItemCount** attribute value.
UpdateDeep	The structural element that is the subject of the change notification has been restored from backup, including all of its components. The restoration process does not track all atomic changes. Therefore, the protocol client is recommended to re-index the element using full traversal, ignoring the value of the **ItemCount** attribute.<27>
Add	Element has been added.
Delete	Element has been deleted.
UpdateSecurity	Permissions to access this element have been changed. The protocol client is recommended to re-index the element and its child elements, because those permissions might affect search results visibility.
RenameMove	Element has been renamed or moved to another container.
UpdateSecurityScope<28>	Specifies that the scope of permissions on this item has changed to a different scope. The new scope can have been previously known so security information can already be cached on the protocol client.

If the structural element has been deleted (`Change="Delete"`), then the only information that the notification conveys about the deleted element is its identification, which is always a GUID. This applies to the following elements: **SPSite**, **SPWeb**, **SPFolder**, **SPFile**, **SPView**, and **SPList**. For the **SPListItem**, this **ID** actually denotes the **ows_UniqueId** attribute, not the sequential identification usually used to identify list items.

3.1.4.3.5.2   Change Object

The **Change Object** attribute defines what type of object the change was performed upon. It is of type **string** and MUST contain one of the values described in the following table.

Name	Description
Microsoft.SharePoint.SPChangeItem	A change on a list item.
Microsoft.SharePoint.SPChangeSite	A change on a site collection.
Microsoft.SharePoint.SPChangeWeb	A change on a site (2).
Microsoft.SharePoint.SPChangeList	A change on a list (1).
Microsoft.SharePoint.SPChangeFile	A change on a document.
Microsoft.SharePoint.SPChangeFolder	A change on a folder.
Microsoft.SharePoint.SPChangeAlert	A change on an alert (2).
Microsoft.SharePoint.SPChangeUser	A change on a user.

Name	Description
Microsoft.SharePoint.SPChangeGroup	A change on a group (2).
Microsoft.SharePoint.SPChangeContentType	A change on a content type.
Microsoft.SharePoint.SPChangeField	A change on a field (2).
Microsoft.SharePoint.SPChangeSecurityPolicy	A change on a security setting.
Microsoft.SharePoint.SPChangeView	A change on a view.

3.1.4.3.5.3 Change Type

The **Change Type** attribute defines what kind of type this change is. It is of type **string** and MUST contain one of the values described in the following table.

Name	Description
NoChange	Element has not been changed.
Add	Element has been added.
Update	Element has been updated.
Delete	Element has been deleted.
Rename	Element's leaf name has been renamed.
MoveAway	Element was removed from a site (2).
MoveInto	Element was moved into a site (2).
Restore	Element was restored from backup.
RoleAdd	A **role definition** was added.
RoleDelete	A role definition was deleted.
RoleUpdate	A role definition was updated.
AssignmentAdd	A role was added to an element.
AssignmentDelete	A role as removed from an element.
MemberAdd	A user was added to a role.
MemberDelete	A user was removed from a role.
SystemUpdate	Element was updated by the system.
Navigation	Element's navigation was updated.

3.1.4.3.5.4 CRC

The **CRC** attribute is of type **integer** and contains a checksum of various properties of a structural element. The indexing service SHOULD keep track of this **CRC** value and once it has changed (compared to a previously value) the indexing service SHOULD re-scan the element and all its structural components.

3.1.4.3.5.5 DisplayUrl

The **DisplayUrl** attribute contains the server-relative URL of a Web page that displays the structural element to users.

3.1.4.3.5.6 Id

The **Id** attribute is of type GUID and uniquely identifies a structural element.

3.1.4.3.5.7 InternalUrl

The **InternalUrl** attribute is a composite string denoting the path to some structural element of the site (2) that appears in change notification (see section 3.1.4.3). The structural element can be one of the following: site collection, site (2), list, folder, or list item. The value of **InternalUrl** is not meant to be used as a URL; any similarity in the name and syntax is purely coincidental. The syntax of the string is specified as follows in Augmented BNF notation [RFC5234] (assuming that the term GUID is implicitly defined):

```
SPACE    =   %x20
ALPHA  =  %x41-7A
DIGIT  =  %x30-39
INTEGER = *DIGIT
UNDERSCORE  =  %x5F ;  _
PUNCTUATION = ( "."   /   "," / ";" /  "!" / "-" / "@"
                                 /   "(" / ")"  / "[" / "]"
                              / ";" )  ; characters  ~ " # % & * : < > ? / \ { | }
are forbidden in names!
NAME  =  (ALPHA / UNDERSCORE)   *[ SPACE / ALPHA  /  UNDERSCORE
        / DIGIT / PUNCTUATION]
        ; NAME cannot begin or end with dot "." , or contain consecutive dots ".."
PATH  = [ NAME / NAME '/' PATH ]
GUIDEMBRACED  = '{' GUID '}'
INTERNALURL  = '/siteurl=' PATH  '/siteid=' GUIDEMBRACED
       [ '/weburl='  PATH  '/webid=' GUIDEMBRACED
       [ ( '/listid =' GUIDEMBRACED
          ( [ '/folderurl='  PATH
          [ '/itemid='  INTEGER ] ] / '/viewid=' GUIDEMBRACED )
          / '/fpfolder='  PATH [ '/fpitemid='  NAME ] )
       ] ]
```

For folder and file changes that are not in a document library, the **InternalUrl** contains **fpfolder** and **fpitem** components. For document library items the **InternalUrl** contains **listid**, **folderurl**, and **itemid** components.

For example, the **InternalUrl**:

```
/siteurl=/siteid={9bd2c245-41d0-4440-a6ad-1f526dcbe116}/weburl=Subsite/webid={6213f860-baea-
4167-ac5f-c5830b52f015}/listid={a04db214-c3c6-45d3-a601-
dd1e0e14816a}/folderurl=Subfolder/itemid=5
```

targets list item with **ID** set to 5, and has the following components:

siteurl: Specifies the URL of the changed site collection responding to this **GetChanges** operation. The URL is a store-relative URL to the server URL; an empty value means the site collection URL is the same as the server URL. For example, if the site collection URL is:

http://servername/sites/site01, then the **siteurl** will be "sites/site01"; if the site collection URL is: http://servername, the **siteurl** will be empty.

siteid: Specifies the GUID of the site (2).

weburl: Specifies the URL of the subsite relative to the site collection URL (included for Webs and child items).

webid: Specifies the GUID of that subsite.

listid: Specifies the GUID of the list (included for lists and child items).

folderurl: Specifies the path to the folder relative to the list itself (an empty value means that the item is contained in the list itself, not in a sub folder).

itemid: Specifies the identifier (integer) of the list item inside of its list (included for list items only).

Another example would be as follows.

```
/siteurl=/siteid={fab5349c-eb94-43bd-9792-917c44b75040}/weburl=/webid={45ff402e-c3be-4612-
98d6-915b569b742a}/fpfolder=Subfolder/fpitemid=default.aspx
```

This targets list item with **ID** set to 5, and has the following components:

siteurl: Specifies the store-relative URL of the structural element responding to this **GetChanges** operation. An empty value means "this site collection itself".

siteid: Specifies the GUID of the site (2).

weburl: Specifies the URL of the subsite relative to the site collection URL (included for Webs and child items).

webid: Specifies the GUID of that subsite.

fpfolder: Specifies the path to the folder relative to the subsite (for folders and files that are not in a document library).

fpitemid: Specifies the file name (included for files only).

viewid: A non-empty GUID that identifies the **View**.

3.1.4.3.5.8 ItemCount

The **ItemCount** attribute is of type **s:int** and indicates whether changes to the child elements of this structural element are included into this change report (and therefore SHOULD be processed by the indexing agent). For example, if **SPList** change notification has ItemCount=7, that means that seven **SPListItem** change notifications are contained in that **SPList** change notification.

Note that ultimate change notifications (**SPFolder**, **SPFile**, **SPView**, **SPListItem**) have no child objects and hence always have ItemCount = 0.

3.1.4.3.5.9 ParentId

The **ParentId** attribute is a GUID, uniquely identifying the parent of the structural element.

Release: February 11, 2013

3.1.4.3.5.10   ServerUrl

The **ServerUrl** attribute contains the URL of the Web application (2) that the structural element belongs to.

3.1.4.3.5.11   UpdateSecurity

The **UpdateSecurity** attribute is of type **TrueFalseType**, as defined in section 2.2.5.4. This value SHOULD<29> be "True" only if permissions to access this structural element have been changed.

3.1.4.3.5.12   Url

The **Url** attribute contains implementation-specific information about the structural element. It is not to be interpreted or used as URL. Its name is purely coincidental.

3.1.4.3.5.13   SequenceNumber

The **SequenceNumber** attribute defines the sequence number for the change.<30>

3.1.4.3.6   Groups

None.

3.1.4.3.7   Attribute Groups

None.

3.1.4.4   GetChangesEx

The **GetChangesEx** operation is used by the protocol client to request changes, similar to **GetChanges**. **GetChangesEx** provides additional functionality but follows the same message sequence events specified in section 3.1.4. **GetChangesEx** has the following **SOAPAction** header value:

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

GetChangesEx has the following schema:

```
<wsdl:operation name="GetChangesEx">
    <wsdl:input message="GetChangesExSoapIn" />
    <wsdl:output message="GetChangesExSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetChangesExSoapIn** request message and the protocol server responds with a **GetChangesExSoapOut** response message.

3.1.4.4.1   Messages

The following WSDL message definitions are specific to this operation.

3.1.4.4.1.1   GetChangesExSoapIn

GetChangesExSoapIn is the request message of the **GetChangesEx** operation.

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

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

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

3.1.4.4.1.2 GetChangesExSoapOut

GetChangesExSoapOut is the response message to **GetChangesEx** operation.

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

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

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

3.1.4.4.2 Elements

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

3.1.4.4.2.1 GetChangesEx

The **GetChangesEx** element is an element of the **GetChangesExSoapIn** request message, and has the following schema:

```
<s:element name="GetChangesEx">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="version" type="s:int" />
      <s:element minOccurs="1" maxOccurs="1" name="xmlInput" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

version: Specifies an integer version number to identify what version of the protocol the protocol client understand. This value MUST be set to 2.

xmlInput: Specifies a string specifying a **GetChangesExRequest** element.

3.1.4.4.2.2 GetChangesExRequest

The **GetChangesExRequest** element is an element of the **GetChangesSoapIn** request message, and has the following schema:

```
<s:element name="GetChangesExRequest">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ObjectType" type="tns:ObjectType" />
      <s:element minOccurs="1" maxOccurs="1" name="ContentDatabaseId" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="StartChangeId" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="EndChangeId" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="IgnoreSecurityIfInherit" type="s:boolean"
/>
```

```
            <s:element minOccurs="0" maxOccurs="1" name="GetMetadata" type="s:boolean" />
            <s:element minOccurs="0" maxOccurs="1" name="RequestLoad" type="s:int" />
        </s:sequence>
    </s:complexType>
</s:element>
```

ObjectType: Specifies the change tracking space to report about, either "ContentDatabase" or "SiteCollection". All other **ObjectType** values, as defined in section 2.2.5.3 MUST NOT be used. Note that "Site" in the context of this parameter actually means site collection.

ContentDatabaseId: Specifies the GUID of the content database, known in advance or obtained by **GetContent** request.

StartChangeId: Specifies a token specifying the starting point for the requested change report. Normally the protocol client obtains this value from the response to a previous **GetContent** or **GetChangesEx** operation.

EndChangeId: Specifies a token specifying the endpoint for the requested change report. If not empty, **EndChangeId** MUST be a valid token obtained from the response to a previous **GetChangesEx** operation. If this element is empty, the protocol server MUST return all changes starting from the starting point up to the present time.

IgnoreSecurityIfInherit: A Boolean value specifying whether permission information for objects in the change report is to be returned, if they inherit permissions from parent objects. If this value is not specified, the protocol server MUST return permission information for all objects.

GetMetadata: A Boolean value specifying whether basic information for each object in the change report is to be returned, along with the change information. If this value is not specified, the protocol server MUST return the metadata for each object in the change report.

RequestLoad: Specifies a value that determines how many changes can be fetched in the current operation, where the value represents the percentage of total changes that can be processed in the request. If this value is not specified or is 100, the protocol server MUST return all changes that can be processed in a single request. If this value is 0, the protocol server MUST return no changes. If this value is x, the protocol server MUST return x% of the total changes that can be processed in the request. This value MUST NOT be less than zero or greater than 100.

The protocol client MUST pass tokens that correspond to the change tracking space specified by the **objectType** and the target URL of the SOAP request.

3.1.4.4.2.3   GetChangesExResponse

The **GetChangesExResponse** element is the sole content of the **GetChangesExSoapOut** response message.

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

GetChangesExResult contains a **GetChangesResult** element, described in section 3.1.4.3.2.2, except for the differences specified in section 3.1.4.4.2.3.6. **GetChangesExResponse** follows a

similar schema to **GetChangesResponse** except for the differences specified in the following sections.

3.1.4.4.2.3.1   SPContentDatabase

The **SPContentDatabase** element is similar to [3.1.4.3.2.3](#), except if **GetChangesExRequest.GetMetadata** is **false**, the **SPContentDatabase.ContentDatabase** element MUST be omitted.

3.1.4.4.2.3.2   SPList

The **SPList** element is similar to [3.1.4.3.2.6](#). If **GetChangesExRequest.IgnoreSecurityIfInherit** is **true** and the list inherits its permissions from its parent object, the **SPList.List.ACL** element MUST be omitted. In addition, if **GetChangesExRequest.GetMetadata** is **false**, the entire **SPList.List** element MUST be omitted.

3.1.4.4.2.3.3   SPListItem

The **SPListItem** element is similar to [3.1.4.3.2.7](#), except if **GetChangesExRequest.IgnoreSecurityIfInherit** is **true** and the list item inherits its permissions from its parent object, the **SPListItem.ListItem.permissions** element MUST be omitted.

3.1.4.4.2.3.4   SPSite

The **SPSite** element is similar to [3.1.4.3.2.8](#). If **GetChangesExRequest.IgnoreSecurityIfInherit** is **true** and the site collection inherits its permissions from its parent object, the **SPSite.Site.Groups** element MUST be omitted. In addition, if **GetChangesExRequest.GetMetadata** is **false**, the entire **SPSite.Site** element MUST be omitted.

3.1.4.4.2.3.5   SPWeb

The **SPWeb** element is similar to [3.1.4.3.2.11](#). If **GetChangesExRequest.IgnoreSecurityIfInherit** is **true** and the site (2) inherits its permissions from its parent object, the **SPWeb.Web.ACL** and **SPWeb.Web.Groups** elements MUST be omitted. In addition, if **GetChangesExRequest.GetMetadata** is **false**, the entire **SPWeb.Web** element MUST be omitted.

