[MS-SITESS]:

Sites Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Support. For questions and support, please contact dochelp@microsoft.com.

Preliminary Documentation. This particular Open Specifications document provides documentation for past and current releases and/or for the pre-release version of this technology. This document provides final documentation for past and current releases and preliminary documentation, as applicable and specifically noted in this document, for the pre-release version. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. Because this documentation might change between the pre-release version and the final

version of this technology, there are risks in relying on this preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
6/27/2008	1.0	Major	Revised and edited the technical content
10/6/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
7/13/2009	1.03	Major	Revised and edited the technical content
8/28/2009	1.04	Editorial	Revised and edited the technical content
11/6/2009	1.05	Editorial	Revised and edited the technical content
2/19/2010	2.0	Major	Updated and revised the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Minor	Clarified the meaning of the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	Major	Significantly changed the technical content.
3/18/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	2.6	Minor	Clarified the meaning of the technical content.
4/11/2012	2.6	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.6	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	2.6	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.7	Minor	Clarified the meaning of the technical content.
2/11/2013	2.8	Minor	Clarified the meaning of the technical content.
7/30/2013	2.9	Minor	Clarified the meaning of the technical content.
11/18/2013	2.9	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
2/10/2014	2.9	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.9	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	2.10	Minor	Clarified the meaning of the technical content.
10/30/2014	2.10	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	3.0	Major	Significantly changed the technical content.
7/15/2016	3.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	3.0	None	No changes to the meaning, language, or formatting of the technical content.
12/15/2016	3.1	Minor	Clarified the meaning of the technical content.
7/24/2018	4.0	Major	Significantly changed the technical content.
10/1/2018	5.0	Major	Significantly changed the technical content.
12/11/2018	5.1	Minor	Clarified the meaning of the technical content.
3/19/2019	5.2	Minor	Clarified the meaning of the technical content.
6/18/2019	5.2	None	No changes to the meaning, language, or formatting of the technical content.
2/16/2021	5.3	Minor	Clarified the meaning of the technical content.
7/20/2021	6.0	Major	Significantly changed the technical content.

Table of Contents

[MS-SITESS] - v20210720
Sites Web Service Protocol
Copyright © 2021 Microsoft Corporation
Release: July 20, 2021

1   Introduction

This document specifies the Sites Web Service Protocol. This protocol enables a protocol client to execute tasks on a **site collection**.

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

1.1   Glossary

This document uses the following terms:

access control list (ACL): A list of access control entries (ACEs) that collectively describe the security rules for authorizing access to some resource; for example, an object or set of objects.

anonymous authentication: An authentication mode in which neither party verifies the identity of the other party.

available site template: An XML-based collection of predefined or user-defined settings that are stored as a site definition configuration or a site template, and can be used when creating a site.

blank site: A site that was created by using the "Blank" site template.

collation order: A rule for establishing a sequence for textual information.

content migration package: A package of XML-formatted files that is used to migrate content between site collections, sites, and lists.

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

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

custom action: An extension to the user interface, such as a button on a toolbar or a link on a site settings page.

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

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

farm: A group of computers that work together as a single system to help ensure that applications and resources are available. Also referred to as server farm.

field: A container for metadata within a SharePoint list and associated list items.

file: A single, discrete unit of content.

form: A structured document with controls and spaces that are reserved for entering and displaying information. Forms can contain special coding for actions such as submitting and querying data.

form digest validation: A type of security validation that helps prevent an attack wherein users are tricked into posting data to a server.

fully qualified URL: A URL that includes a protocol scheme name, a host name, optionally a port number, a path, optionally a search part, and optionally a fragment identifier, as described in [RFC2616].

Hypertext Markup Language (HTML): An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [HTML].

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

language code identifier (LCID): A 32-bit number that identifies the user interface human language dialect or variation that is supported by an application or a client computer.

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

membership group: See group.

page: A file that consists of HTML and can include references to graphics, scripts, or dynamic content such as Web Parts.

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

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

presence: A setting for the User field that determines whether instant-messaging status information appears with user names in that field.

security group: A named group of principals on a SharePoint site.

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

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

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

site collection identifier: A GUID that identifies a site collection. In stored procedures, the identifier is typically "@SiteId" or "@WebSiteId". In databases, the identifier is typically "SiteId/tp_SiteId".

site definition: A family of site definition configurations. Each site definition specifies a name and contains a list of associated site definition configurations.

site definition configuration: An XML-based definition of lists, features, modules, and other data that collectively define a type of SharePoint site. Site definition configurations are stored in the ONET.xml file.

site solution: A deployable, reusable package that contains a set of features, site definitions, and assemblies that apply to sites, and can be enabled or disabled individually.

site template: An XML-based definition of site settings, including formatting, lists, views, and elements such as text, graphics, page layout, and styles. Site templates are stored in .stp files in the content database.

site-collection relative URL: A URL that is relative to the site collection that contains a resource, and does not begin with a leading slash (/).

site-relative URL: A URL that is relative to the site that contains a resource and does not begin with a leading slash (/).

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

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

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

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

solution gallery: A gallery that is used to store solution packages.

solution package: A compressed file that can be deployed to a server farm or a site. It can contain assemblies, resource files, site and feature definitions, templates, code access security policies, and Web Parts. Solution packages have a .wsp file name extension.

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

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

user code: Managed code that can be uploaded to a site by a site collection administrator, without approval from the server farm administrator. It cannot access code or data on other site collections.

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

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

workflow: An automation of business processes that passes business documents and tasks automatically from one user to another for action, according to a defined sequence.

workflow template: A definition of operations, the sequence of operations, constraints, and timing for a specific process.

WSDL message: An abstract, typed definition of the data that is communicated during a **WSDL operation** [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

WSDL operation: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

XML attribute: A name/value pair, separated by an equal sign (=) and included in a tagged element, that modifies features of an element. All XML attribute values are stored as strings enclosed in quotation marks.

XML element: An XML structure that typically consists of a start tag, an end tag, and the information between those tags. Elements can have attributes and can contain other elements.

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

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

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2   References

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

1.2.1   Normative References

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

[MS-CSOM] Microsoft Corporation, "SharePoint Client Query Protocol".

[MS-PRIMEPF] Microsoft Corporation, "Deployment Package Format".

[MS-WSSFO3] Microsoft Corporation, "Windows SharePoint Services (WSS): File Operations Database Communications Version 3 Protocol".

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

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

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[SOAP1.2-2/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 2: Adjuncts (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part2-20070427

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

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

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

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

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

1.2.2 Informative References

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

1.3 Protocol Overview (Synopsis)

This protocol provides methods to create and delete **sites** on the server, export and import sites, retrieve **available site template** information, and obtain a **form digest validation** value for authentication of a secure submit request to the server.

Each method in the protocol is a **Simple Object Access Protocol (SOAP)** operation which accepts a set of parameters as a SOAP request and returns a set of values as a SOAP response.

The client sends a request to the server via a SOAP request message and the server sends return values to the client via a SOAP response message, as described in the following figure. All SOAP requests are made to one of several well-defined **Uniform Resource Locators (URLs)** on the server, which can be discovered by clients. The server never initiates any communication with the client.

The following diagram shows the sequence of events.
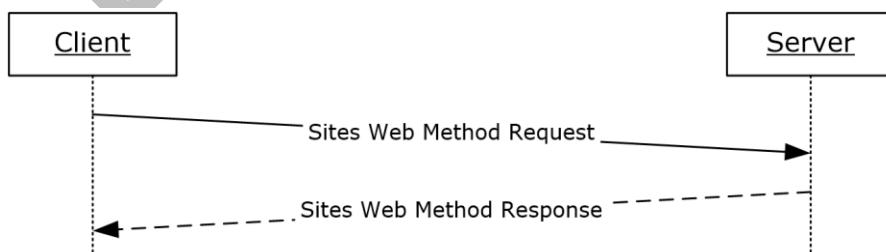
Figure 1: Sites Web method sequence diagram

1.4 Relationship to Other Protocols

This protocol uses the **SOAP** message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2-1/2007] and [SOAP1.2-2/2007]. 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:

Sites Web Service	This Protocol
SOAP	
HTTP | HTTPS	Industry Standard
TCP	
IP	
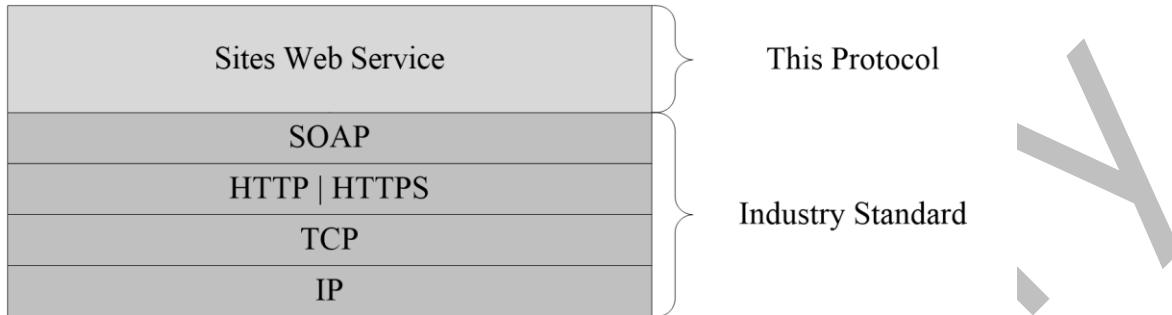
Figure 2: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

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

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

1.6 Applicability Statement

This protocol is applicable in the following scenarios:

- Exporting and importing sites on the server for purposes of migrating, backing up, or restoring a **site**.

- Creating and deleting a site.

- Retrieving **available site template** information.

- Obtaining a **form digest validation** value for the authentication of a secure submit request to a server.

1.7 Versioning and Capability Negotiation

This protocol uses multiple transports with SOAP as specified 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 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 structures 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 specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular **XML namespace prefix** is implementation-specific and not significant for interoperability. These namespaces are described in the following table.

Prefix	Namespace URI	Reference
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2-1/2007] [SOAP1.2-2/2007]
tns	http://schemas.microsoft.com/sharepoint/soap/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
(none)	http://schemas.microsoft.com/sharepoint/soap/	

2.2.2 Messages

None.

2.2.3 Elements

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

2.2.4 Complex Types

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

Complex Type	Description
SOAPFaultDetails	Specifies the details of a **SOAP fault**.
ArrayOfString	Specifies an array of string elements.

2.2.4.1 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"/>
      </s:sequence>
   </s:complexType>
</s:schema>
```

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

errorcode: The hexadecimal representation of a four-byte result code. This element is optional.

2.2.4.2 ArrayOfString

The **ArrayOfString** complex type contains an array of string elements.

```
<s:complexType name="ArrayOfString">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="unbounded" name="string" nillable="true"
               type="s:string" />
  </s:sequence>
</s:complexType>
```

string: A file full path or **URL**.

2.2.5 Simple Types

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

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

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

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 [SOAP1.2-1/2007] section 5.4.5, which conforms to the **XML schema** of the **SOAPFaultDetails** complex type specified in section 2.2.4.1. 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 SitesSoap Server Details

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

Each operation consists of a basic request-response pair, and the server treats each request as an independent transaction that is unrelated to any previous request. The following table summarizes the **WSDL operations** defined in this specification.

Operation	Description
CreateWeb	Creating a **subsite**.
DeleteWeb	Deleting a subsite.
ExportSolution	Exporting a **site** as a solution.
ExportWeb	Exporting site content.
ExportWorkflowTemplate	Exporting a **workflow template** as a solution.
GetSite	Getting information about the **site collection**.
GetSiteTemplates	Getting **available site template** information.

Operation	Description
GetUpdatedFormDigest	Getting a new **form digest validation**.
GetUpdatedFormDigestInformation	Getting a new form digest validation and its expiration time.
ImportWeb	Importing a site.

3.1.4.1 ExportSolution

This operation is used to export the content related to a **site** to the **solution gallery**.[<1>]

```
<wsdl:operation name="ExportSolution">
  <wsdl:input message="tns:ExportSolutionSoapIn" />
  <wsdl:output message="tns:ExportSolutionSoapOut" />
</wsdl:operation>
```

The client sends an **ExportSolutionSoapIn** request message and the server responds with an **ExportSolutionSoapOut** response message.

3.1.4.1.1 Messages

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

3.1.4.1.1.1 ExportSolutionSoapIn

This message is the request message for exporting information about a **site**.

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

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

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

3.1.4.1.1.2 ExportSolutionSoapOut

This message is the response message for exporting information about a **site**.

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

3.1.4.1.2 Elements

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

3.1.4.1.2.1 ExportSolution

Defines the input parameters for the **ExportSolution** operation.

```
<s:element name="ExportSolution">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="solutionFileName" 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="fullReuseExportMode" type="s:boolean" />
            <s:element minOccurs="1" maxOccurs="1" name="includeWebContent" type="s:boolean" />
          </s:sequence>
        </s:complexType>
      </s:element>
```

solutionFileName: The name of the solution file that will be created. The file name MUST include an extension of ".wsp". If a solution with the specified name already exists in the **solution gallery**, the server MUST retry with the following sequence of names:

▪ <filename>2.wsp, <filename>3.wsp, …, <filename>1000.wsp

Where <filename> is obtained from **solutionFileName** after excluding the extension.

If a unique name is obtained, the server MUST continue with that name, and if the sequence is exhausted, the server MUST return a **SOAP fault** with no error code.

title: The name of the solution.

description: Detailed information that describes the solution.

fullReuseExportMode: Specifies the scope of data that needs to be exported. If set to true, only the **lists**, **workflows**, **fields**, **content types**, **custom actions** and **files** corresponding to the **site** MUST be packaged into the **solution package**. If set to **false**, in addition to all the items corresponding to the site, the fields, content types and workflows corresponding to the **site collection** MUST be packaged into the solution. If the exported solution will be reused only within the same site collection (that is, items specific to the site collection will be reused and do not have to be packaged into the solution) **fullReuseExportMode** SHOULD be set to **true**. If the exported solution will be reused in other site collections (that is, the exported solution needs to package all items necessary for making it portable to different site collections) **fullReuseExportMode** MUST be set to **false**.

includeWebContent: Specifies whether the solution needs to include the contents of all lists and **document libraries** in the site. If set to **true**, all contents of lists and document libraries MUST be packaged in the solution.

3.1.4.1.2.2 ExportSolutionResponse

Defines the output of the **ExportSolution** operation.

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

ExportSolutionResult: It MUST be the **site-collection relative URL** of the created solution file.

3.1.4.2 ExportWeb

This operation<2> is used to export the content related to a **site** into one or more **content migration package** files, as specified in [MS-PRIMEPF]. This operation saves the files to the server, which can be used to migrate or restore a site.