3.1.4.4.2.3.6   StartChangeId and EndChangeId

The **LastChangeId** and **CurrentChangeId** element in **GetChangesResponse** have been renamed to **StartChangeId** and **EndChangeId**, respectively. Also, the **StartChangeId**, **EndChangeId**, and **MoreChanges** elements have become sub-elements of **GetChangesResult**, which is a child of **GetChangesExResult**.[<31>](#)

3.1.4.4.3   Complex Types

None.

3.1.4.4.4   Simple Types

None.

3.1.4.4.5   Attributes

The attributes that are different for **GetChangesEx** compared to **GetChanges** are specified in the following sections.

3.1.4.4.5.1   Change

The **Change** attribute is similar to 3.1.4.3.5.1.<32>

3.1.4.4.5.2   Change Type

The **Change Type** attribute is similar to 3.1.4.3.5.3, but also includes the additional types described in the following table.

Name	Description
ScopeAdd	Element does not inherit permissions from parent.
ScopeDelete	Element now inherits permissions from parent.

3.1.4.4.5.3   UpdateSecurityScope

The **UpdateSecurityScope** attribute is similar to 3.1.4.3.5.11. It is of type **TrueFalseType**, as defined in section 2.2.5.4, and indicates whether the scope of permissions on this element has changed to a different scope.

3.1.4.4.6   Groups

None.

3.1.4.4.7   Attribute Groups

None.

3.1.4.5   GetContent

The **GetContent** operation is used to get the following information:

- Basic change information about the following:

  ▪Content database.

  ▪Site collection.

- Detailed information about the following:

  ▪Web application (2).

  ▪Site collection.

  ▪site (2).

  ▪List.

  ▪List item.

▪Folder.

▪Attachment.

The value of the *objectType* parameter controls what type of object this operation gets information about.

```
<wsdl:operation name="GetContent">
  <wsdl:input message="GetContentSoapIn" />
  <wsdl:output message="GetContentSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetContentSoapIn** request message and the protocol server responds with a **GetContentSoapOut** response message. The data returned by the protocol server depends on the type of object specified by the input parameter *objectType*, as follows:

1. If the value of the *objectType* input parameter is "VirtualServer", the following apply:

   ▪If the value of the *retrieveChildItems* input parameter is **false**, the protocol server MUST only return general metadata.

   ▪If the value of the *retrieveChildItems* input parameter is **true**, the protocol server MUST return general metadata, information about the content databases, and permission information about the Web application (2).

   ▪The protocol server MUST ignore the rest of the input parameters. If the user has site (2) administrator or site (2) auditor permissions then the protocol server MUST return all the information as an **XML node** whose schema is defined in section 3.1.4.5.2.9.

2. If the value of the *objectType* input parameter is "ContentDatabase", the following apply:

   ▪If the value of the *retrieveChildItems* input parameter is **true**, the protocol server MUST return change information about the content database and information about all top-level sites (2) served by the content database.

   ▪If the value of the *retrieveChildItems* input parameter is **false**, the protocol server MUST return change information about the content database.

   ▪In all cases, the protocol client MUST specify the GUID of the content database in the *objectId* input parameter. This GUID SHOULD be enclosed in curly braces ({}).

   ▪The protocol server MUST ignore the rest of the input parameters. If the user has site (2) administrator or site (2) auditor permissions then the protocol server MUST return the information as an XML node whose schema is defined in section 3.1.4.5.2.1.

3. If the value of the *objectType* input parameter is "SiteCollection", the following apply:

   ▪If the value of the *retrieveChildItems* input parameter is **true**, the protocol server MUST return change information, metadata, user and group information, and information about the root site of the site collection containing the context site.

   ▪If the value of the *retrieveChildItems* input parameter is **false**, the protocol server MUST return change information, metadata, user and group information about the site collection containing the context site.

   ▪The protocol server MUST ignore the rest of the input parameters. The information MUST be returned as an XML node whose schema is defined in section 3.1.4.5.2.8.

4. If the value of the *objectType* input parameter is "Site", the following apply:

- If the value of the *retrieveChildItems* input parameter is **true**, the protocol server MUST return metadata, user information, permission information, information about all the subsites, lists and folders of the context site.

- If the value of the *retrieveChildItems* input parameter is **false**, the protocol server MUST only return metadata, user information and permission information of the context site. An empty set of information about subsites and list MUST also be returned.

- The protocol server MUST ignore the rest of the input parameters. The information MUST be returned as an XML node whose schema is specified in section 3.1.4.5.2.10.

5. If the value of the *objectType* input parameter is "List", the protocol server MUST return metadata, permission information, view, and schema information about the specified list in the context site. The protocol client MUST specify the name of the list or GUID of the list in the *objectId* input parameter. This GUID SHOULD be enclosed in curly braces ({}). The protocol server MUST ignore the rest of the input parameter. The information MUST be returned as an XML node whose schema is defined in section 3.1.4.5.2.7. The list schema specified in this information MUST only contain fields which are not excluded from indexing.

6. If the value of the *objectType* input parameter is "ListItem", the following apply:

- If the value of the *securityOnly* input parameter is **true**, the protocol server MUST return the following:

  - Permission information of the list that contains the list item.

  - Subset of metadata information about the list item. This information SHOULD include the following attributes: **ows_Modified**, **ows_ServerRedirected, ows_UniqueId**, **ows_scopeID**, **ows_FileRef, ows_ID** and **ows_FSObjType**.

- If the value of the *securityOnly* input parameter is **false**, the protocol server MUST return the following:

  - Permission information about the list that contains the list item.

  - All the of metadata information about the list item. This information MUST include the following attributes: **ows_Modified**, **ows_ServerRedirected, ows_UniqueId**, **ows_scopeID**, **ows_FileRef, ows_ID** and **ows_FSObjType**.

- In all cases, the following apply:

  - The protocol client MUST specify the name or GUID of the list in the *objectId* input parameter. The GUID SHOULD be enclosed in curly braces ({}).

  - The protocol client MUST specify the identification of the list item in the *itemId* input parameter.

  - The protocol client MUST specify the folder name in the *folderUrl* input parameter. If the *folderUrl* is empty, the protocol server MUST consider *folderUrl* as the root folder of the list identified by *objectId*.

  - The protocol server MUST ignore the rest of the input parameters. The information MUST be returned as an XML node whose schema is defined in section 3.1.4.5.2.5.

7. If the value of the *objectType* input parameter is "Folder", the protocol server MUST return information about the folder identified by the input parameters *objectId* and *folderUrl*. The information returned depends on the input parameter *objectId*:

▪If the name or the GUID of the list is specified in the *objectId* input parameter, then:

   ▪ If the value of the *securityOnly* input parameter is **true**, the protocol server MUST return the following:

      ▪ Permission information of the list that contains the folder.

      ▪ Subset of metadata information about the items of the folder. This information SHOULD contain the **ows_Modified**, **ows_ServerRedirected, ows_UniqueId**, **ows_scopeID**, **ows_FileRef, ows_ID** and **ows_FSObjType** attributes. The protocol client MUST ignore rest of the attributes.

   ▪ If the value of the *securityOnly* input parameter is **false**, the protocol server MUST return the following:

      ▪ Permission information of the list that contains the folder.

      ▪ All the metadata information about the items of the folder as defined by the **Xml** element in section 3.1.4.5.3.5.

   ▪ The protocol client MUST specify the following:

      ▪ The folder name in the input parameter *folderUrl*, which specifies the list root folder relative URL. If *folderUrl* is empty, the protocol server MUST consider *folderUrl* as the root folder of the list identified by the *objectId* parameter.

      ▪ A value for *lastItemIdOnPage* input parameter for pagination. If there is more content than one **GetContent** operation can retrieve (the returned *lastItemIdOnPage* is not null), the next **GetContent** operation's *lastItemIdOnPage* parameter SHOULD be set to the *lastItemIdOnPage* value of the previous **GetContentResponse** to retrieve the remaining content.

   ▪ The protocol server MUST ignore the rest of the input parameters. The protocol server MUST support pagination; it MUST set *lastItemIdOnPage* to the identification of the last item in the result (in the format of a 32-bit integer), or it MUST set *lastItemIdOnPage* to "NULL" if there are no more items to fetch. The information MUST be returned as an XML node whose schema is defined in section 3.1.4.5.2.11.

▪If the value of the input parameter *objectId* is empty, the protocol server MUST return information about all the child folders, documents, and pages of the folder in the context site. The protocol client MUST either specify the folder name in the input parameter *folderUrl* or pass an empty value. If *folderUrl* is empty, the protocol server MUST use the root folder of the (context site) as the default. The protocol server MUST ignore the rest of the input parameters. The information MUST be returned as an XML node whose schema is defined in section 3.1.4.5.2.2.

8. If the value of the *objectType* input parameter is "ListItemAttachments", the protocol server MUST return information about all the attachments of the list item identified by the input parameters *objectId*, *itemId* and *folderUrl*, and the following apply:

▪The protocol client MUST specify the following:

- The name or GUID of the list in the *objectId* input parameter. The GUID SHOULD be enclosed in curly braces ({}).

- The identification of the list item in the *itemId* input parameter.

- The folder name in the *folderUrl* input parameter. If the input parameter *folderUrl* is empty, the protocol server MUST consider *folderUrl* as the root folder of the list identified by *objectId* input parameter.

- The protocol server MUST ignore the rest of the input parameters. The information MUST be returned as an XML node whose schema is defined in section 3.1.4.5.2.6.

3.1.4.5.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.5.1.1 GetContentSoapIn

The **GetContentSoapIn** message contains the **GetContent** SOAP element.

The SOAP action value of the message is defined as:

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

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

3.1.4.5.1.2 GetContentSoapOut

The **GetContentSoapOut** message contains the **GetContentResponse** element.

The SOAP action value of the message is defined as:

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

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

3.1.4.5.2 Elements

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

3.1.4.5.2.1 ContentDatabase

The **ContentDatabase** element contains change information and information about all the top level sites (2) served by the content database.

```
<s:element name="ContentDatabase">
  <s:complexType>
    <s:sequence>
      <s:element name="Metadata" type="tns:ContentDatabaseMetadata"
                 minOccurs="1" maxOccurs="1" />
      <s:element name="Sites" type="tns:Sites" minOccurs="0" maxOccurs="1" />
    </s:sequence>
  </s:complexType>
```

```
        </s:element>
```

Metadata: Specifies change information for the content database as defined in section 2.2.4.7.

Sites: Specifies an array of information about the top level sites (2) in the content database as defined in section 3.1.4.5.3.8.

3.1.4.5.2.2   FPFolder

The **FPFolder** element contains information about all the child folders, documents and pages in the folder.

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

FPFolder: Contains information about all the child folders, documents and pages in the folder. It MUST be of type section 3.1.4.5.3.3.

3.1.4.5.2.3   GetContent

The **GetContent** element is the request message for the **GetContent** operation. The definition is as follows.

```
<s:element name="GetContent">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="objectType"
              type="tns:ObjectType" />
      <s:element minOccurs="0" maxOccurs="1" name="objectId" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="folderUrl" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="itemId" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="retrieveChildItems"
              type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="securityOnly"
              type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="lastItemIdOnPage"
              type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

objectType: Specifies the type of object whose information is required. It is of type **ObjectType**, as defined in section 2.2.5.3.

objectId: Specifies the name or GUID of an object. The GUID SHOULD be enclosed in curly braces ({}). The detailed usage of this parameter is listed in section 3.1.4.5.

folderUrl: Specifies the folder name or the URL of a folder. The URL MUST be an absolute URL or a server-relative URL or a list root folder relative URL. The detailed usage of this parameter is listed in section 3.1.4.5.

itemId: Specifies the identification of the list item. The detailed usage of this parameter is listed in section 3.1.4.5.

retrieveChildItems: Specifies whether to get information about the child sites (2), lists, and folders. The detailed usage of this parameter is listed in section 3.1.4.5.

securityOnly: Specifies whether to get all the information for a list item or items of a folder. If **true**, the operation MUST return only the mandatory information, otherwise the operation MUST return all the information. The detailed usage of this parameter is listed in section 3.1.4.5.

lastItemIdOnPage: This element is used for pagination. Protocol clients use this to sequentially retrieve large amount of data. The detailed usage of this parameter is listed in section 3.1.4.5.

3.1.4.5.2.4 GetContentResponse

The **GetContentResponse** element is the response message for the **GetContent** operation.

The definition is as follows.

```
<s:element name="GetContentResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetContentResult"
              type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="lastItemIdOnPage"
              type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetContentResult: Specifies the result information as an XML node. The detailed schema of this XML node is defined in 3.1.4.5.

lastItemIdOnPage: Specifies the identification of the last item in the result data. The detailed usage is listed in section 3.1.4.5.

3.1.4.5.2.5 Item

The **Item** element contains an element of type **ItemData**.

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

Item: Specifies permission information and metadata information about the list item or the items of the folder as defined in section 3.1.4.5.3.5.

3.1.4.5.2.6 Item Attachments

The **Item Attachments** element contains an array of information about all the attachments of a list item.

```
<s:element name="Item">
  <s:complexType>
    <s:sequence>
      <s:element maxOccurs="unbounded" name="Attachment">
        <s:complexType>
          <s:attribute name="URL" type="s:string" use="required" />
        </s:complexType>
      </s:element>
```

```
      </s:sequence>
      <s:attribute name="Count" type="s:unsignedInt" use="required" />
    </s:complexType>
  </s:element>
```

Count: Specifies the number of attachments in the list item.

Attachment.URL: Specifies the absolute URL of the attachment.

3.1.4.5.2.7   List

The **List** element contains information about a list.

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

List: Specifies list information. It MUST be of type **List**, as defined in section 2.2.4.13.

3.1.4.5.2.8   Site

The **Site** element contains information about a site collection.

```
<s:element name="Site">
  <s:complexType>
    <s:sequence>
      <s:element name="Metadata" type="tns:SPSiteMetadata" minOccurs="1"
              maxOccurs="1" />
      <s:element name="Groups" type="tns:GroupMembership" minOccurs="1"
              maxOccurs="1" />
      <s:element name="Web" type="tns:Web" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

Metadata: Specifies the metadata for the site collection, as defined in section 2.2.4.21.

Groups: Specifies information about the groups (2) and users of the site collection, as defined in section 2.2.4.10.

Web: Specifies the information of the root site of the site collection, as defined in section 3.1.4.5.2.10.