```
<wsdl:operation name="ExportWeb">
  <wsdl:input message="tns:ExportWebSoapIn" />
  <wsdl:output message="tns:ExportWebSoapOut" />
```

```
                </wsdl:operation>
```

The client sends an **ExportWebSoapIn** request message and the server responds with an **ExportWebSoapOut** response message, as follows:

1. In cases of permission, space restrictions or other conditions that prevent the execution of the operation, an error code MUST be included in the response message as specified in section 3.1.4.2.2.2.

2. Upon the start of the export operation, a result of 1 MUST be sent as the response indicating that the operation is in progress.

3. Upon the completion of the export operation, the result of the operation MUST be available to be retrieved in an implementation-specific format<3>.

4. Upon the successful completion of the export operation, the content migration package file(s) MUST be created in the server location specified in the request message.

3.1.4.2.1 Messages

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

3.1.4.2.1.1   ExportWebSoapIn

This message is the request message for exporting information about a **site**.

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

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

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

3.1.4.2.1.2   ExportWebSoapOut

This message is the response message for exporting information about a **site**.

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

3.1.4.2.2 Elements

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

3.1.4.2.2.1   ExportWeb

Defines the input parameters for the **ExportWeb** operation.

```
<s:element name="ExportWeb">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="jobName" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="webUrl" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="dataPath" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="includeSubwebs"
                 type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="includeUserSecurity"
                 type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="overWrite" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="cabSize" type="s:int" />
```

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

jobName: Specifies the operation. It MUST NOT include any of the following characters:

- Slash (/)

- Double backslash (\\)

- Colon (:)

- Asterisk (*)

- Question mark (?)

- Double quotes (")

- Angle brackets (< >)

- Vertical bar (|)

- Number sign (#)

- \t

- Curly braces ({ })

- Percent sign (%.)

The server MUST append the file extension .cmp to the **jobName** to form the file name for the first **content migration package** file. If multiple content migration package files are created, then the server MUST also append a positive, incrementing integer to the name of each subsequent content migration package file.

webUrl: The **URL** of the **site** to export. The **URL** SHOULD be case-sensitive. The host (server) name portion of the URL MUST be in lowercase.

dataPath: The full path of the location on the server where the content migration package file(s) are saved. The user invoking this method MUST have write permission on the server.

includeSubwebs: Specifies whether to include the **subsite**. The server MUST recursively include subsites in the content migration package file(s) when **true** is specified.

includeUserSecurity: Specifies whether to include **access control list (ACL)**, **security group**, and **membership group** information. If set to **true**, the server MUST include security group and membership group information in the content migration package file(s).

overWrite: Specifies whether to overwrite the content migration package file(s) if they exist. The server MUST NOT overwrite existing file(s) with the new file(s) if **false** is specified. If the server cannot create a new file because this parameter is set to **false**, it MUST return error code 5 as specified in section 3.1.4.2.2.2.

cabSize: Indicates the suggested size in megabytes for the content migration package file(s).

The following algorithm MUST be used for determining the resulting cabinet file sizes based on the value of the **cabSize** element:

1. If **cabSize** is less than zero, then ExportError (code 7) is returned in **ExportWebResult**, as defined in **ExportWebResponse** (section 3.1.4.2.2.2).<4>

2. If **cabSize** is equal to zero or **cabSize** is greater than 0x0400, then modify **cabSize** to be 0x0018.

3. Start a new cabinet file.

4. Add a site content file to the cabinet.

5. If no more files, exit.

6. If ((current size of the cabinet) + (.5 * the size of the file just added)) is greater than or equal to **cabSize**, then go to step 3.

7. Go to step 4.

3.1.4.2.2.2 ExportWebResponse

Defines the output of the **ExportWeb** operation.

```
<s:element name="ExportWebResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ExportWebResult" type="s:int" />
    </s:sequence>
  </s:complexType>
</s:element>
```

ExportWebResult: The result of the operation as specified in the following table.

Value	Description
1	Pending: The operation is in progress.
4	InvalidExportUrl: The **site** specified in the **webUrl** is not accessible.
5	ExportFileNoAccess: The location specified in the **dataPath** is not accessible.
6	ExportWebNoAccess: The user does not have the required permissions to execute this operation successfully. The user MUST have the ManageLists permission ([MS-WSSFO3] section 2.2.2.15) for the location on the server where the **content migration package files** are saved.
7	ExportError: An error other than the errors listed in this table occurred while exporting the site. An example of a condition that would result in this error is if there was no content on the site to export.
8	UploadDataError: The content migration package file is not uploaded to the server successfully. This value is only returned if a server internal exception occurs.

3.1.4.3 ExportWorkflowTemplate

This operation is used to export a declarative **workflow template** as a **site solution** to the specified **document library**.<5>

```
<wsdl:operation name="ExportWorkflowTemplate">
  <wsdl:input message="tns:ExportWorkflowTemplateSoapIn" />
  <wsdl:output message="tns:ExportWorkflowTemplateSoapOut" />
</wsdl:operation>
```

The client sends an **ExportWorkflowTemplateSoapIn** request message and the server responds with an **ExportWorkflowTemplateSoapOut** response message.

3.1.4.3.1 Messages

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

3.1.4.3.1.1 ExportWorkflowTemplateSoapIn

This message is the request message for exporting information about a **site**.

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

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

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

3.1.4.3.1.2 ExportWorkflowTemplateSoapOut

This message is the response message for exporting information about a **site**.

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

3.1.4.3.2 Elements

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

3.1.4.3.2.1 ExportWorkflowTemplate

Defines the input parameters for the **ExportWorkflowTemplate** operation.

```
<s:element name="ExportWorkflowTemplate">
  <s:complexType>
   <s:sequence>
     <s:element minOccurs="1" maxOccurs="1" name="solutionFileName" 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="workflowTemplateName" type="s:string" />
     <s:element minOccurs="1" maxOccurs="1" name="destinationListUrl" type="s:string" />
   </s:sequence>
  </s:complexType>
</s:element>
```

solutionFileName: The name of the solution file that will be created. The file name MUST include an extension of ".wsp". If a solution with the specified name already exists in the **document library** in which the solution file needs to be created, the server MUST retry with the following sequence of names:

- *<filename>*2.wsp, *<filename>*3.wsp, …, *<filename>*1000.wsp

Where *<filename>* is obtained from **solutionFileName** after excluding the extension.

If a unique name is obtained, the protocol server MUST continue with that name. If the sequence is exhausted, the server MUST return a **SOAP fault** with no error code.

title: The name of the solution.

description: Detailed information that describes the solution.

workflowTemplateName: The name of the **workflow template** that is to be exported. It MUST refer to an existing workflow template. If a workflow template with the specified name does not exist, the server will return a **SOAP fault** with an implementation-specific error code.

destinationListUrl: The **server-relative URL** of the document library in which the solution file needs to be created. It MUST refer to an existing document library. If a document library with the specified name does not exist, the server MUST return a **SOAP fault** with no error code.

3.1.4.3.2.2   ExportWorkflowTemplateResponse

Defines the output of the **ExportWorkflowTemplate** operation.

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

ExportWorkflowTemplateResult: It MUST be the **site-relative URL** of the created solution file.

3.1.4.4 GetSite

This operation is used to retrieve information about the **site collection**.<6>

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

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

3.1.4.4.1 Messages

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

3.1.4.4.1.1   GetSiteSoapIn

This message is the request message for retrieving information about a **site collection**.

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

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

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

3.1.4.4.1.2   GetSiteSoapOut

This message is the response message for retrieving information about a **site collection**.

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

3.1.4.4.2 Elements

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

3.1.4.4.2.1 GetSite

Defines the input parameter of the **GetSite** operation.

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

SiteUrl: Specifies the absolute **Uniform Resource Locator (URL)** of a **site collection** or of a location within a site collection. It MUST be in the same server domain as the protocol server.

3.1.4.4.2.2 GetSiteResponse

Defines the output of the **GetSite** operation.

```
<s:element name="GetSiteResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetSiteResult"
                 type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetSiteResult: Represents information about the **site collection** that contains the **site URL**. It is in the following form:

```
<Site Url=UrlString Id=IdString UserCodeEnabled=UserCodeEnabledString />
```

Where *UrlString* is a quoted string that is the absolute URL of the site collection, *IdString* is a quoted string that is the **site collection identifier** of the site collection and *UserCodeEnabledString* is either "true" or "false" and specifies whether **user code** is enabled for the site collection. The value for *UserCodeEnabledString* MUST be "true" if user code is enabled for the site collection or "false" if it is not enabled.

3.1.4.5 GetSiteTemplates

This operation is used to retrieve information about the collection of **available site templates**, including both built-in **site definition configuration** and custom **site templates** defined by users for the **site collection** and the **farm**.

```
<wsdl:operation name="GetSiteTemplates">
  <wsdl:input message="tns:GetSiteTemplatesSoapIn" />
  <wsdl:output message="tns:GetSiteTemplatesSoapOut" />
</wsdl:operation>
```

The client sends a **GetSiteTemplatesSoapIn** request message, and the server responds with a **GetSiteTemplatesSoapOut** response message.

3.1.4.5.1 Messages

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

3.1.4.5.1.1   GetSiteTemplatesSoapIn

This message is the request message for retrieving **available site template** information of a **site**.

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

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

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

3.1.4.5.1.2   GetSiteTemplatesSoapOut

This message is the response message for retrieving **available site template** information.

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

3.1.4.5.2 Elements

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

3.1.4.5.2.1   GetSiteTemplates

Defines the input parameter of the **GetSiteTemplates** operation.

```
<s:element name="GetSiteTemplates">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="LCID" type="s:unsignedInt" />
    </s:sequence>
  </s:complexType>
</s:element>
```

LCID: Specifies the **language code identifier (LCID)**.

3.1.4.5.2.2   GetSiteTemplatesResponse

Defines the output of the **GetSiteTemplates** operation.

```
<s:element name="GetSiteTemplatesResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetSiteTemplatesResult"
                 type="s:unsignedInt" />
      <s:element minOccurs="0" maxOccurs="1" name="TemplateList"
                 type="tns:ArrayOfTemplate" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetSiteTemplatesResult: Represents the result of the operation. It MUST be 0 when the operation succeeded.

TemplateList: An array whose elements provide **fields** containing information about each **available site template**. The type of **TemplateList** is specified in section 3.1.4.5.3.1.

In case the **LCID** included in the request message indicates a language that is not installed on the server, this method MUST return a **SOAP fault** with the error code specified in the following table.<7>

Error Code	Description
0x81070209	The language of the LCID included in request message is not installed on the server.

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   ArrayOfTemplate

Describes the set of **available site templates**. It contains the element **Template**, which is defined in section 3.1.4.5.3.2.

```
<s:complexType name="ArrayOfTemplate">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="Template" nillable="true"
            type="tns:Template" />
  </s:sequence>
</s:complexType>
```

Template: XML element that contains available site template information.

3.1.4.5.3.2   Template

Contains a set of **XML attributes**. It is used to specify details about an **available site template**.