3.1.4.5.2.9   VirtualServer

The **VirtualServer** element contains information about a Web application (2).

```
<s:element name="VirtualServer">
  <s:complexType>
    <s:sequence>
      <s:element name="Metadata" type="tns:SPWebAppMetadata" minOccurs="1" maxOccurs="1"/>
      <s:element name="ContentDatabases" type="tns:ContentDatabases" minOccurs="0"
              maxOccurs="1"/>
      <s:element name="Policies" type="tns:Policies" minOccurs="0" maxOccurs="1"/>
      <s:element name="Servers" minOccurs="0">
        <s:complexType>
```

```
        <s:sequence>
          <s:element name="Server" minOccurs="0 minOccurs="unbounded">
            <s:complexType>
              <s:attribute name="URL" type="s:string"/>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
</s:element>
```

Metadata: Specifies information about the Web application (2), as defined in section 2.2.4.22.

ContentDatabases: Specifies information about all the content databases in the Web application (2), as defined in section 3.1.4.5.3.1.

Policies: Specifies an array of permission information about all the users of the Web application (2), as defined in section 3.1.4.5.3.6.

Servers: A list of server URIs that are available to access content through the SiteData service.<33>

3.1.4.5.2.10   Web

The **Web** element specifies information about a site (2). This information includes the following:

▪ Metadata information.

▪ User information.

▪ Permissions information.

▪ Information about all the subsites and lists.

▪ Information about the root folder.

```
<s:complexType name="Web">
    <s:sequence>
      <s:element name="Metadata" type="tns:WebMetadata" minOccurs="1"
              maxOccurs="1"/>
      <s:element name="Users" type="tns:Users" minOccurs="1" maxOccurs="1"/>
      <s:element name="ACL" type="tns:ACL" minOccurs="1" maxOccurs="1"/>
      <s:element name="Webs" type="tns:Webs" minOccurs="1" maxOccurs="1"/>
      <s:element name="Lists" type="tns:Lists" minOccurs="1" maxOccurs="1"/>
      <s:element name="FPFolder" type="tns:FolderData" minOccurs="0"
              maxOccurs="1"/>
    </s:sequence>
</s:complexType>
```

Metadata: Specifies the metadata information for a site (2) as defined in section 2.2.4.27.

Users: Specifies information of the users of a site (2). It is an array of user information as defined in section 2.2.4.24.

ACL: If the site (2) has an external security provider, it specifies the permission information; the schema of this information depends on the implementation of the external security provider. Otherwise, it specifies an array of permission information about all the **role assignments** of the site (2), as defined in section 2.2.4.3.

Webs: Specifies information about all the subsites in the site (2), as defined in section 2.2.4.28.

Lists: Specifies information about all the lists of the site (2), as defined in section 2.2.4.15.

FPFolder: Specifies information about the all the child folders, files, and pages of the root folder of a site (2), as defined in section 3.1.4.5.3.3.

3.1.4.5.2.11   Folder

The **Folder** element contains an element of type **ItemData**.

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

Folder: Specifies permission and metadata information about the list item or the items of the folder, as defined in section 3.1.4.5.3.5.

3.1.4.5.3   Complex Types

The following XML schema complex type definitions are specific to this operation.

3.1.4.5.3.1   ContentDatabases

The **ContentDatabases** complex type contains an array of information of all the content databases in a Web application.

```
<s:complexType name="ContentDatabases">
  <s:sequence>
    <s:element maxOccurs="unbounded" name="ContentDatabase">
      <s:complexType>
        <s:attribute name="ID" type="s:string" use="required" />
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
```

ContentDatabase.ID: Specifies the GUID of the content database. This GUID MUST be enclosed in curly braces ({}).

3.1.4.5.3.2   Files

The **Files** complex type contains an array of information about all the documents and pages in a folder.

```
<s:complexType name="Files">
  <s:sequence>
    <s:element name="File" minOccurs="0"  maxOccurs="unbounded">
      <s:complexType>
        <s:attribute name="URL" type="s:string" use="required" />
        <s:attribute name="ID" type="s:string" use="required" />
```

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

File.URL: Specifies the absolute URL of the document or page.

File.ID: Specifies the GUID of the document or page. This GUID MUST be enclosed in curly braces ({}).

File.LastModified: Specifies the date and time, as a [ISO-8601] separate date and time UTC value, when the document or page was last modified.

3.1.4.5.3.3   FolderData

The **FolderData** complex type contains information about all the child folders, documents, and pages in the folder.

```
    <s:complexType name="FolderData">
      <s:sequence>
        <s:element name="Folders" type="tns:Folders"  minOccurs="0" maxOccurs="unbounded" />
        <s:element name="Files" type="tns:Files" minOccurs="0" maxOccurs="unbounded" />
      </s:sequence>
    </s:complexType>
```

Folders: Specifies an array of information about all the child folders in the folder, as defined in section 3.1.4.5.3.4.

Files: Specifies an array of information of the all documents and pages in the folder, as defined in section 3.1.4.5.3.2.

3.1.4.5.3.4   Folders

The **Folders** complex type contains an array of information about all the child folders of a folder.

```
    <s:complexType name="Folders">
      <s:sequence>
        <s:element name="Folder" maxOccurs="unbounded" >
          <s:complexType>
            <s:attribute name="URL" type="s:string" use="required" />
            <s:attribute name="ID" type="s:string" use="required" />
            <s:attribute name="LastModified" type="s:string" use="required" />
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
```

Folder.URL: Specifies the absolute URL of the folder.

Folder.ID: Specifies the GUID of the folder. This GUID MUST be enclosed in curly braces ({}).

Folder.LastModified: Specifies the date and time, as a [ISO-8601] separate date and time UTC value, when the folder was last modified.

3.1.4.5.3.5 ItemData

The **ItemData** complex type has two parts:

```
<s:complexType name="ItemData">
  <s:sequence>
    <s:element name="Metadata" type="tns:Metadata" />
    <s:element name="xml" type="rs:row" />
  </s:sequence>
</s:complexType>
```

Metadata: Specifies permission information about all the role assignments of the list containing the folder or the list item.

```
<s:element name="Metadata">
  <s:complexType>
    <s:sequence>
      <s:element name="scope">
        <s:complexType>
          <s:sequence>
            <s:element name="permissions" type="tns:permissionsForACL" />
          </s:sequence>
          <s:attribute name="id" type="s:string" use="required" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

scope.id: Specifies the GUID of the collection of role assignments of the list or the folder of the list.

scope.permissions: Specifies an array of permission information about all the role assignments of the list or the folder, as defined in section 2.2.4.17.

xml: Specifies the metadata information about the list item or the items of the folder. This MUST be a well-formed XML string compliant with Microsoft ADO Persistence Format schema, as specified in [MS-PRSTFR] (also see section 8). It consists of an XDR schema for rows (all fields (2) with their returned and original names, data types and maximum length) and row data, one row for each item returned. The row data MUST contain an additional attribute named "ows_ServerRedirected". This **ows_ServerRedirected** attribute indicates whether users who do not have access to view the item or who do not have the appropriate client application installed are redirected to server-side rendering of the item when they click a link to view it. The **ows_ServerRedirected** attribute MUST have one of the values listed in the following table.

Value	Description
0	Not redirected.
1	Redirected.

3.1.4.5.3.6 Policies

The **Policies** complex type contains an array of permission information about the users of the Web application.

```
<s:complexType name="Policies">
  <s:sequence>
    <s:element maxOccurs="unbounded" name="PolicyUser" type="tns:PolicyUser" />
  </s:sequence>
  <s:attribute name="AnonymousGrantMask" type="s:unsignedLong" use="required" />
  <s:attribute name="AnonymousDenyMask" type="s:unsignedLong" use="required" />
</s:complexType>
```

PolicyUser: Specifies permission information about the user, as defined in section 3.1.4.5.3.7.

AnonymousGrantMask: A mask specifying the grant permissions for an anonymous user, as defined in [MS-WSSFO2] section 2.2.2.14.

AnonymousDenyMask: A mask specifying the deny permissions for an anonymous user, as defined in [MS-WSSFO2] section 2.2.2.14.

3.1.4.5.3.7 PolicyUser

The **PolicyUser** complex type contains permission information of the user of the Web application.

```
<s:complexType name="PolicyUser">
  <s:attribute name="LoginName" type="s:string" use="required" />
  <s:attribute name="Sid" type="s:string" use="required" />
  <s:attribute name="GrantMask" type="s:unsignedLong" use="required" />
  <s:attribute name="DenyMask" type="s:unsignedLong" use="required" />
  <s:attribute name="BinaryIdentifier" type="s:string" use="optional" />
  <s:attribute name="BinaryIdentifierType" type="s:string" use="optional" />
</s:complexType>
```

LoginName: Specifies the login name of the user.

Sid: Specifies the SID of the user.

GrantMask: A mask specifying the permissions that are granted to the user of the Web application, as specified in [MS-WSSFO2] section 2.2.2.14.

DenyMask: A mask specifying the permissions that are denied to the user of the Web application, as specified in [MS-WSSFO2] section 2.2.2.14.

BinaryIdentifier: Specifies the unique identifier of the user.

BinaryIdentifierType: The type of unique identifier represented by the **BinaryIdentifier** value, either "SecurityIdentifier" or "UserKey". If this value is "SecurityIdentifier", then the **BinaryIdentifier** value represents the SID of the user. If the value is "UserKey", the **BinaryIdentifier** is a Provider User Key.

3.1.4.5.3.8 Sites

The **Sites** complex type contains an array of information about all the site collections served by the content database.

```
<s:complexType name="Sites">
  <s:sequence>
    <s:element maxOccurs="unbounded" name="Site">
      <s:complexType>
```

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

Site.URL: Specifies the absolute URL of the site collections.

Site.ID: Specifies the GUID of the site collections. This GUID MUST be enclosed in curly braces ({}).

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.4.6 GetContentEx

The **GetContentEx** operation<34> is used by the protocol client to request content information, similar to **GetContent** as described in section 3.1.4.5. **GetContentEx** has the following **SOAPAction** header value:

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

GetContentEx has the following schema:

```
<wsdl:operation name="GetContentEx">
    <wsdl:input message="GetContentExSoapIn" />
    <wsdl:output message="GetContentExSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetContentExSoapIn** request message and the protocol server responds with a **GetContentExSoapOut** response message.

3.1.4.6.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.6.1.1 GetContentExSoapIn

GetContentExSoapIn is the request message of the **GetContentEx** operation.

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

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

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

3.1.4.6.1.2 GetContentExSoapOut

GetContentExSoapOut is the response message to **GetContentEx** operation.

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

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

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

3.1.4.6.2 Elements

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

3.1.4.6.2.1 GetContentEx

The **GetContentEx** element is an element of the **GetContentExSoapIn** request message, and has the following schema:

```
<s:element name="GetContentEx">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="version" type="s:int" />
      <s:element minOccurs="1" maxOccurs="1" name="xmlInput" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

version: Specifies an integer version number to identify what version of the protocol the protocol client understand. This value MUST be set to 2.

xmlInput: Specifies a string specifying a **GetContentExRequest** element.

3.1.4.6.2.2 GetContentExRequest

The **GetContentExRequest** element is an element of the **GetContentExSoapIn** request message, and has the following schema:

```
<s:element name="GetContentExRequest">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ObjectType" type="tns:ObjectType" />
      <s:element minOccurs="0" maxOccurs="1" name="ObjectId" type="s:string" />
```

```
            <s:element minOccurs="0" maxOccurs="1" name="FolderUrl" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="ItemId" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="RetrieveChildItems" type="s:boolean" />
            <s:element minOccurs="0" maxOccurs="1" name="SecurityOnly" type="s:boolean" />
            <s:element minOccurs="0" maxOccurs="1" name="LastItemIdOnPage" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="AllowRichText" type="s:boolean" />
            <s:element minOccurs="0" maxOccurs="1" name="RequestLoad" type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="RemoveInvalidXmlChars" type="s:boolean" />
        </s:sequence>
    </s:complexType>
</s:element>
```

ObjectType: Specifies the type of object whose information is required. It is of type **ObjectType**, as defined in section 2.2.5.3.

ObjectId: Specifies the name or GUID of an object. The GUID SHOULD be enclosed in curly braces ({}). The detailed usage of this parameter is listed in section 3.1.4.5.

FolderUrl: Specifies the folder name or the URL of a folder. The URL MUST be an absolute URL or a server-relative URL or a list root folder relative URL. The detailed usage of this parameter is listed in section 3.1.4.5.

ItemId: Specifies the identification of the list item. The detailed usage of this parameter is listed in section 3.1.4.5.

RetrieveChildItems: Specifies whether to get information about the child sites (2), lists, and folders. The detailed usage of this parameter is listed in section 3.1.4.5.

SecurityOnly: Specifies whether to get all the information for a list item or items of a folder. If **true**, the operation MUST return only the mandatory information, otherwise the operation MUST return all the information. The detailed usage of this parameter is listed in section 3.1.4.5.

LastItemIdOnPage: Is used for pagination. Protocol clients use this to sequentially retrieve large amount of data. The detailed usage of this parameter is listed in section 3.1.4.5.

AllowRichText: Is used to specify whether the item metadata returned in the result should contain rich text tags.

RequestLoad: Specifies a value that determines how many items can be fetched in the current operation, where the value represents the percentage of total changes that can be processed in the request. If this value is not specified or is 100, the protocol server MUST return the whole batch that can be processed in a single request. If this value is x, the protocol server MUST return min(1, x%) of the batch that can be processed in the request. This value MUST NOT be less than zero or greater than 100.

RemoveInvalidXmlChars: Specifies whether invalid XML chars should be removed from the result.

3.1.4.6.2.3   GetContentExResponse

The **GetContentExResponse** element is the sole content of the **GetContentExSoapOut** response message.

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

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

GetContentExResult contains a **GetContentResponse** element as described in section 3.1.4.5.2.4.

3.1.4.6.3 Complex Types

None.

3.1.4.6.4 Simple Types

None.

3.1.4.6.5 Attributes

None.

3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.4.7 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 (2).

```
<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.

3.1.4.7.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.7.1.1 GetListSoapIn

The **GetListSoapIn** message contains the **GetList** element, which 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. It requests general information about the list, its schema, and access permissions.

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

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

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

3.1.4.7.1.2 GetListSoapOut

GetListSoapOut 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 follows.

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

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

3.1.4.7.2 Elements

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

3.1.4.7.2.1 GetList

The definition of the **GetList** element is as follows.

```
<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: Can be either a list name, such as "Documents", or a GUID of the list, with or without curly braces, in the following format:

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

3.1.4.7.2.2 GetListResponse

The definition of the **GetListResponse** element is as follows.

```
<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: Specifies the status of the operation. On successful completion of the operation, the protocol server MUST set it to 0.

sListMetadata: Specifies metadata information about the list, as defined in section 3.1.4.7.3.1.

vProperties: See section 3.1.4.7.3.2.

If the list name or the GUID supplied by the protocol client are not well-formed or do not correspond to any list that the target site (2) contains, then the protocol server MUST return a SOAP fault.

Note that this protocol assumes that the list collection of any site (2) is non-hierarchical, that is, lists do not have sublists. Hence, list name is not a composite name and MUST NOT contain slashes. One example of an ill-formed name is, "Lists/Contacts".