```
<s:complexType name="Template">
  <s:attribute name="ID" type="s:int" use="required" />
  <s:attribute name="Title" type="s:string" use="required" />
  <s:attribute name="Name" type="s:string" use="required" />
  <s:attribute name="IsUnique" type="s:boolean" use="required" />
  <s:attribute name="IsHidden" type="s:boolean" use="required" />
  <s:attribute name="Description" type="s:string" />
  <s:attribute name="ImageUrl" type="s:string" use="required" />
  <s:attribute name="IsCustom" type="s:boolean" use="required" />
  <s:attribute name="IsSubWebOnly" type="s:boolean" use="required" />
  <s:attribute name="IsRootWebOnly" type="s:boolean" use="required" />
  <s:attribute name="DisplayCategory" type="s:string" />
  <s:attribute name="FilterCategories" type="s:string" />
  <s:attribute name="HasProvisionClass" type="s:boolean" use="required" />
</s:complexType>
```

ID: Identifier of the **site definition** of this available site template.

Title: Display name of the available site template. It MUST be localized with the **LCID** specified in the request message.

Name: Internal name of the available site template. It contains the name of the site definition followed by a number sign (#), and then the **site definition configuration** number.

IsUnique: This is not used in this protocol. It MUST be **false** when sending and ignored on receipt.

IsHidden: Indicates whether it is a hidden available site template.

Description: Description of the available site template. It MUST be localized with the LCID specified in the request message.

ImageUrl: The **URL** of the image that is used to represent the available site template in the user interface. It MUST be the URL in relative to the URL of the **site**.

IsCustom: Reserved. It MUST be **true**.

IsSubWebOnly: Indicates whether the available site template is only applicable to **subsites** created within the **site collection**.

IsRootWebOnly: Indicates whether the available site template is only applicable to the root site in the site collection.

DisplayCategory: Name of the category that this available site template belongs to when creating a new site. This MUST be localized with the LCID specified in the request message.

FilterCategories: Name of the category of the site that serves as the source for the custom **site template**.<8> Possible values are "Meetings" and "Collaboration". It MUST NOT be included if the available site template is not a custom site template.

HasProvisionClass: Indicates whether the available site template has a provision class, which provides logic for sites created with this available site template.

3.1.4.6 GetUpdatedFormDigest

This operation<9> is used to request renewal of an expired security validation, also known as a message digest.

The client calls **GetUpdatedFormDigest** in the following sequence of events when using **form digest validation**. Normally, the client requests a **page** from the server, and the server sends a page to the client containing a **form** and a security validation. This security validation expires after a server-configurable time period that is implementation–specific.<10> If the client submits the form to the server within the security validation time period, the server verifies that the security validation has not changed, and accepts the information submitted. The following diagram depicts this normal sequence of events.
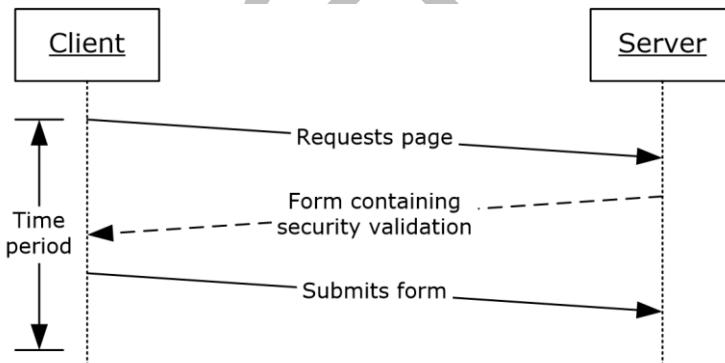
Figure 3: Form submission

If, however, the security validation time period expires, the client can call **GetUpdatedFormDigest** to request an updated security validation before submitting the form. In this case the server MUST return a new security validation to the client. When the client then submits the page containing the renewed security validation, the server verifies the renewed security validation and accepts the information submitted. The client determines that the security validation has expired through a

process that is implementation-specific and independent of this protocol. This protocol does not specify a method to explicitly determine whether a particular security validation has expired.

The following diagram depicts this sequence of events.
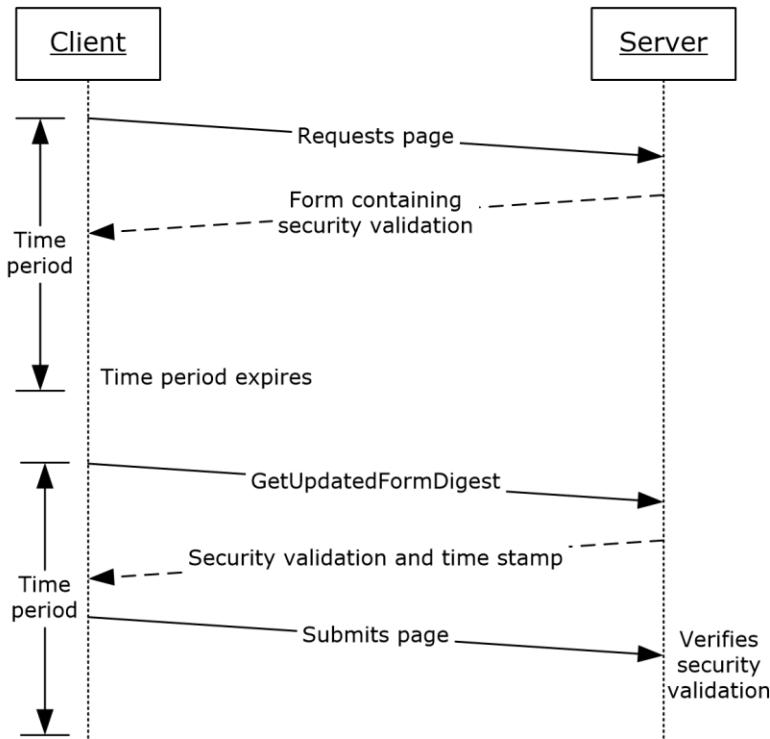
Figure 4: Renewal of an expired form digest validation

The format and contents of the security validation are implementation-specific; this protocol does not specify or require a particular format of security validation. The security validation is not read by the client; the page containing the security validation is submitted to the server.

The purpose of form digest validation is to help prevent security attacks where a user unknowingly posts data to a server.

```
<wsdl:operation name="GetUpdatedFormDigest">
  <wsdl:input message="tns:GetUpdatedFormDigestSoapIn" />
  <wsdl:output message="tns:GetUpdatedFormDigestSoapOut" />
</wsdl:operation>
```

The client sends a **GetUpdatedFormDigestSoapIn** request message, and the server responds with a **GetUpdatedFormDigestSoapOut** response message.

3.1.4.6.1 Messages

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

3.1.4.6.1.1   GetUpdatedFormDigestSoapIn

This message is the request message for obtaining a **form digest validation** value from the **site**.

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

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

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

3.1.4.6.1.2   GetUpdatedFormDigestSoapOut

This message is the response message for the request for a **form digest validation** value.

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

3.1.4.6.2 Elements

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

3.1.4.6.2.1   GetUpdatedFormDigest

Defines the input parameter of the **GetUpdatedFormDigest** operation.

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

3.1.4.6.2.2   GetUpdatedFormDigestResponse

Defines the output of the **GetUpdatedFormDigest** operation.

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

GetUpdatedFormDigestResult: A security validation token generated by the server and used by the server to validate the request.<11>

3.1.4.7 GetUpdatedFormDigestInformation

This operation is used to request renewal of an expired security validation token, also known as a message digest, and the new security validation token's expiration time.<12>

The security validation token and its usage are specified in operation **GetUpdatedFormDigest** (section 3.1.4.6). In this operation, the protocol server MUST return the security validation token's expiration time in addition to the security validation token.

```
<wsdl:operation name="GetUpdatedFormDigestInformation">
  <wsdl:input message="tns:GetUpdatedFormDigestInformationSoapIn" />
  <wsdl:output message="tns:GetUpdatedFormDigestInformationSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetUpdatedFormDigestInformationSoapIn** request message and the protocol server responds with a **GetUpdatedFormDigestInformationSoapOut** response message.

3.1.4.7.1 Messages

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

3.1.4.7.1.1 GetUpdatedFormDigestInformationSoapIn

This message is the request message for obtaining a security validation token and the security validation token's expiration time from the **site**.

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

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

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

3.1.4.7.1.2 GetUpdatedFormDigestInformationSoapOut

This message is the response message for the request for a security validation token and the security validation token's expiration time.

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

3.1.4.7.2 Elements

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

3.1.4.7.2.1 GetUpdatedFormDigestInformation

This element defines the input parameter of the **GetUpdatedFormDigestInformation** operation.

```
<s:element name="GetUpdatedFormDigestInformation">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

url: An optional **page URL**. If this element is either omitted altogether or included as an empty string, the protocol server MUST return the **FormDigestInformation** of the current requested **site**. Otherwise, the protocol server MUST return the **FormDigestInformation** of the site that contains the page specified by this element.

3.1.4.7.2.2 GetUpdatedFormDigestInformationResponse

This element defines the output of the **GetUpdatedFormDigestInformation** operation.

```
<s:element name="GetUpdatedFormDigestInformationResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1"
                 name="GetUpdatedFormDigestInformationResult"
                 type="tns:FormDigestInformation" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetUpdatedFormDigestInformationResult: A **FormDigestInformation**, as specified in section 3.1.4.7.3.1, that contains a security validation token generated by the protocol server, the security validation token's expiration time in seconds, and other information. For more details, see section 3.1.4.7.3.1.

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   FormDigestInformation

This complex type contains the security validation token generated by the protocol server and its expiration time in seconds.

```
<s:complexType name="FormDigestInformation">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="DigestValue" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="TimeoutSeconds" type="s:int" />
    <s:element minOccurs="0" maxOccurs="1" name="WebFullUrl" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="LibraryVersion" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="SupportedSchemaVersions"
type="s:string" />
  </s:sequence>
</s:complexType>
```

DigestValue: Security validation token generated by the protocol server.

TimeoutSeconds: The time in seconds in which the security validation token will expire after the protocol server generates the security validation token server.

WebFullUrl: The absolute **URL** of the **site**.

LibraryVersion: The library version, as specified in [MS-CSOM] section 3.1.4.1.4.12, of the protocol server.

SupportedSchemaVersions: The comma-separated list of the SharePoint Client Query Protocol versions, as specified in [MS-CSOM] section 3.1.4.1.3.36.2, that are supported by the protocol server.

3.1.4.8 ImportWeb

This operation<13> is used to import a **site** from one or more **content migration package** files, as specified in [MS-PRIMEPF], to a specified **URL**.

```
<wsdl:operation name="ImportWeb">
  <wsdl:input message="tns:ImportWebSoapIn" />
  <wsdl:output message="tns:ImportWebSoapOut" />
</wsdl:operation>
```

The client sends an **ImportWebSoapIn** request message and the server responds with an **ImportWebSoapOut** response message as follows:

- If a condition occurs that prevents the server from executing the operation, an error code MUST be included in the response message as specified in section 3.1.4.8.2.2.

- Upon the start of the import operation, a result of 1 MUST be sent as the response indicating that the operation is in progress.

- Upon completion of the import operation, the result of the operation MUST be available to be retrieved in an implementation-specific format.<14>

- Upon successful completion of the operation, the site at the specified URL MUST contain all the content from the specified content migration package files.

3.1.4.8.1 Messages

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

3.1.4.8.1.1 ImportWebSoapIn

This message is the request message for importing a **site** from the specified **content migration package** files on the server.

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

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

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

3.1.4.8.1.2 ImportWebSoapOut

This message is the response message for importing a **site** on the server.

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

3.1.4.8.2 Elements

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

3.1.4.8.2.1 ImportWeb

Defines the input parameter of the **ImportWeb** operation.

```
<s:element name="ImportWeb">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="jobName" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="webUrl" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="dataFiles"
                type="tns:ArrayOfString" />
      <s:element minOccurs="0" maxOccurs="1" name="logPath" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="includeUserSecurity"
                type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="overWrite" type="s:boolean" />
    </s:sequence>
  </s:complexType>
</s:element>
```

jobName: Specifies the operation. It MUST NOT include any of the following characters:

- Slash (/)

- Double backslash (\\)

- Colon (:)

- Asterisk (*)

- Question mark (?)

- Double quotes (")

- Angle brackets (< >)

- Vertical bar (|)

- Number sign (#)

- \t

- Curly braces ({ })

- Percent sign (%)

webUrl: The **URL** of the resulting **Web site**. The caller MUST have site creation rights to perform this operation.

dataFiles: The URLs of the **content migration package** files on the server that the server imports to create the resulting Web site. The user MUST have read permission to all the specified content migration package files. This **dataFiles** element is an **ArrayOfString** as defined in section 2.2.4.2.

If the **dataFiles** array contains multiple content migration package files, the first content migration package file listed in the array MUST be the one that contains the list of all the content migration package files.

logPath:<15> The URL where the server places files describing the progress or status of the operation. The user MUST have write permission to this location. If this element is omitted, the server MUST NOT create any files describing the progress or status of the operation.

includeUserSecurity: Specifies whether or not to include **access control list (ACL)**, **security group**, and **membership group** information in the resulting Web site. If true, the server MUST include ACL, security group, and membership group information from the content migration package files in the resulting Web site.

overWrite: Specifies whether or not to overwrite existing files at the location specified by **logPath**. The server MUST NOT overwrite existing files with new files if false is specified. If the server cannot create a new file because this parameter is set to **false**, it MUST return error code 11 as specified in section 3.1.4.8.2.2.

3.1.4.8.2.2   ImportWebResponse

Defines the output of the **ImportWeb** operation.

```
<s:element name="ImportWebResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ImportWebResult" type="s:int" />
    </s:sequence>
  </s:complexType>
</s:element>
```

ImportWebResult: The result of the operation as specified in the following table.

Value	Description
1	Pending: The operation is in progress.
2	GenericError: An error other than the errors listed in this table occurred importing the **site**.

Value	Description
4	InvalidImportUrl: The site specified in the **webUrl** is not accessible.
5	ImportFileNoAccess: At least one location specified in **dataFiles** is not accessible.
6	ImportWebNoAccess: The user has insufficient permission to import to the location specified in **webUrl**.
7	ImportWebCantCreate: The site specified in the **webUrl** cannot be created.
8	ImportWebNotEmpty: The location specified by **webUrl** corresponds to an existing site that is not a **blank site**.
11	LogFileNoAccess: The location specified by **logPath** is not accessible.
12	ImportError: There was an error importing the site. See message for details.

3.1.4.9 CreateWeb

This operation is used to create a new **subsite** of the current **site**.<16>

```
<wsdl:operation name="CreateWeb">
  <wsdl:input message="tns:CreateWebSoapIn" />
  <wsdl:output message="tns:CreateWebSoapOut" />
</wsdl:operation>
```

The client sends a **CreateWebSoapIn** request message and the server responds with a **CreateWebSoapOut** response message upon successful completion of creating the subsite. If any of the error conditions specified by the following table occur, this method MUST return a **SOAP fault** with the specified error code.

Error code	Specification
0x800700b7	The location specified by **CreateWeb/url** is already in use.
0x8102009f	The template specified by **CreateWeb/templateName** does not exist.

3.1.4.9.1 Messages

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

3.1.4.9.1.1 CreateWebSoapIn

This message is the request message for creating a new **subsite** of the current **site**.

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

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

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

3.1.4.9.1.2 CreateWebSoapOut

This message is the response message for creating a new **subsite** of the current **site**.

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

3.1.4.9.2 Elements

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

3.1.4.9.2.1   CreateWeb

Defines the input parameters for the **CreateWeb** operation.

```
<s:element name="CreateWeb">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="url" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="title" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="description" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="templateName" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="language" type="s:unsignedInt" />
      <s:element minOccurs="0" maxOccurs="1" name="locale" type="s:unsignedInt" />
      <s:element minOccurs="0" maxOccurs="1" name="collationLocale" type="s:unsignedInt" />
      <s:element minOccurs="0" maxOccurs="1" name="uniquePermissions" type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="anonymous" type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="presence" type="s:boolean" />
    </s:sequence>
  </s:complexType>
</s:element>
```

url: The **site-relative URL** of the **subsite** to be created.

title: The display name of the subsite to be created.

description: The description of the subsite to be created. If this element is omitted, an empty string MUST be used.

templateName: The name of an **available site template** to be used for the subsite to be created. MUST match a value in **GetSiteTemplatesResponse/TemplateList/Template/@Name** as would be returned by a call to **GetSiteTemplates** (section 3.1.4.5).

language: An **LCID** that specifies the language of the user interface of the subsite to be created. If omitted or zero, the subsite to be created MUST use the server's default language for the user interface.

locale: An LCID that specifies the display format for numbers, dates, times, and currencies in the subsite to be created. If omitted or zero, specifies that the subsite to be created MUST use the server's default settings for displaying data.

collationLocale: An LCID that specifies the **collation order** to use in the subsite to be created. If omitted or zero, specifies that the subsite to be created MUST use the server's default collation order.

uniquePermissions: If set to **true**, specifies that the subsite to be created uses its own set of **permissions**, which default to the current user having full control and no other users having access. If set to false or omitted, the subsite to be created MUST inherit its permissions from its **parent site**.

anonymous: If set to **true**, the **anonymous authentication** is to be allowed for the subsite to be created. If set to false or omitted, the anonymous authentication MUST NOT be allowed for the subsite to be created.

presence: If set to **true**, the online **presence** information is to be enabled for the subsite to be created. If set to false or omitted, the online presence information MUST NOT be enabled for the subsite to be created.<17>

3.1.4.9.2.2 CreateWebResponse

Defines the output of the **CreateWeb** operation.

```
<s:element name="CreateWebResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="CreateWebResult">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="CreateWeb">
              <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>
```

Url: The **fully qualified URL** to the **subsite** which was successfully created.