3.1.4.7.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

3.1.4.7.3.1 _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: Specifies the title of the list.

Description: Specifies the description of the list.

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

BaseTemplate: Specifies the name of the base template (end-user look-and-feel) of the list. It MUST be of type **ListBaseTemplate**, as defined in section 2.2.5.2.

DefaultViewUrl: Specifies the server-relative URL of the page containing the default view for the list.

LastModified: Specifies the time of last modification (for example, item added) as a locale-dependent short date value and longtime value, separated by a space, in UTC.

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

Author: Specifies the full name of the user who created the list.

ValidSecurityInfo: A Boolean value that MUST be **true** if, and only if, the requester has the right to view permissions.

InheritedSecurity: A Boolean value that MUST be **true** if, and only if, the list inherits security from its parent site (2).

AllowAnonymousAccess: A Boolean value that MUST be **true** if, and only if, anonymous users have access permissions for the site (2).

AnonymousViewListItems: A Boolean value that MUST be **true** if, and only if, anonymous users can view the list items of the list.

ReadSecurity: Specifies read security setting. Possible values are 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** has no rights to read permissions, the field (2) is empty. If **InheritedSecurity** is set to **true**, the **Permissions** element is not returned. Otherwise (when the list has individually configured permissions), **Permissions** string contains a well-formed XML **Permissions** element containing an array of **Permission** elements, as specified in section 3.1.4.7.3.4.

3.1.4.7.3.2   ArrayOf_sProperty

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

```
  <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 3.1.4.7.3.3.

Release: February 11, 2013

3.1.4.7.3.3   _sProperty

The **_sProperty** complex type defines the name and type of a list field (2).

```
<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: Name of the field, which can be used to construct a query or interpret **GetListItems** response (see section 3.1.4.9).

Title: Title shown in the site (2) pages.

Type: Type of the field, as specified in [MS-WSSTS] section 2.5.

3.1.4.7.3.4   ListPermissions

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

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

Permission: See section 3.1.4.7.3.5.

3.1.4.7.3.5   ListPermission

The **ListPermission** complex type specifies information about the permissions granted to certain users or groups (2) 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="s: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>
```

MemberID: Identification of the user or the group in site membership.

Mask: A mask of permission flags as defined in [MS-WSSFO2] section 2.2.2.14.

MemberIsUser: true if **MemberID** specifies a user in the in site membership; **false** if **MemberID** specifies a group. Note that the user in site membership can be an Active Directory Group, as specified in [MS-ADTS]. Each member of the Active Directory Group is then impersonated by the same user and has the same permissions as far as this protocol is concerned.

MemberGlobal: true if **MemberID** specifies a group, **false** otherwise; actually, the inverse of **MemberIsUser**. This attribute is obsolete.

UserLogin: Specifies the login name of the user. MUST be present when **MemberIsUser** is set to **true**.

GroupName: Contains group name as defined in the site membership, for example, "Home Owners". MUST be present when **MemberIsUser** is set to **false**.

3.1.4.7.4 Simple Types

None.

3.1.4.7.5 Attributes

None.

3.1.4.7.6 Groups

None.

3.1.4.7.7 Attribute Groups

None.

3.1.4.8 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.

3.1.4.8.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.8.1.1 GetListCollectionSoapIn

This 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.

3.1.4.8.1.2 GetListCollectionSoapOut

This 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.

3.1.4.8.2 Elements

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

3.1.4.8.2.1 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>
```

3.1.4.8.2.2 GetListCollectionResponse

The **GetListCollectionResponse** element is defined as follows.

```
<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: Specifies the status of the operation. On successful completion of the operation, the protocol server MUST set it to 0.

vLists: Specifies an array of list information, about all the lists in the context site. This element is of type **Array Of_sList**, as defined in section 3.1.4.8.3.2.

If the protocol server encounters an error during this operation, a SOAP fault MUST be returned. There are no errors specific to this operation.

3.1.4.8.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

3.1.4.8.3.1   _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: Specifies the GUID of the list. This GUID MUST be enclosed in curly braces ({}).

Title: Specifies the title of the list.

Description: Specifies the description of the list.

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

BaseTemplate: Specifies the name of the base template (end-user look-and-feel) of the list. It MUST be of type section 2.2.5.2.

DefaultViewUrl: Specifies the server-relative URL of the page containing the default view for the list.

LastModified: Specifies the date and time when the list was last modified.

PermId: Not used. This element MUST NOT be included.

InheritedSecurity: A Boolean value that MUST be **true** if, and only if, this list inherits security from its parent site (2).

AllowAnonymousAccess: A Boolean value that MUST be **true** if, and only if, anonymous users have access permissions for the list.

AnonymousViewListItems: A Boolean value that MUST be **true** if, and only if, anonymous users can view the list items of the list.

ReadSecurity: Specifies what items users have read access to. Possible values are 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.

3.1.4.8.3.2 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: contains metadata information about a list as defined in section 3.1.4.8.3.1.

3.1.4.8.4 Simple Types

None.

3.1.4.8.5 Attributes

None.

3.1.4.8.6 Groups

None.

3.1.4.8.7 Attribute Groups

None.

3.1.4.9 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, which defines the selection criteria and the ordering of list items in the result.

- The maximum number of list items to be included into the result.

The protocol server responds by sending a **GetListItemsSoapOut** message, which 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 Persistence format, as defined in [MS-PRSTFR], (see also section 8 with one row for each list item that matches the selection criteria).

The protocol client limits the number of list items to be returned by the *uRowLimit* parameter of the **GetListItems** element. This *uRowLimit* parameter has the same meaning as the ANSI-SQL **top** clause.

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 (2) (see section 3.1.4.9.3.1). The protocol client can optionally include that field in the OrderBy clause (see section 3.1.4.9.3.2), 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 well-known minimum value of 1, hence the protocol client can specify the criterion **ID** is greater than zero to start the pagination.

In the response, row attribute names are not the same as field names prescribed by the schema returned by **GetList**. To relate row attributes to field names, the protocol client has to look up XDR schema of the response, and then look up the schema returned by **GetList,** as specified in section 3.1.4.7.

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 attribute, **ows_ContentTypeId**.

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

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

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

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

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

```
            </_sProperty>
```

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

3.1.4.9.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.9.1.1 GetListItemsSoapIn

The **GetListItemsSoapIn** message contains the **GetListItems** SOAP element.

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.

3.1.4.9.1.2 GetListItemsSoapOut

The **GetListItemsSoapOut** message contains the **GetListItemsResponse** element.

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

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

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

3.1.4.9.2 Elements

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

3.1.4.9.2.1 GetListItems

The **GetListItems** element contains details of the **GetListItems** request.

```
<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 SHOULD be enclosed in curly braces ({}).

strQuery: A string that, unless empty, MUST contain well-formed XML elements **Where**, as described in section 3.1.4.9.3.1, and **OrderBy**, as described in 3.1.4.9.3.2. This **strQuery**, if not empty, MUST contain at most one **Where** clause and at most one **OrderBy** clause, in that order. If empty, prescribes to return all list items in any order.

strViewFields: It is not used and MUST be empty.

uRowLimit: This unsigned integer indicates the maximum number of list items allowed in the response. The protocol server MUST NOT return any more list items in the response than specified by this element.

3.1.4.9.2.2   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: This MUST be a well-formed XML string compliant with Microsoft ADO Persistence format schema as specified in [MS-PRSTFR]. It consists of XDR schema for rows (all fields (2) 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 as a UTC value.

3.1.4.9.3   Complex Types

The following XML schema complex type definitions are specific to this operation.

3.1.4.9.3.1   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)** schema 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 this criterion would be:

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

The **Where** clause can contain an arbitrary number of conditions, joined into And/Or groups (2) 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**, **Get**, or **Le**, as specified in the complete XSD in [MS-WSSCAML].

- **FieldRef** element is a reference to one of the fields. **FieldRef** name attribute MUST be a valid field name. Valid field names are provided in the **Name** attribute of the **_sProperty** element (see section 3.1.4.7.3.3) returned from the **GetList** operation.

- **Value** of appropriate type suitable to be compared using the **Predicate** with the field in question. Two predicates (**IsNull** and **IsNotNull**) do not require a value for comparison.

3.1.4.9.3.2   Order By 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 (2).

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 (see section 3.1.4.7.3.3) returned from the **GetList** operation. The **Ascending** attribute is optional and indicates ascending order if value is **true**, descending otherwise. By default, **Ascending** is **true**.

3.1.4.9.4   Simple Types

None.

3.1.4.9.5   Attributes

None.

3.1.4.9.6   Groups

None.

3.1.4.9.7   Attribute Groups

None.

3.1.4.10   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.

3.1.4.10.1   Messages

The following WSDL message definitions are specific to this operation.

3.1.4.10.1.1   GetSiteSoapIn

This 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.

3.1.4.10.1.2   GetSiteSoapOut

This 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.

3.1.4.10.2 Elements

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

3.1.4.10.2.1 GetSite

The definition of the **GetSite** element is as follows.

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

3.1.4.10.2.2 GetSiteResponse

The definition of the **GetSiteResponse** element is as follows.

```
<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: Specifies the status of the operation. On successful execution of the operation, the protocol server MUST set it to 0.

sSiteMetadata: Specifies the metadata information about the site collection, as defined in section 3.1.4.10.3.1.

vWebs: Specifies an array of information about all the subsites in the site collection. **vWebs** MUST be returned only if the **SmallSite** property of the output parameter *sSiteMetadata* is **true** and the current user is the administrator of the site collection.

strUsers: The protocol server MUST set it to **null**.

strGroups: Specifies information about all the groups (2) in the site collection. It is a string that contains an XML node whose schema is as defined in section 3.1.4.10.2.3.

vGroups: Specifies information about the users of all the groups (2) for the site collection. **vGroups** is an array of strings as defined in section 2.2.4.6. Each string in the array is an XML node

containing information about all the users of a group. 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 in section 3.1.4.10.2.4.

If the protocol server encounters an error during this operation, a SOAP fault MUST be returned. There are no errors specific to this operation.

3.1.4.10.2.3 Groups

The **Groups** element contains a list of group (2) information.

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

Groups: See section 2.2.4.11.

3.1.4.10.2.4 Users

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

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

Users: See section 2.2.4.24.

3.1.4.10.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

3.1.4.10.3.1 _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="0" maxOccurs="1" name="UserProfileGUID"
              type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ValidSecurityInfo"
             type="s:boolean" />
  </s:sequence>
</s:complexType>
```

LastModified: Specifies the date and time, as a locale-dependent short date value and longtime value, separated by a space in UTC, when the site collection was last modified.

LastModifiedForceRecrawl: Specifies the date and time, as a locale-dependent short date value and long time value, separated by a space in UTC, when the permissions of any structural element of the site collection were last modified.

SmallSite: A Boolean value that MUST be **true** if, and only if, the site collection has less than 1000 subsites. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

PortalUrl: Specifies the URL of the portal associated with the site collection.

UserProfileGUID: If the site (2) is a personal site<35>, it MUST specify the GUID of the [user profile] who owns the site (2). Otherwise, it MUST be null.

ValidSecurityInfo: A Boolean value that MUST be **true** if, and only if, the current user is the administrator of the site collection. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

3.1.4.10.4 Simple Types

None.

3.1.4.10.5 Attributes

None.

3.1.4.10.6 Groups

None.

3.1.4.10.7 Attribute Groups

None.

3.1.4.11 GetSiteAndWeb

The **GetSiteAndWeb** operation is used to get the URL of the site collection and the site (2) 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.

3.1.4.11.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.11.1.1 GetSiteAndWebSoapIn

This is the request message to get the URL of the site collection and the site (2) 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.

3.1.4.11.1.2 GetSiteAndWebSoapOut

This is the response message to get the URL of the site collection and the site (2) 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.

3.1.4.11.2 Elements

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

3.1.4.11.2.1 GetSiteAndWeb

The definition of the **GetSiteAndWeb** element is as follows.

```
<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: Specifies the URL of any item on the site (2).

3.1.4.11.2.2 GetSiteAndWebResponse

The definition of the **GetSiteAndWebResponse** element is as follows.

```
<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: Contains the status of the operation. On successful completion of the operation, the protocol server MUST set it to zero.

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

strWeb: Specifies the absolute URL of the site (2) that contains the item specified by the *strUrl* parameter.

If the protocol server encounters an error during 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".<36>

3.1.4.11.3 Complex Types

None.

3.1.4.11.4 Simple Types

None.

3.1.4.11.5 Attributes

None.

3.1.4.11.6 Groups

None.

3.1.4.11.7 Attribute Groups

None.

3.1.4.12 GetSiteUrl

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

```
<wsdl:operation name="GetSiteUrl">
    <wsdl:input message="GetSiteUrlSoapIn" />
    <wsdl:output message="GetSiteUrlSoapOut" />
</wsdl:operation>
```

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

3.1.4.12.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.12.1.1 GetSiteUrlSoapIn

This is the request message to get the GUID and URL of the site collection to which the input URL belongs.

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

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

The SOAP body contains a GetSiteUrl element.

3.1.4.12.1.2 GetSiteUrlSoapOut

This is the response message to get the GUID and URL of the site collection to which the input URL belongs.

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

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

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

3.1.4.12.2 Elements

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

3.1.4.12.2.1 GetSiteUrl

The **GetSiteUrl** element is defined as follows.

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

Url: Specifies the URL of any item on the site (2).

3.1.4.12.2.2 GetSiteUrlResponse

The **GetSiteUrlResponse** element is defined as follows.

```
<s:element name="GetSiteUrlResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetSiteUrlResult"
              type="s:unsignedInt" />
      <s:element minOccurs="1" maxOccurs="1" name="siteUrl" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="siteId" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetSiteUrlResult: Contains the status of the operation. On successful completion of the operation, the protocol server MUST set it to 0.

siteUrl: Specifies the absolute URL of the root site (2) of the site collection which contains the item specified by the *Url* parameter.

siteId: Specifies the GUID of the site collection that contains the item specified by the *Url* parameter. The GUID MUST be enclosed in curly braces ({}).

If the protocol server encounters an error during this operation, a SOAP fault MUST be returned which SHOULD contain one of the error codes described in the following table.

Error code	Error message	Error condition
None	Value cannot be null. Parameter name: uriString	**Url** element is missing.
None	Invalid URI: The URI is empty.	Specified **Url** element is empty.

3.1.4.12.3 Complex Types

None.

3.1.4.12.4 Simple Types

None.

3.1.4.12.5 Attributes

None.

3.1.4.12.6 Groups

None.

3.1.4.12.7 Attribute Groups

None.

3.1.4.13 GetURLSegments

The **GetURLSegments** operation is used to get the identifiers of a site (2), 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 response message from the protocol server contains different GUIDs and identifiers depending on the type of element addressed by the *strURL* parameter, as follows:

  - If the *strURL* parameter is the URL of the context site<37>, and if the context site has an external security provider, the protocol server MUST return the GUID of the site (2) and the GUID of the external security provider.

  - If the *strURL* parameter is a URL of a list item within the context site<38>, the protocol server MUST return:

    - The GUID of the list that contains the list item. This GUID MUST be enclosed in curly braces ({}).

- The identification of the list item.

  - Else if the *strURL* parameter is a URL of a list within the request site (2)<39>, the protocol server MUST return the GUID of the list. This GUID MUST be enclosed in curly braces ({}).

  - Else if the *strURL* parameter is a URL of a document within the context site<40>, the protocol server MUST return:

    - The GUID of the list that contains the document list item. The GUID MUST be enclosed in curly braces ({}).

    - The identification of the document list item.

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

3.1.4.13.1   Messages

The following WSDL message definitions are specific to this operation.

3.1.4.13.1.1   GetURLSegmentsSoapIn

This 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.

3.1.4.13.1.2   GetURLSegmentsSoapOut

This 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.

3.1.4.13.2   Elements

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

3.1.4.13.2.1   GetURLSegments

The definition of the **GetURLSegments** element is as follows.