3.1.4.10 DeleteWeb

This operation is used to delete an existing **subsite** of the current **site**.<18>

```
<wsdl:operation name="DeleteWeb">
  <wsdl:input message="tns:DeleteWebSoapIn" />
  <wsdl:output message="tns:DeleteWebSoapOut" />
</wsdl:operation>
```

The client sends a **DeleteWebSoapIn** request message, and the server responds with a **DeleteWebSoapOut** response message upon successful completion of deleting the subsite. Otherwise, the server MUST return a **SOAP fault** that is defined in the **DeleteWebResponse** message.

3.1.4.10.1 Messages

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

3.1.4.10.1.1 DeleteWebSoapIn

This message is the request message for deleting an existing **subsite** of the current **site**.

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

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

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

3.1.4.10.1.2 DeleteWebSoapOut

This message is the response message for deleting an existing **subsite** of the current **site**.

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

3.1.4.10.2   Elements

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

3.1.4.10.2.1 DeleteWeb

Defines the input parameters for the **DeleteWeb** operation.

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

url: The **site-relative URL** of the **subsite** to be deleted.

3.1.4.10.2.2 DeleteWebResponse

Defines the output of the **DeleteWeb** operation, which contains a **DeleteWebResponse** element with the following definition.

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

3.1.4.11   IsScriptSafeUrl

This operation<19> is used to validate whether the specified **URLs** are valid script safe URLs for the current **site**.

Script safe URLs are URLs the server trusts as valid sources for IFRAMEs in **Hypertext Markup Language (HTML)** content.

```
<wsdl:operation name="IsScriptSafeUrl">
  <wsdl:input message="tns:IsScriptSafeUrlSoapIn" />
  <wsdl:output message="tns:IsScriptSafeUrlSoapOut" />
</wsdl:operation>
```

The client sends an **IsScriptSafeUrlSoapIn** request message, and the server responds with an **IsScriptSafeUrlSoapOut** response message.

3.1.4.11.1   Messages

The following WSDL message definitions are specific to this operation.

3.1.4.11.1.1 IsScriptSafeUrlSoapIn

This message is the request message for validating whether the specified **URLs** are valid script safe URLs for the current **site**.

The SOAP action value of the message is defined as:

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

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

3.1.4.11.1.2 IsScriptSafeUrlSoapOut

This message is the response message for validating whether the specified **URLs** are valid script safe URLs for the current **site**.

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

3.1.4.11.2 Elements

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

3.1.4.11.2.1 IsScriptSafeUrl

This element defines the input parameter for the **IsScriptSafeUrl** operation.

```
<s:element name="IsScriptSafeUrl">
  <s:complexType>
    <s:sequence>
      <s:element name="urls" type="tns:ArrayOfString" maxOccurs="1" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

urls: An **ArrayOfString** as defined in section 2.2.4.2. This element contains all **URLs** that need to be validated.

Each URL in **urls** MUST NOT be null or empty, and MUST be relative or absolute Uniform Resource Locator (URL) values. Relative URLs MUST be **site-relative URLs**.

3.1.4.11.2.2 IsScriptSafeUrlResponse

Defines the output of the **IsScriptSafeUrl** operation, which contains an **IsScriptSafeUrlResponse** element with the following definition.

```
<s:element name="IsScriptSafeUrlResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="IsScriptSafeUrlResult"
type="tns:ArrayOfBoolean" />
    </s:sequence>
  </s:complexType>
</s:element>
```

IsScriptSafeUrlResult: An **ArrayOfBoolean** as defined in section 3.1.4.11.3.1, it contains results for validating the **URLs** specified by the parameter **urls** in section 3.1.4.11.2.1.

3.1.4.11.3 Complex Types

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

3.1.4.11.3.1 ArrayOfBoolean

This complex type contains an array of **boolean** elements. It is used for the response element **IsScriptSafeUrlResult** for the **IsScriptSafeUrl** operation.

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

boolean: Indicates whether a **URL** is a valid script safe URL for the current **site**.

3.1.4.12   IsScriptSafeUrlUsingCustomizedDomain

This operation[20] is used to validate whether the specified URLs are valid script safe URLs for the current site by checking against the customized script safe domains property of the site collection.

Script safe URLs are URLs the server trusts as valid sources for IFRAMES in **Hypertext Markup Language (HTML)** content.

```
<wsdl:operation name="IsScriptSafeUrlUsingCustomizedDomain">
  <wsdl:input message="tns:IsScriptSafeUrlUsingCustomizedDomainSoapIn" />
  <wsdl:output message="tns:IsScriptSafeUrlUsingCustomizedDomainSoapOut" />
</wsdl:operation>
```

The client sends an **IsScriptSafeUrlUsingCustomizedDomainSoapIn** request message, and the server responds with an **IsScriptSafeUrlUsingCustomizedDomainSoapOut** response message.

3.1.4.12.1   Messages

The following WSDL message definitions are specific to this operation.

3.1.4.12.1.1 IsScriptSafeUrlUsingCustomizedDomainSoapIn

This message is the request message for validating whether the specified URLs are valid script safe URLs for the current site by checking against the customized script safe domains property of the site collection.

The SOAP action value of the message is defined as:

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

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

3.1.4.12.1.2 IsScriptSafeUrlUsingCustomizedDomainSoapOut

This message is the response message for validating whether the specified URLs are valid script safe URLs for the current site by checking against the customized script safe domains property of the site collection.

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

3.1.4.12.2   Elements

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

3.1.4.12.2.1 IsScriptSafeUrlUsingCustomizedDomain

This element defines the input parameter for the **IsScriptSafeUrlUsingCustomizedDomain** operation.

```
<s:element name="IsScriptSafeUrlUsingCustomizedDomain">
  <s:complexType>
    <s:sequence>
      <s:element name="urls" type="tns:ArrayOfString" maxOccurs="1" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

urls: An **ArrayOfString** as defined in section 2.2.4.2. This element contains all URLs that need to be validated.

Each URL in **urls** MUST NOT be null or empty, and MUST be relative or absolute Uniform Resource Locator (URL) values. Relative URLs MUST be site-relative URLs.