```
<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: Specifies a lower-cased absolute URL of any item in the context site. MUST NOT be NULL.

3.1.4.13.2.2 GetURLSegmentsResponse

The definition of the **GetURLSegmentsResponse** element is as follows.

```
<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: Specifies whether operation returned a result, **true** or **false**.

strWebID: Specifies the GUID of the site (2). This GUID MUST be enclosed in curly braces ({}).

strBucketID: Specifies the GUID of the external security provider. This GUID MUST be enclosed in curly braces ({}).

strListID: Specifies the GUID of the list. This GUID MUST be enclosed in curly braces ({}).

strItemID: Specifies the identification of the list item.

3.1.4.13.3 Complex Types

None.

3.1.4.13.4 Simple Types

None.

3.1.4.13.5 Attributes

None.

3.1.4.13.6 Groups

None.

3.1.4.13.7 Attribute Groups

None.

3.1.4.14 GetWeb

The **GetWeb** operation is used to get the metadata information, information of all the subsites, lists, and role definitions about 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.

3.1.4.14.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.14.1.1 GetWebSoapIn

It is the request message to get information about a site (2).

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.

3.1.4.14.1.2 GetWebSoapOut

It is the response message to get information about a site (2).

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.

3.1.4.14.2 Elements

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

3.1.4.14.2.1 GetWeb

The definition of the **GetWeb** element is as follows.

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

3.1.4.14.2.2 GetWebResponse

The definition of the **GetWebResponse** element is as follows.

```
<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: Specifies the status of the operation. On successful completion the operation, the protocol server MUST set it to 0.

sWebMetadata: Specifies metadata information about the context site. It is of type **_sWebMetadata**, as defined in section 3.1.4.14.3.2.

vWebs: Specifies information about all the subsites in the context site. It is of type **ArrayOf_sWebWithTime**, as defined in section 2.2.4.5.

vLists: Specifies information about all the lists in the context site. It is of type **ArrayOf_sListWithTime**, as defined in section 3.1.4.14.3.3.

vFPUrls: The protocol server MUST set it to null.

strRoles: 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. **strRoles** is string that contains an XML node specifying information of all the role definition in the site (2), the schema of the **strRoles** is as defined in section 3.1.4.14.2.4.

vRolesUsers: 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. vRolesUsers is of type **ArrayOfString**, as defined in section 2.2.4.6. Each string in the array is an 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 2.2.4.24.

vRolesGroups: 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. **vRolesGroups** is of type **ArrayOfString**, as defined in section 2.2.4.6. Each string in the array is an 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 section 2.2.4.11.

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

3.1.4.14.2.3 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>
```

Permission: Specifies information of a role definition.

Permission.MemberId: Specifies the identification role definition.

Permission.Mask: 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.

Permission.MemberIsUser: MUST be set to **false**.

Permission.MemberGlobal: MUST be set to **false**.

Permission.RoleName: Specifies the name of the role definition.

3.1.4.14.2.4 Roles

The **Roles** element contains array of 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>
```

Role.ID: Specifies the identification of a role definition.

Role.Name: Specifies the name for the role definition.

Role.Description: Specifies description for the role definition.

Role.Order: Specifies the order in which role definitions are displayed in the user interface.

Role.Hidden: A Boolean value that MUST be **true** if, and only if, the role definition is not displayed in the user interface. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

Role.Type: Specifies the type of role definition.

Role.BasePermissions: Specifies the base permissions for the role definition.

3.1.4.14.3   Complex Types

The following XML schema complex type definitions are specific to this operation.

3.1.4.14.3.1   _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: Specifies the GUID of list. This GUID MUST be enclosed in curly braces ({}).

LastModified: Specifies the date and time, as a . locale-dependent short date value and long time value, separated by a space in UTC, when the list was last modified.

IsEmpty: A Boolean value that MUST be **true** if, and only if, the list contains no list items. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

3.1.4.14.3.2   _sWebMetadata

The **_sWebMetadata** complex type contains metadata information for a site (2).

```
<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="0" 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="1" maxOccurs="1" name="IsBucketWeb" type="s:boolean" />
        <s:element minOccurs="1" 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 (2). This GUID MUST be enclosed in curly braces ({}).

Title: Specifies the title of the site (2).

Description: Specifies the description of the site (2).

Author: Specifies the author name of the site (2).

Language: Specifies the LCID of the site (2).

LastModified: Specifies the date and time, as a locale-dependent short date value and long time value, separated by a space in UTC, when the site collection containing the site (2) was last modified.

LastModifiedForceRecrawl: If the site (2) has an external security provider **LastModifiedForceRecrawl** specifies date and time when the site (2) was last modified, otherwise it is DateTime.Min. Format of **LastModifiedForceRecrawl** depends on the implementation of the external security provider.

NoIndex: Specifies whether the site (2) can be crawled for indexing. The value MUST be one of the values described in the following table.

Value	Meaning
true	Do not crawl the site (2) for indexing.
enumerate	Crawl the site (2) for indexing.

ValidSecurityInfo: A Boolean value that MUST be **true** if, and only if, the site (2) does not have an external security provider or the site (2) has an external security provider and the provider has permissions information for the site (2). It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

InheritedSecurity: A Boolean value that MUST be **true** if, and only if, the site (2) inherits security from the parent site (2). It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

AllowAnonymousAccess: A Boolean value that MUST be **true** if, and only if, anonymous users have access permissions for the site (2). It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

AnonymousViewListItems: A Boolean value that MUST be **true** if, and only if, anonymous users can view the list items of the lists of the site (2). It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

Permissions: If the site (2) inherits security from its parent site (2), it specifies the absolute URL of its parent site (2), otherwise it contains permission information of user roles for the site (2). Its schema is defined in section 3.1.4.14.2.3.

ExternalSecurity: A Boolean value that MUST be **true** if, and only if, an external security provider is provided for the site (2). It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4.

CategoryId: Specifies the GUID of the site (2). This GUID MUST be enclosed in curly braces ({}). It MUST be returned only if the site (2) has an external security provider. The protocol client MUST ignore it.

CategoryName: Specifies the name of the site (2). It MUST be returned only if the site (2) has an external security provider. The protocol client MUST ignore it.

CategoryIdPath: A colon-delimited list of the GUIDs of the Web application, all the parent sites (2), and the site (2) itself. It MUST begin with a colon. It MUST be returned only if the site (2) has an external security provider. The protocol client MUST ignore it.

For example, **CategoryIdPath** set to ":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 (2) with GUID "144960A2-F7F9-4E03-B4DE-5A4DB793E900". This site (2) contains a subsite with GUID "3CB4310E-706F-41A5-A85B-FAAFA01A7A8A". This subsite contains a site (2) with GUID "50D959CF-9804-47A3-AE1F-59959B71DFFB".

IsBucketWeb: A Boolean value that MUST be **true** if, and only if, the site (2) is a **bucket Web**. It MUST be returned only if the site (2) has an external security provider. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4. The protocol client MUST ignore it.

UsedInAutocat: A Boolean value that MUST be **true** if, and only if, the site (2) has categories. It MUST be returned only if the site (2) has an external security provider. It MUST be of type **TrueFalseType**, as defined in section 2.2.5.4. The protocol client MUST ignore it.

CategoryBucketID: Specifies the identification of the bucket Web that contains the site (2). It MUST be returned only if the site (2) has an external security provider. The protocol client MUST ignore it.

3.1.4.14.3.3 ArrayOf_sListWithTime

The **ArrayOf_sLisWithTime** complex time contains 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 information of a list. It MUST be of type **_sListWithTime**, as defined in section 3.1.4.14.3.1.

3.1.4.14.4 Simple Types

None.

3.1.4.14.5 Attributes

None.

3.1.4.14.6 Groups

None.

3.1.4.14.7 Attribute Groups

None.

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 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 following occurs:

1. 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/" />
   ```

2. 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
   ```

3. The protocol server returns information about the site collection in a response SOAP message as defined in section 3.1.4.10.2.2.

4. The protocol client parses the response and collects URLs of all the subsites in the site collection.

5. 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.

6. The request SOAP message to call the **GetWeb** operation to get information about the site (2) TestSubSite is as follows.

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

7. 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
   ```

8. The protocol server sends a response SOAP message as defined in section 3.1.4.14.2.2. The response contains all the information about of the subsite "TestSubSite".

9. 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 Indexing

In this scenario, there is a site (2) addressed by the URL http://fabrikam on the protocol server. This site (2) contains two lists: "Documents" and "Contacts". The following occurs.

1. The protocol client calls the **GetListCollection** operation to get information about all the lists in the fabrikam site (2). The protocol client sends the following request SOAP message to the protocol server:

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

2. 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/GetListCollection
```

3. The protocol server returns the following response SOAP message:

```
<GetListCollectionResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <GetListCollectionResult>0</GetListCollectionResult>
  <vLists>
    <_sList>
      <InternalName>{1CDC1A48-695E-4D10-BF84-87946607A456}</InternalName>
      <Title>Contacts</Title>
      <Description>Create a contacts list to manage information about people that your
team works with such as customers or partners.</Description>
      <BaseType>GenericList</BaseType>
      <BaseTemplate>Contacts</BaseTemplate>
      <DefaultViewUrl>/Lists/Contacts/AllItems.aspx</DefaultViewUrl>
      <LastModified>2007-01-09 17:11:57Z</LastModified>
      <InheritedSecurity>true</InheritedSecurity>
      <AllowAnonymousAccess>false</AllowAnonymousAccess>
      <AnonymousViewListItems>false</AnonymousViewListItems>
      <ReadSecurity>1</ReadSecurity>
    </_sList>
    <_sList>
      <InternalName>{0772ECC5-7FE1-4C8D-88BC-A3AFD527BA4D}</InternalName>
      <Title>Documents</Title>
      <Description>This system library was created by the Publishing feature to store
documents that are used on pages in this site.</Description>
      <BaseType>DocumentLibrary</BaseType>
      <BaseTemplate>DocumentLibrary</BaseTemplate>
      <DefaultViewUrl>/Documents/Forms/AllItems.aspx</DefaultViewUrl>
      <LastModified>2008-02-05 18:47:06Z</LastModified>
      <InheritedSecurity>true</InheritedSecurity>
      <AllowAnonymousAccess>false</AllowAnonymousAccess>
      <AnonymousViewListItems>false</AnonymousViewListItems>
      <ReadSecurity>1</ReadSecurity>
    </_sList>
  </vLists>
</GetListCollectionResponse>
```

4. The protocol client parses the response data and retrieves the GUIDs of the lists. The protocol client extracts the **InternalName** of all the lists.

5. The protocol client calls the **GetList** operation to get detailed information about each list.

6. The protocol client calls the **GetList** operation by sending the request SOAP message to obtain information about the list with GUID "1CDC1A48-695E-4D10-BF84-87946607A456" as shown in the following example:

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

7. 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/GetList
```

8. The protocol server returns detailed information of the list as a response SOAP message as defined in section 3.1.4.7.2.2.

9. The protocol client parses the response and updates the index.

4.3 GetListItems

In this scenario, there is a site (2) addressed by the URL http://fabrikam on the protocol server. The site (2) fabrikam contains a document library identified by GUID "8AC68D3D-8A09-4403-8860-D0E494BBE894". The document library contains multiple documents that have a PDF file name 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 following occurs:

1. The 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>
```

2. 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
```

3. The protocol server returns the information about the matching pdf documents in a response SOAP message as defined in section 3.1.4.9.2.2.

4. The protocol client parses the response and collects information about all the documents in the response.

4.4 Incremental Indexing

This example illustrates obtaining incremental changes to a site collection using the **GetChanges** operation.

In this scenario, there is a Web application running on the protocol server. The Web application contains a content database identified by the GUID "690252de-3653-4c4d-93a0-41545e070f46". The content database contains several site collections; one of the site collections is addressed by the URL http://fabrikam. This site collection contains many subsites. One of these subsites is addressed by the URL http://fabrikam/TestSubSite.

This example makes the following assumptions:

- The protocol client has finished a traversal of the site (2) content.

- The protocol client has a **LastChangeId** token in the site (2) change tracking space.

- The contents of the subsite "TestSubSite" has changed since the protocol server issued the **LastChangeId** token. Specifically, one of the documents in the document library list was modified.

The following occurs:

1. The protocol client requests the change report.

2. The protocol client calls the **GetChanges** operation to get detailed change information about the site collection at http://fabrikam. The protocol client sends the following SOAP message to the protocol server to call the **GetChanges** operation:

```
<GetChanges xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <objectType>Site</objectType>
  <contentDatabaseId>690252de-3653-4c4d-93a0-41545e070f46</contentDatabaseId>
  <LastChangeId>1;0;690252de-3653-4c4d-93a0-
41545e070f46;633386167086370000;3815</LastChangeId>
  <CurrentChangeId></CurrentChangeId>
  <Timeout>600</Timeout>
</GetChanges>
```

3. 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/GetChanges
```

4. The protocol server returns the detailed change information in a response SOAP message as defined in section 3.1.4.3.2.2.

5. The response indicates that in the http://fabrikam site collection, in the TestWebSite site (2), in the Documents list, a specific list item has been modified.

6. The protocol client parses the response and updates the indexes. The protocol client can repeatedly call the **GetChanges** operation.

4.5   GetURLSegment

In this scenario, there is a site (2) addressed by the URL http://fabrikam on the protocol server. This site (2) 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 **ID** 1. The absolute URL of the list item is http://fabrikam/Lists/Documents/DispForm.aspx?ID=1. The following occurs:

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.

2. 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 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/GetURLSegments
```

3. 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 which contains the list item Customers.

   ▪The identifier of the list item customers.

5 Security

5.1 Security Considerations for Implementers

The network account used by the protocol client has to belong to the group of site collection administrators to be able to fully retrieve information about site (2) structure, its users and groups, as well as permissions granted to those.

To be able to use the operations related to incremental change tracking, the client network account requires even higher permissions, namely, **Full Read** permissions vs. content database.

Those permissions can be granted to the protocol client by site collection administrator using the **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.

5.2 Index of Security Parameters

None.

6   Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in this appendix.

```xml
<?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:rs="urn:schemas-microsoft-com:rowset" 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: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="0" 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: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="0" 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" />
```

```xml
              <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="1" maxOccurs="1" name="IsBucketWeb" type="s:boolean" />
              <s:element minOccurs="1" 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="UserInformation"/>
          <s:enumeration value="WebPartCatalog"/>
          <s:enumeration value="ListTemplateCatalog"/>
          <s:enumeration value="XMLForm"/>
          <s:enumeration value="MasterPageCatalog"/>
          <s:enumeration value="NoCodeWorkflows"/>
          <s:enumeration value="WorkflowProcess"/>
          <s:enumeration value="WebPageLibrary"/>
          <s:enumeration value="CustomGrid"/>
          <s:enumeration value="DataConnectionLibrary"/>
          <s:enumeration value="WorkflowHistory"/>
          <s:enumeration value="GanttTasks"/>
          <s:enumeration value="Meetings"/>
          <s:enumeration value="Agenda"/>
          <s:enumeration value="MeetingUser"/>
          <s:enumeration value="Decision"/>
```

```xml
          <s:enumeration value="MeetingObjective"/>
          <s:enumeration value="TextBox"/>
          <s:enumeration value="ThingsToBring"/>
          <s:enumeration value="HomePageLibrary"/>
          <s:enumeration value="Posts"/>
          <s:enumeration value="Comments"/>
          <s:enumeration value="Categories"/>
          <s:enumeration value="IssueTracking"/>
          <s:enumeration value="AdminTasks"/>
        </s:restriction>
      </s:simpleType>
      <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">
```

```xml
      <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:element name="GetContent">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="objectType" type="tns:ObjectType" />
          <s:element minOccurs="0" maxOccurs="1" name="objectId" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="folderUrl" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="itemId" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="retrieveChildItems" type="s:boolean"
 />
          <s:element minOccurs="1" maxOccurs="1" name="securityOnly" type="s:boolean" />
          <s:element minOccurs="0" maxOccurs="1" name="lastItemIdOnPage" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:simpleType name="ObjectType">
      <s:restriction base="s:string">
        <s:enumeration value="VirtualServer" />
        <s:enumeration value="ContentDatabase" />
        <s:enumeration value="SiteCollection" />
        <s:enumeration value="Site" />
        <s:enumeration value="List" />
        <s:enumeration value="Folder" />
        <s:enumeration value="ListItem" />
        <s:enumeration value="ListItemAttachments" />
      </s:restriction>
    </s:simpleType>
    <s:element name="GetContentResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="GetContentResult" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="lastItemIdOnPage" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetSiteUrl">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="Url" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetSiteUrlResponse">
      <s:complexType>
```

```xml
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="GetSiteUrlResult"
  type="s:unsignedInt" />
          <s:element minOccurs="1" maxOccurs="1" name="siteUrl" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="siteId" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetChanges">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="objectType" type="tns:ObjectType" />
          <s:element minOccurs="0" maxOccurs="1" name="contentDatabaseId" type="s:string"
  />
          <s:element minOccurs="1" maxOccurs="1" name="LastChangeId" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="CurrentChangeId" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Timeout" type="s:int" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetChangesResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="GetChangesResult" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="LastChangeId" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="CurrentChangeId" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="moreChanges" type="s:boolean" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="ACL ">
      <s:sequence>
        <s:element name="permissions" type="tns:permissionsForACL"/>
      </s:sequence>
    </s:complexType>
    <s:complexType name="ContentDatabaseMetadata">
      <s:attribute name="ChangeId" type="s:string" use="required"/>
      <s:attribute name="ID" type="s:string" use="required"/>
    </s:complexType>
    <s:complexType name="Field">
      <s:attribute name="Name" type="s:string" use="required"/>
      <s:attribute name="Type" type="s:string" use="required"/>
      <s:attribute name="TypeAsString" type="s:string" use="optional"/>
      <s:attribute name="Title" type="s:string" use="required"/>
      <s:attribute name="IsSurveyQuestion" type="tns:TrueFalseType" use="optional"/>
      <s:attribute name="ID" type="s:string" use="required"/>
      <s:attribute name="SourceId" type="s:string" use="required"/>
      <s:attribute name="Indexed" type="s:boolean" use="required"/>
    </s:complexType>
    <s:complexType name="GroupDescription">
      <s:simpleContent>
        <s:extension base="s:string">
          <s:attribute name="ID" type="s:int" use="required"/>
          <s:attribute name="Name" type="s:string" use="required"/>
          <s:attribute name="Description" type="s:string" use="required"/>
          <s:attribute name="OwnerID" type="s:int" use="required"/>
          <s:attribute name="OwnerIsUser" type="tns:TrueFalseType" use="required"/>
        </s:extension>
      </s:simpleContent>
```

```
          </s:complexType>
    <s:complexType name='GroupMembership'>
      <s:sequence>
        <s:element name='Group' maxOccurs='unbounded'>
          <s:complexType>
            <s:sequence>
              <s:element name='Group' type='tns:GroupDescription'/>
              <s:element name='Users' type='tns:Users'/>
            </s:sequence>
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
          <s:complexType name="Groups">
            <s:sequence>
              <s:element minOccurs="0" maxOccurs="unbounded" name="Group"
               type="tns:GroupDescription"/>
            </s:sequence>
          </s:complexType>
          <s:complexType name="ListMetadata">
            <s:attribute name="ID" type="s:string" use="required"/>
            <s:attribute name="Title" type="s:string" use="required"/>
            <s:attribute name="Description" type="s:string" use="required"/>
            <s:attribute name="DefaultViewUrl" type="s:string" use="required"/>
            <s:attribute name="DefaultViewItemUrl" type="s:string" use="required"/>
            <s:attribute name="DefaultTitle" type="s:string" use="required"/>
            <s:attribute name="ScopeID" type="s:string"/>
            <s:attribute name="RootFolder" type="s:string" use="required"/>
            <s:attribute name="ReadSecurity" type="s:int" use="required"/>
            <s:attribute name="NoIndex" type="tns:TrueFalseType" use="required"/>
            <s:attribute name="BaseType" type="tns:ListBaseType" use="required"/>
            <s:attribute name="BaseTemplate" type="tns:ListBaseTemplate" use="required"/>
            <s:attribute name="Author" type="s:string" use="required"/>
            <s:attribute name="AnonymousViewListItems" type="tns:TrueFalseType" use="required"/>
            <s:attribute name="AnonymousPermMask" type="s:unsignedLong" use="required"/>
            <s:attribute name="AnonymousPermMaskRaw" type="s:unsignedLong" use="required"/>
            <s:attribute name="AllowAnonymousAccess" type="tns:TrueFalseType" use="required"/>
            <s:attribute name="LastModified" type="s:string" use="required"/>
            <s:attribute name="ItemCount" type="s:int" use="required"/>
            <s:attribute name="CRC" type="s:int" use="required"/>
          </s:complexType>
          <s:complexType name="permission">
            <s:attribute name="memberid" type="s:int" use="required"/>
            <s:attribute name="mask" type="s:unsignedLong" use="required"/>
          </s:complexType>
          <s:complexType name="SPSiteMetadata">
            <s:attribute name="SiteSubscriptionId" type="s:string"/>
            <s:attribute name="URL" type="s:string" use="required"/>
            <s:attribute name="ID" type="s:string" use="required"/>
            <s:attribute name="LastModified" type="s:string" use="required"/>
            <s:attribute name="PortalURL" type="s:string" use="required"/>
            <s:attribute name="UserProfileGUID" type="s:string" use="required"/>
            <s:attribute name="RootWebId" type="s:string" use="required"/>
            <s:attribute name="ChangeId" type="s:string" use="required"/>
            <s:attribute name="ContentDatabaseId" type="s:string" use="required" />
          </s:complexType>
          <s:complexType name="SPWebAppMetadata">
            <s:attribute name="ID" type="s:string" use="optional"/>
            <s:attribute name="Version" type="s:string" use="optional"/>
```

```xml
          <s:attribute name="URL" type="s:string" use="required"/>
          <s:attribute name="URLZone" type="s:string" use="optional"/>
          <s:attribute name="URLIsHostHeader" type="tns:TrueFalseType" use="optional"/>
      </s:complexType>
      <s:complexType name="Users">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="User"
 type="tns:UserDescription"/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="View">
        <s:attribute name="URL" type="s:string" use="required"/>
        <s:attribute name="Title" type="s:string" use="required"/>
        <s:attribute name="ID" type="s:string" use="required"/>
        <s:attribute name="AllowedInRoot" type="tns:TrueFalseType" use="optional"/>
      </s:complexType>
      <s:complexType name="WebMetadata">
        <s:attribute name="URL" type="s:string" use="required"/>
        <s:attribute name="LastModified" type="s:string" use="required"/>
        <s:attribute name="Created" type="s:string" use="required"/>
        <s:attribute name="ID" type="s:string" use="required"/>
        <s:attribute name="Title" type="s:string" use="required"/>
        <s:attribute name="Description" type="s:string" use="required"/>
        <s:attribute name="LogoURL" type="s:string"/>
        <s:attribute name="LogoDescription" type="s:string"/>
        <s:attribute name="Author" type="s:string" use="required"/>
        <s:attribute name="Language" type="s:unsignedShort" use="required"/>
        <s:attribute name="CRC" type="s:unsignedInt" use="required"/>
        <s:attribute name="NoIndex" type="s:string" use="required"/>
        <s:attribute name="UseFormsForDisplay" type="tns:TrueFalseType" use="optional"/>
        <s:attribute name="DefaultHomePage" type="s:string" use="required"/>
        <s:attribute name="DefaultHomePageListId" type="s:string" use="optional"/>
        <s:attribute name="DefaultHomePageItemId" type="s:integer" use="optional"/>
        <s:attribute name="ExternalSecurity" type="tns:TrueFalseType" use="required"/>
        <s:attribute name="ScopeID" type="s:string"/>
        <s:attribute name="AllowAnonymousAccess" type="tns:TrueFalseType" use="required"/>
        <s:attribute name="AnonymousViewListItems" type="tns:TrueFalseType" use="required"/>
        <s:attribute name="AnonymousPermMask" type="s:unsignedLong" use="required"/>
      </s:complexType>
      <s:complexType name="Webs">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="Web">
            <s:complexType>
              <s:attribute name="URL" type="s:string" use="required"/>
              <s:attribute name="ID" type="s:string" use="required"/>
              <s:attribute name="LastModified" type="s:string" use="required"/>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
      <s:element name="SPContentDatabase">
       <s:complexType>
        <s:sequence>
          <s:element name="ContentDatabase">
            <s:complexType>
              <s:sequence>
                <s:element name="Metadata" type="tns:ContentDatabaseMetadata"/>
              </s:sequence>
            </s:complexType>
```

```
      </s:element>
      <s:element ref="tns:SPSite" minOccurs="0" maxOccurs="unbounded"/>
     </s:sequence>
     <s:attribute name="ItemCount" type="s:int" use="required"/>
     <s:attribute name="Change" type="s:string" use="required"/>
    </s:complexType>
   </s:element>
   <s:element name="SPSite">
    <s:complexType>
     <s:sequence>
     <s:element name="Messages" >
       <s:complexType>
        <s:sequence>
          <s:element name="Message" type="tns:Message" minOccurs="0" maxOccurs="unbounded"
  />
        </s:sequence>
       </s:complexType>
     </s:element>
        <s:element name="Site" minOccurs="0">
         <s:complexType>
          <s:sequence>
            <s:element name="Metadata" type="tns:SPSiteMetadata"/>
            <s:element name="Groups" type="tns:GroupMembership"/>
          </s:sequence>
         </s:complexType>
        </s:element>
        <s:element ref="tns:SPWeb" minOccurs="0" maxOccurs="unbounded"/>
     </s:sequence>
     <s:attribute name="ItemCount" type="s:int" use="required"/>
     <s:attribute name="Change" type="s:string" use="required"/>
     <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" use="optional"/>
     <s:attribute name="Id" type="s:string" use="optional"/>
     <s:attribute name="ParentId" type="s:string" use="optional"/>
     <s:attribute name="InternalUrl" type="s:string" use="optional"/>
     <s:attribute name="DisplayUrl" type="s:string" use="optional"/>
     <s:attribute name="CRC" type="s:int" use="optional"/>
     <s:attribute name="ServerUrl" type="s:string" use="optional"/>
     <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
     <s:attribute name="PresentInFarm" type="tns:TrueFalseType" use="optional"/>
    </s:complexType>
   </s:element>
   <s:complexType name="Message">
    <s:sequence>
     <s:element name="Exception" minOccurs="1" maxOccurs="1" type="s:string" >
     </s:element>
     <s:element name="ChangeInfo" minOccurs="1" maxOccurs="1">
      <s:complexType>
      <s:attribute name="SiteId" type="s:string" use="required" />
      <s:attribute name="WebId" type="s:string" use="optional" />
      <s:attribute name="ListId" type="s:string" use="optional" />
      <s:attribute name="AlertId" type="s:string" use="optional" />
      <s:attribute name="Id" type="s:string" use="optional" />
      <s:attribute name="FieldId" type="s:string" use="optional" />
      <s:attribute name="Url" type="s:string" use="optional" />
      <s:attribute name="DocId" type="s:string" use="optional" />
      <s:attribute name="UserId" type="s:string" use="optional" />
      <s:attribute name="Activate" type="s:boolean" use="optional" />
      <s:attribute name="IsSiteAdminChange" type="s:boolean" use="optional" />
      </s:complexType>
```

```
          </s:element>
        </s:sequence>
        <s:attribute name="ChangeToken" type="s:string" use="required" />
        <s:attribute name="ChangeObject" type="tns:ChangeObject" use="required" />
        <s:attribute name="ChangeType" type="tns:ChangeType" use="required" />
      </s:complexType>
      <s:element name="SPView">
        <s:complexType>
          <s:attribute name="ItemCount" type="s:int" use="required"/>
          <s:attribute name="Change" type="s:string" use="required"/>
          <s:attribute name="Id" type="s:string" use="optional"/>
          <s:attribute name="ServerUrl" type="s:anyURI" use="optional"/>
          <s:attribute name="DisplayUrl" type="s:string" use="optional"/>
          <s:attribute name="ParentId" type="s:string" use="optional"/>
          <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" use="optional"/>
          <s:attribute name="InternalUrl" type="tns:InternalUrl" use="optional"/>
          <s:attribute name="CRC" type="s:int" use="optional"/>
          <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
        </s:complexType>
      </s:element>
      <s:element name="SPWeb">
        <s:complexType>
          <s:sequence>
            <s:element name="Web">
              <s:complexType>
                <s:sequence>
                  <s:element name="Metadata" minOccurs="0" type="tns:WebMetadata"/>
                  <s:element name="Users" minOccurs="0" type="tns:Users"/>
                  <s:element name="ACL" minOccurs="0" type="tns:ACL"/>
                  <s:element name="Webs" minOccurs="0" type="tns:Webs"/>
                  <s:element name="Lists" minOccurs="0" type="tns:Lists"/>
                </s:sequence>
              </s:complexType>
            </s:element>
            <s:choice maxOccurs="unbounded">
              <s:element ref="tns:SPFolder"/>
              <s:element ref="tns:SPList"/>
              <s:element ref="tns:SPFile"/>
            </s:choice>
          </s:sequence>
          <s:attribute name="ItemCount" type="s:int" use="required"/>
          <s:attribute name="Change" type="s:string" use="required"/>
          <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" use="optional"/>
          <s:attribute name="Id" type="s:string" use="optional"/>
          <s:attribute name="ParentId" type="s:string" use="optional"/>
          <s:attribute name="InternalUrl" type="s:string" use="optional"/>
          <s:attribute name="DisplayUrl" type="s:string" use="optional"/>
          <s:attribute name="CRC" type="s:int" use="optional"/>
          <s:attribute name="ServerUrl" type="s:string" use="optional"/>
          <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
        </s:complexType>
      </s:element>
      <s:element name="ContentDatabase">
        <s:complexType>
          <s:sequence>
            <s:element name="Metadata" type="tns:ContentDatabaseMetadata" minOccurs="1"
maxOccurs="1"/>
            <s:element name="Sites" type="tns:Sites" minOccurs="0" maxOccurs="1"/>
          </s:sequence>
```

```
          </s:complexType>
        </s:element>
        <s:element name="FPFolder" type="tns:FolderData"/>
        <s:element name="Item">
          <s:complexType>
            <s:sequence>
              <s:element maxOccurs="unbounded" name="Attachment">
                <s:complexType>
                  <s:attribute name="URL" type="s:string" use="required"/>
                </s:complexType>
              </s:element>
            </s:sequence>
            <s:attribute name="Count" type="s:unsignedInt" use="required"/>
          </s:complexType>
        </s:element>
        <s:element name="List" type="tns:List"/>
        <s:element name="Site">
          <s:complexType>
            <s:sequence>
              <s:element name="Metadata" type="tns:SPSiteMetadata" minOccurs="1"
 maxOccurs="1"/>
              <s:element name="Groups" type="tns:GroupMembership" minOccurs="1" maxOccurs="1"/>
              <s:element name="Web" type="tns:Web" minOccurs="0" maxOccurs="1"/>
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="VirtualServer">
          <s:complexType>
            <s:sequence>
              <s:element name="Metadata" type="tns:SPWebAppMetadata" minOccurs="1"
 maxOccurs="1"/>
              <s:element name="ContentDatabases" type="tns:ContentDatabases" minOccurs="0"
 maxOccurs="1"/>
              <s:element name="Policies" type="tns:Policies" minOccurs="0" maxOccurs="1"/>
              <s:element name="Servers" minOccurs="0">
                <s:complexType>
                  <s:sequence>
                    <s:element name="Server" minOccurs="0 minOccurs="unbounded">
                      <s:complexType>
                        <s:attribute name="URL" type="s:string"/>
                      </s:complexType>
                    </s:element>
                  </s:sequence>
                </s:complexType>
              </s:element>
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="Folder" type="tns:ItemData"/>
        <s:complexType name="Web">
            <s:sequence>
              <s:element name="Metadata" type="tns:WebMetadata" minOccurs="1" maxOccurs="1"/>
              <s:element name="Users" type="tns:Users" minOccurs="1" maxOccurs="1"/>
              <s:element name="ACL" type="tns:ACL" minOccurs="1" maxOccurs="1"/>
              <s:element name="Webs" type="tns:Webs" minOccurs="1" maxOccurs="1"/>
              <s:element name="Lists" type="tns:Lists" minOccurs="1" maxOccurs="1"/>
              <s:element name="FPFolder" type="tns:FolderData" minOccurs="0" maxOccurs="1"/>
            </s:sequence>
        </s:complexType>
```

```xml
<s:complexType name="ContentDatabases">
  <s:sequence>
    <s:element maxOccurs="unbounded" name="ContentDatabase">
      <s:complexType>
        <s:attribute name="ID" type="s:string" use="required"/>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
<s:complexType name="Files">
  <s:sequence>
    <s:element name="File" minOccurs="0" maxOccurs="unbounded">
      <s:complexType>
        <s:attribute name="URL" type="s:string" use="required"/>
        <s:attribute name="ID" type="s:string" use="required"/>
        <s:attribute name="LastModified" type="s:string" use="required"/>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
<s:complexType name="FolderData">
  <s:sequence>
    <s:element name="Folders" type="tns:Folders" minOccurs="0" maxOccurs="unbounded"/>
    <s:element name="Files" type="tns:Files" minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
<s:complexType name="Folders">
  <s:sequence>
    <s:element name="Folder" maxOccurs="unbounded">
      <s:complexType>
        <s:attribute name="URL" type="s:string" use="required"/>
        <s:attribute name="ID" type="s:string" use="required"/>
        <s:attribute name="LastModified" type="s:string" use="required"/>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
<s:complexType name="ItemData">
  <s:sequence>
    <s:element name="Metadata" type="tns:Metadata"/>
    <s:element name="xml" type="xml"/>
  </s:sequence>
</s:complexType>
<s:complexType name="Policies">
  <s:sequence>
    <s:element maxOccurs="unbounded" name="PolicyUser" type="tns:PolicyUser"/>
  </s:sequence>
  <s:attribute name="AnonymousGrantMask" type="s:unsignedLong" use="required"/>
  <s:attribute name="AnonymousDenyMask" type="s:unsignedLong" use="required"/>
</s:complexType>
<s:complexType name="PolicyUser">
  <s:attribute name="LoginName" type="s:string" use="required"/>
  <s:attribute name="Sid" type="s:string" use="required"/>
  <s:attribute name="GrantMask" type="s:unsignedLong" use="required"/>
  <s:attribute name="DenyMask" type="s:unsignedLong" use="required"/>
  <s:attribute name="BinaryIdentifier" type="s:string" use="optional" />
  <s:attribute name="BinaryIdentifierType" type="s:string" use="optional" />
</s:complexType>
<s:complexType name="Sites">
```

```
        <s:sequence>
          <s:element maxOccurs="unbounded" name="Site">
            <s:complexType>
              <s:attribute name="URL" type="s:string" use="required"/>
              <s:attribute name="ID" type="s:string" use="required"/>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
      <s:element name="GetChangesEx">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="version" type="s:int"/>
            <s:element minOccurs="1" maxOccurs="1" name="xmlInput" type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetChangesExRequest">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="ObjectType" type="tns:ObjectType"/>
            <s:element minOccurs="1" maxOccurs="1" name="ContentDatabaseId" type="s:string"/>
            <s:element minOccurs="1" maxOccurs="1" name="StartChangeId" type="s:string"/>
            <s:element minOccurs="0" maxOccurs="1" name="EndChangeId" type="s:string"/>
        <s:element minOccurs="0" maxOccurs="1" name="IgnoreSecurityIfInherit" type="s:boolean"
  />
        <s:element minOccurs="0" maxOccurs="1" name="GetMetadata" type="s:boolean" />
        <s:element minOccurs="0" maxOccurs="1" name="RequestLoad" type="s:int" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetChangesExResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetChangesExResult"
             type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="Permissions">
        <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="s: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:element name="Groups" type="tns:Groups"/>
      <s:element name="Users" type="tns:Users"/>
      <s:element name="Permissions">
        <s:complexType>
          <s:sequence>
            <s:element maxOccurs="unbounded" name="Permission">
              <s:complexType>
```

```xml
                  <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:simpleType name="TrueFalseType">
          <s:restriction base="s:string">
          <s:enumeration value="True"/>
          <s:enumeration value="False"/>
          </s:restriction>
        </s:simpleType>
    <s:simpleType name="ChangeType"  >
          <s:restriction base="s:string">
            <s:enumeration value="NoChange"/>
            <s:enumeration value="Add"/>
            <s:enumeration value="Update"/>
            <s:enumeration value="Delete"/>
            <s:enumeration value="Rename"/>
            <s:enumeration value="MoveAway"/>
            <s:enumeration value="MoveInto"/>
            <s:enumeration value="Restore"/>
            <s:enumeration value="RoleAdd"/>
            <s:enumeration value="RoleDelete"/>
            <s:enumeration value="RoleUpdate"/>
            <s:enumeration value="AssignmentAdd"/>
            <s:enumeration value="AssignmentDelete"/>
            <s:enumeration value="MemberAdd"/>
            <s:enumeration value="MemberDelete"/>
            <s:enumeration value="SystemUpdate"/>
            <s:enumeration value="Navigation"/>
          </s:restriction>
        </s:simpleType>

        <s:simpleType name="ChangeObject"  >
          <s:restriction base="s:string">
            <s:enumeration value="Microsoft.SharePoint.SPChangeItem"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeSite"/>
```

```
            <s:enumeration value="Microsoft.SharePoint.SPChangeWeb"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeList"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeFile"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeFolder"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeAlert"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeUser"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeGroup"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeContentType"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeField"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeSecurityPolicy"/>
            <s:enumeration value="Microsoft.SharePoint.SPChangeView"/>
        </s:restriction>
    </s:simpleType>

    <s:complexType name="Metadata">
      <s:sequence>
        <s:element name="scope">
          <s:complexType>
            <s:sequence>
              <s:element name="permissions" type="tns:permissionsForACL" />
            </s:sequence>
            <s:attribute name="id" type="s:string" use="required" />
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>

    <s:complexType name='permissionsForACL'>
      <s:sequence >
        <s:element name='permission' type='tns:permission' maxOccurs='unbounded' />
      </s:sequence>
    </s:complexType>

    <s:complexType name='UserDescription'>
      <s:attribute name='Sid' type='s:string' use='required'/>
      <s:attribute name='Name' type='s:string' use='required' />
      <s:attribute name='LoginName' type='s:string' use='required' />
      <s:attribute name='IsDomainGroup' type= 'tns:TrueFalseType'
       use='required'/>
      <s:attribute name='Email' type='s:string' use='optional'/>
      <s:attribute name='Notes' type='s:string' use='required' />
      <s:attribute name='ID' type='s:int' use='required' />
      <s:attribute name='IsSiteAdmin' type= 'tns:TrueFalseType' use='required'/>
    </s:complexType>

    <s:complexType name="Lists">
      <s:sequence>
        <s:element maxOccurs="unbounded" name="List" >
          <s:complexType>
            <s:attribute name="ID" type="s:string" use="required" />
            <s:attribute name="LastModified" type="s:string" use="required" />
            <s:attribute name="DefaultViewUrl" type="s:string" use="required" />
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>

    <s:element name='SPFile'>
      <s:complexType>
```

```
            <s:attribute name='ItemCount' type='s:int' use='required'/>
            <s:attribute name='Change' type='s:string' use='required'/>
            <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" />
            <s:attribute name="Id" type="s:string" />
            <s:attribute name="ParentId" type="s:string" />
            <s:attribute name="InternalUrl" type="s:string" />
            <s:attribute name="DisplayUrl" type="s:string" />
            <s:attribute name="Url" type="s:string" />
            <s:attribute name="CRC" type="s:int" />
            <s:attribute name="ServerUrl" type="s:string" />
            <s:attribute name="SequenceNumber" type="s:string"/>
        </s:complexType>
    </s:element>

    <s:element name='SPFolder'>
        <s:complexType>
            <s:attribute name='ItemCount' type='s:int' use='required'/>
            <s:attribute name='Change' type='s:string' use='required'/>
            <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" />
            <s:attribute name="Id" type="s:string" />
            <s:attribute name="ParentId" type="s:string" />
            <s:attribute name="InternalUrl" type="s:string" />
            <s:attribute name="DisplayUrl" type="s:string" />
            <s:attribute name="Url" type="s:string" />
            <s:attribute name="CRC" type="s:int" />
            <s:attribute name="ServerUrl" type="s:string" />
            <s:attribute name="SequenceNumber" type="s:string"/>
        </s:complexType>
    </s:element>

    <s:element name='SPList'>
        <s:complexType>
            <s:sequence >
                <s:element ref='tns:List' minOccurs='0' />
                <s:choice minOccurs='0' maxOccurs='unbounded'>
                    <s:element ref='tns:SPView' minOccurs='0' maxOccurs='unbounded'/>
                    <s:element ref='tns:SPListItem' minOccurs='0' maxOccurs='unbounded'/>
                </s:choice>
            </s:sequence>
            <s:attribute name='ItemCount' type='s:int' use='required'/>
            <s:attribute name='Change' type='s:string' use='required'/>
            <s:attribute name="UpdateSecurity" type="tns:TrueFalseType" />
            <s:attribute name="Id" type="s:string" />
            <s:attribute name="ParentId" type="s:string" />
            <s:attribute name="InternalUrl" type="s:string" />
            <s:attribute name="DisplayUrl" type="s:string" />
            <s:attribute name="CRC" type="s:int" />
            <s:attribute name="ServerUrl" type="s:string" />
            <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
        </s:complexType>
    </s:element>

    <s:complexType name="List">
        <s:sequence>
            <s:element name="Metadata" type="tns:ListMetadata"/>
            <s:element name="ACL" type="tns:ACL" >

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

```
              <s:element name="Schema" type="tns:Schema"/>
           </s:sequence>
        </s:complexType>

        <s:complexType name='Views'>
           <s:sequence >
              <s:element name='View' type='tns:View' maxOccurs='unbounded'/>
           </s:sequence>
        </s:complexType>

        <s:complexType name='Schema'>
           <s:sequence >
              <s:element name='Field' type='tns:Field' maxOccurs='unbounded' />
           </s:sequence>
        </s:complexType>

        <s:element name='SPListItem'>
           <s:complexType>
              <s:sequence>
                 <s:element name='ListItem' >
                    <s:complexType >
                       <s:sequence >
                          <s:element name='row' type='s:string' />
                          <s:element name='permissions' type='tns:permissionsForACL' />
                       </s:sequence>
                    </s:complexType>
                 </s:element>
              </s:sequence>
              <s:attribute name="Change" type="s:string" use="required"/>
              <s:attribute name="ItemCount" type="s:int" use="required" />
              <s:attribute name="UpdateSecurity" type="tns:TrueFalseType"
               use="optional" />
              <s:attribute name="ServerUrl" type="s:anyURI" use="optional" />
              <s:attribute name="ParentId" type="s:string" use="optional" />
              <s:attribute name="InternalUrl" type="s:string" use="optional"/>
              <s:attribute name="Id" type="s:string" use="optional" />
              <s:attribute name="DisplayUrl" type="s:string" use="optional" />
              <s:attribute name="CRC" type="s:unsignedInt" use="optional"/>
              <s:attribute name="Url" type="s:int" use="optional" />
              <s:attribute name="SequenceNumber" type="s:string" use="optional"/>
           </s:complexType>
        </s:element>
     </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:message name="GetContentSoapIn">
  <wsdl:part name="parameters" element="tns:GetContent" />
</wsdl:message>
<wsdl:message name="GetContentSoapOut">
  <wsdl:part name="parameters" element="tns:GetContentResponse" />
</wsdl:message>
<wsdl:message name="GetSiteUrlSoapIn">
  <wsdl:part name="parameters" element="tns:GetSiteUrl" />
</wsdl:message>
<wsdl:message name="GetSiteUrlSoapOut">
  <wsdl:part name="parameters" element="tns:GetSiteUrlResponse" />
</wsdl:message>
<wsdl:message name="GetChangesSoapIn">
  <wsdl:part name="parameters" element="tns:GetChanges" />
</wsdl:message>
<wsdl:message name="GetChangesSoapOut">
  <wsdl:part name="parameters" element="tns:GetChangesResponse" />
</wsdl:message>
<wsdl:message name="GetChangesExSoapIn">
  <wsdl:part name="parameters" element="tns:GetChangesEx" />
```

```
    </wsdl:message>
    <wsdl:message name="GetChangesExSoapOut">
      <wsdl:part name="parameters" element="tns:GetChangesExResponse" />
    </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:operation name="GetContent">
        <wsdl:input message="tns:GetContentSoapIn" />
        <wsdl:output message="tns:GetContentSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="GetSiteUrl">
        <wsdl:input message="tns:GetSiteUrlSoapIn" />
        <wsdl:output message="tns:GetSiteUrlSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="GetChanges">
        <wsdl:input message="tns:GetChangesSoapIn" />
        <wsdl:output message="tns:GetChangesSoapOut" />
      </wsdl:operation>
        <wsdl:operation name="GetChangesEx">
          <wsdl:input message="tns:GetChangesExSoapIn" />
          <wsdl:output message="tns:GetChangesExSoapOut" />
        </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:operation name="GetContent">
        <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetContent"
 style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetSiteUrl">
        <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSiteUrl"
 style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetChanges">
        <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetChanges"
 style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
```

```
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetChangesEx">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetChangesEx"
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:operation name="GetContent">
        <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetContent"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
```

```
      </wsdl:operation>
      <wsdl:operation name="GetSiteUrl">
        <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSiteUrl"
  style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetChanges">
        <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetChanges"
  style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetChangesEx">
        <soap12:operation
  soapAction="http://schemas.microsoft.com/sharepoint/soap/GetChangesEx" 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: Site Data Extension to ADO XML Persistence Format

ADO XML Persistence Format, as described in [MS-PRSTFR], is open, which means that additional elements and attributes can be added by sender (of course, the receiver has to be aware of those extensions). This protocol does extend the data type namespace (xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882").

Scopes element

By default, ADO XML Persistence Format contains the following elements:

```
<s:complexType name="xml">
  <s:sequence>
    <s:element ref="xdr:Schema"/>
    <s:element ref="rs:data"/>
  </s:sequence>
</s:complexType>
```

This protocol extends this group by adding the **scopes** element:

```
<s:complexType name="xml">
  <s:sequence>
    <s:element ref="xdr:Schema"/>
    <s:element name="scopes">
      <s:complexType>
        <s:sequence>
          <s:element name="scope" minOccurs="0" maxOccurs="unbounded">
            <s:complexType>
              <s:sequence>
                <s:element name="permission" type="tns:permission"/>
              </s:sequence>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element ref="rs:data"/>
  </s:sequence>
</s:complexType>
```

scope.id: Specifies the GUID of the collection of role assignments of the list or the folder of the list.

scope.permission: Specifies permission information, as defined in section 2.2.4.16.

Lookup fields

Originally, ADO XML Persistence Format contains the following attribute group:

```
<s:attributeGroup name="typeAttributes">
    <s:attribute ref="dt:type" />
    <s:attribute ref="dt:values"/>
    <s:attribute ref="dt:minLength"/>
    <s:attribute ref="dt:maxLength"/>
  </s:attributeGroup>
```

This protocol extends this group by adding **dt:lookup** attribute:

```
<s:attributeGroup name="typeAttributes">
     <s:attribute ref="dt:type" />
     <s:attribute ref="dt:values"/>
     <s:attribute ref="dt:minLength"/>
     <s:attribute ref="dt:maxLength"/>
     <s:attribute ref="dt:lookup"/>
  </s:attributeGroup>
```

The lookup attribute is defined as follows.

```
<s:attribute name="lookup" type="s:boolean"/>
```

Its value has to be interpreted as follows. If it is **true** (by default, it is **false**) the attribute specifies that the data includes a lookup prefix. The lookup prefix specifies the identification of the list item. When lookup prefix is present, it is separated from the actual data by using the semicolon number sign (;#) lookup prefix separator. For example, **LastModified** field (2) is defined in an XDR schema as follows.

```
<s:AttributeType  name="ows_Last_x0020_Modified"
                  rs:name="Modified"
                  rs:number="32">
    <s:datatype  dt:type="datetime"
                 dt:lookup="true"
                 dt:maxLength="8">
    </s:datatype>
</s:AttributeType>
```

In the **rs:data** section, the field data will contain the following.

```
 ows_Last_x0020_Modified="3;#2008-01-12T00:26:20Z"
```

The lookup prefix here is "3;#". The implementer has to remove the prefix before interpreting the field value, in this case, UTC time **2008-01-12T00:26:20Z** (with prefix, it will not be a valid UTC, as prescribed by dt:type="datetime" attribute in the schema).

The lookup prefix cannot be considered as part of the data value. For example, if the **Author** field is shown in XDR schema as follows.

```
<s:AttributeType  name="ows_Author"
                  rs:name="Created By"
                  rs:number="13">
     <s:datatype  dt:type="string"
                  dt:lookup="true"
                  dt:maxLength="512">
     </s:datatype>
</s:AttributeType>
```

The field data might be as follows.

```
   ows_Author="12;#Nancy Anderson"
```

This is to be interpreted as "Author is Nancy Anderson", not as "Author is 12;#Nancy Anderson".
Values provided to **Query** criteria (see section 3.1.4.9.3.1) are to comply with this rule.

Note that the fields described in section 3.1.4.7 schema as having `<Type>Lookup</Type>` can
contain multiple values, separated by the same `;#` lookup prefix separator. For example, if a
Lookup field refers to various colors, possible field data might be as follows.

```
   '2;#Green;#3;#Blue"
```

Stripped of lookup prefixes and lookup prefix separators, actual field value is to be interpreted as
containing "Green" and "Blue".

This protocol does not define an escape prefix for the lookup prefix separator.

Binary fields

It is possible that a binary value might be defined in an XDR schema as an **int** with a big length,
rather than as **bin.hex**.

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

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 3.0

- Microsoft SharePoint Foundation 2010

- Microsoft SharePoint Foundation 2013

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

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

<1> Section 2.2.4: This type is available only in SharePoint Foundation 2013.

<2> Section 2.2.4: This type is available only in SharePoint Foundation 2013.

<3> Section 2.2.4.8: This attribute is only available in SharePoint Foundation 2013.

<4> Section 2.2.4.12: This type is available only in SharePoint Foundation 2013.

<5> Section 2.2.4.14: Windows SharePoint Services 3.0 does not support this attribute.

<6> Section 2.2.4.18: This type is available only in SharePoint Foundation 2013.

<7> Section 2.2.4.21: Windows SharePoint Services 3.0 does not contain the **SiteSubscriptionId** attribute.

<8> Section 2.2.4.21: Personal sites are supported in Office SharePoint Server 2007 and SharePoint Server 2010. They are not supported in Windows SharePoint Services 3.0 and SharePoint Foundation 2010.

<9> Section 2.2.4.22: **ID** attribute is required for SharePoint Foundation 2010, but not supported by Windows SharePoint Services 3.0.

<10> Section 2.2.4.22: **Version** attribute is required for SharePoint Foundation 2010, but not supported by Windows SharePoint Services 3.0.

<11> Section 2.2.4.22: **URLZone** attribute is required for SharePoint Foundation 2010, but not supported by Windows SharePoint Services 3.0.

<12> Section 2.2.4.22: **URLIsHostHeader** attribute is required for SharePoint Foundation 2010, but not supported by Windows SharePoint Services 3.0.

<13> Section 2.2.4.23: Windows SharePoint Services 3.0 does not support this attribute.

<14> Section 2.2.4.25: Windows SharePoint Services 3.0 does not support the **AllowedInRoot** attribute.

<15> Section 2.2.4.27: Windows SharePoint Services 3.0 does not return the **LogoURL** attribute.

<16> Section 2.2.4.27: Windows SharePoint Services 3.0 does not return the **LogoDescription** attribute.

<17> Section 2.2.4.27: This attribute is available only in SharePoint Foundation 2013.

<18> Section 3.1.3: For this protocol, SharePoint Foundation 2010 and Windows SharePoint Services 3.0 servers use for the port with standard address location being http://root/_vti_bin/SiteData.asmx where **root** denotes a root URL of a site (2) (or some subsite thereof).

<19> Section 3.1.4.2.2.2: In SharePoint Foundation 2010, the error string for this condition is "List does not exist. The page you selected contains a list that does not exist. It may have been deleted by another user." The error code is 0x82000006.

<20> Section 3.1.4.2.2.2: In SharePoint Foundation 2010, the error string for this condition is "Item does not exist. It may have been deleted by another user."

<21> Section 3.1.4.2.2.2: In SharePoint Foundation 2010, the error string for this condition is "List does not exist. The page you selected contains a list that does not exist. It may have been deleted by another user." The error code is 0x82000006.

<22> Section 3.1.4.3.2.4: In Windows SharePoint Services 3.0, if the file (attachment) is inside a list, the server will always set this element to "unchanged". In SharePoint Foundation 2010, if the file (attachment) is inside a list and the change is not deleting, the server will always set this element to "unchanged".

<23> Section 3.1.4.3.2.8: SharePoint Foundation 2010 Service Pack 1 provides information about whether the site collection is present in the farm in the delete change notification. Windows SharePoint Services 3.0 and other versions of SharePoint Foundation 2010 do not provide the present in farm information in the change notification.

<24> Section 3.1.4.3.2.9: SharePoint Foundation 2010 returns the **Message** element only when the change could not be properly processed. Unlike SharePoint Foundation 2010, **Message** will appear in **GetChangesResponse** even when there is no exception during change processing for Windows SharePoint Services 3.0.

<25> Section 3.1.4.3.2.9: **Message** from Windows SharePoint Services 3.0 does not follow this XML schema. Instead, it is represented with a string containing tokens delimited with spaces. Below is an example of the **Message** value:

1;1;f9945c20-f2d4-4aea-be7e-1d89a73efe2d;634190155384170000;125328 Microsoft.SharePoint.SPChangeItem Add Exception Specified method is not supported.

The first token represents the **ChangeToken** attribute of the **Message** element. The second token represents the **ChangeObject** attribute of the **Message** element. The third token represents the **ChangeType** attribute. The first three tokens are required. There can be multiple additional tokens in the string after the three required tokens. They are:

•"*RETRY ATTEMPT*": This means process the previous change over again.

•"SiteAdminChange": This is a site admin change notification. Only present when the change is updating the site admin user.

•"Unknown change type": The change type is unknown.

•"Done": The change has been processed successfully.

•"Exception": There is an exception during the change processing. Detailed exception message follows.

<26> Section 3.1.4.3.5: Windows SharePoint Services 3.0 does not support the **SequenceNumber** attribute.

<27> Section 3.1.4.3.5.1: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 do not return this value for this condition.

<28> Section 3.1.4.3.5.1: Windows SharePoint Services 3.0 does not support the **UpdateSecurityScope** value for the **change** attribute in **GetChanges**.

<29> Section 3.1.4.3.5.11: SharePoint Foundation 2010 always sets this value to "False".

<30> Section 3.1.4.3.5.13: This attribute is available only in SharePoint Foundation 2013.

<31> Section 3.1.4.4.2.3.6: This behavior applies only to SharePoint Foundation 2013.

<32> Section 3.1.4.4.5.1: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 both support **UpdateSecurityScope** value for the **change** attribute in **GetChangesEx**.

<33> Section 3.1.4.5.2.9: Windows SharePoint Services 3.0 does not support the **Servers** element.

<34> Section 3.1.4.6: This operation is available only in SharePoint Foundation 2013.

<35> Section 3.1.4.10.3.1: Personal sites are supported in Office SharePoint Server 2007 and SharePoint Server 2010. They are not supported in Windows SharePoint Services 3.0 and SharePoint Foundation 2010.

<36> Section 3.1.4.11.2.2: SharePoint Foundation 2010 returns the error message "Value cannot be null. Parameter name: *strUrl*".

<37> Section 3.1.4.13: In SharePoint Foundation 2010, the string comparison is case insensitive. In Windows SharePoint Services 3.0, the string comparison is case sensitive.

<38> Section 3.1.4.13: In SharePoint Foundation 2010, the string comparison is case insensitive. In Windows SharePoint Services 3.0, the string comparison is case sensitive.

<39> Section 3.1.4.13: In SharePoint Foundation 2010, the string comparison is case insensitive. In Windows SharePoint Services 3.0, the string comparison is case sensitive.

<40> Section 3.1.4.13: In SharePoint Foundation 2010, the string comparison is case insensitive. In Windows SharePoint Services 3.0, the string comparison is case sensitive.

9   Change Tracking

This section identifies changes that were made to the [MS-SITEDATS] protocol document between the October 2012 and February 2013 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.

- An extensive rewrite, addition, or deletion of major portions of content.

- The removal of a document from the documentation set.

- Changes made for template compliance.

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

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

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

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

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- New content added for template compliance.

- Content updated for template compliance.

- Content removed for template compliance.

- Obsolete document removed.

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

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

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.3.1 Site Structure	Added the number for the glossary item "lists".	N	Content updated.
2.2.4 Complex Types	Updated the description for the Lists complex type.	N	Content updated.
2.2.4.8 Field	Added a definition number for the term "field".	N	Content updated.
2.2.4.13 List	Added a definition number for the term "list".	N	Content updated.
3.1.1 Abstract Data Model	Added definition numbers for the terms "field" and "list".	N	Content updated.

10 Index