3.1.4.12.2.2 IsScriptSafeUrlUsingCustomizedDomainResponse

Defines the output of the **IsScriptSafeUrlUsingCustomizedDomain** operation, which contains an **IsScriptSafeUrlUsingCustomizedDomainResponse** element with the following definition.

```
<s:element name="IsScriptSafeUrlUsingCustomizedDomainResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1"
  name="IsScriptSafeUrlUsingCustomizedDomainResult" type="tns:ArrayOfBoolean" />
    </s:sequence>
  </s:complexType>
</s:element>
```

IsScriptSafeUrlUsingCustomizedDomainResult: An **ArrayOfBoolean** as defined in section 3.1.4.12.3.1, it contains results for validating the URLs specified by the parameter **urls** in section 3.1.4.12.2.1.

3.1.4.12.3   Complex Types

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

3.1.4.12.3.1 ArrayOfBoolean

This complex type contains an array of **boolean** elements. It is used for the response element **IsScriptSafeUrlUsingCustomizedDomainResult** for the **IsScriptSafeUrlUsingCustomizedDomain** operation.

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

boolean: Indicates whether a URL is a valid script safe URL for the current site by checking against the customized script safe domains property of the site collection.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 Migrating a Site

In this scenario, a user needs to migrate the content of a **site** (*http://<serverName>/<siteNameA>*) to a different location (*http://<serverName>/<siteNameB>*). The following occurs:

1. User verifies the permission of the caller to the Web method. The caller either belongs to administrator role or a custom role which includes the open Web permission and manage Web permission on the site to be migrated and to the destination location (*http://<serverName>/<siteNameA>* and *http://<serverName>/<siteNameB>*), as well as the location where the **content migration package** files are stored (*http://<serverName>/<siteNameA>/Shared%20Documents/*).

2. User calls the **ExportWeb** method with the required information in the request message as follows:

```
<ExportWeb xmlns="http://schemas.microsoft.com/sharepoint/soap/">
    <jobName>ExportSite</jobName>
    <webUrl>http://serverName/siteNameA</webUrl>
<dataPath>http://serverName/siteNameA/Shared%20Documents</dataPath>
    <includeSubwebs>true</includeSubwebs>
    <includeUserSecurity>true</includeUserSecurity>
    <overWrite>true</overWrite>
    <cabSize>50</cabSize>
</ExportWeb>
```

3. Server starts the export process and sends a response message including **ExportWebResult** value of 1 indicating that the export process is in progress.

```
<ExportWebResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
    <ExportWebResult>1</ExportWebResult>
</ExportWebResponse>
```

4. Upon the successful completion of the export process, the server generates one or more content migration package files in the location specified in the request message. For the preceding response message, the server generates content migration package files **ExportSite.cmp** and **ExportSite1.cmp** in *http://<serverName>/<siteNameA>/Shared%20Documents*.

5. User calls **ImportWeb** with the content migration package files and the server migrates the content of the site to the new location *http://<serverName>/<siteNameB>*. For this example, the request message is:

```
<ImportWeb xmlns="http://schemas.microsoft.com/sharepoint/soap/">
    <jobName>ImportSite</jobName>
    <webUrl>http://serverName/siteNameB</webUrl>
    <dataFiles>
<string>http://serverName/siteNameA/Shared%20Documents/ExportSite.cmp</string>
<string>http://serverName/siteNameA/Shared%20Documents/ExportSite1.cmp</string>
    </dataFiles>
    <logPath>http://serverName/siteNameA/Shared%20Documents</logPath>
    <includeUserSecurity>true</includeUserSecurity>
    <overWrite>true</overWrite>
</ImportWeb>
```

6. Server starts the import process and sends a response message, which includes **ImportWebResult** value of 1 indicating the import process is in progress.

```
<ImportWebResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
   <ImportWebResult>1</ImportWebResult>
</ImportWebResponse>
```

7. As a result of the successful execution of the import process, a new site
   http://*<serverName>*/*<siteNameB>* is created with the same content of
   http://*<serverName>*/*<siteNameA>*.

5   Security

5.1   Security Considerations for Implementers

None.

5.2   Index of Security Parameters

None.

6 Appendix A: Full WSDL

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

```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:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
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="CreateWeb">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="url" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="title" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="description" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="templateName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="language" type="s:unsignedInt" />
            <s:element minOccurs="0" maxOccurs="1" name="locale" type="s:unsignedInt" />
            <s:element minOccurs="0" maxOccurs="1" name="collationLocale"
type="s:unsignedInt" />
            <s:element minOccurs="0" maxOccurs="1" name="uniquePermissions" type="s:boolean"
/>
            <s:element minOccurs="0" maxOccurs="1" name="anonymous" type="s:boolean" />
            <s:element minOccurs="0" maxOccurs="1" name="presence" type="s:boolean" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="CreateWebResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="CreateWebResult">
              <s:complexType>
                <s:sequence>
                  <s:element minOccurs="1" maxOccurs="1" name="CreateWeb">
                    <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="DeleteWeb">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="url" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="DeleteWebResponse">
        <s:complexType />
      </s:element>
      <s:element name="GetSite">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="SiteUrl"
                type="s:string" />
          </s:sequence>
```

```
        </s:complexType>
      </s:element>
      <s:element name="GetSiteResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetSiteResult"
                type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetSiteTemplates">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="LCID"
                type="s:unsignedInt" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetSiteTemplatesResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetSiteTemplatesResult"
                type="s:unsignedInt" />
            <s:element minOccurs="0" maxOccurs="1" name="TemplateList"
                type="tns:ArrayOfTemplate" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfTemplate">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="Template"
              nillable="true" type="tns:Template" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="Template">
        <s:attribute name="ID" type="s:int" use="required" />
        <s:attribute name="Title" type="s:string" use="required" />
        <s:attribute name="Name" type="s:string" use="required" />
        <s:attribute name="IsUnique" type="s:boolean" use="required" />
        <s:attribute name="IsHidden" type="s:boolean" use="required" />
        <s:attribute name="Description" type="s:string" />
        <s:attribute name="ImageUrl" type="s:string" use="required" />
        <s:attribute name="IsCustom" type="s:boolean" use="required" />
        <s:attribute name="IsSubWebOnly" type="s:boolean" use="required" />
        <s:attribute name="IsRootWebOnly" type="s:boolean" use="required" />
        <s:attribute name="DisplayCategory" type="s:string" />
        <s:attribute name="FilterCategories" type="s:string" />
        <s:attribute name="HasProvisionClass" type="s:boolean" use="required" />
      </s:complexType>
      <s:element name="GetUpdatedFormDigest">
        <s:complexType />
      </s:element>
      <s:element name="GetUpdatedFormDigestResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1"
                name="GetUpdatedFormDigestResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetUpdatedFormDigestInformation">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetUpdatedFormDigestInformationResponse">
        <s:complexType>
```

```
            <s:sequence>
              <s:element minOccurs="1" maxOccurs="1"
                        name="GetUpdatedFormDigestInformationResult"
                        type="tns:FormDigestInformation" />
            </s:sequence>
          </s:complexType>
      </s:element>
      <s:complexType name="FormDigestInformation">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="DigestValue" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="TimeoutSeconds" type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="WebFullUrl" type="s:string"
  />
            <s:element minOccurs="0" maxOccurs="1" name="LibraryVersion"
  type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="SupportedSchemaVersions"
  type="s:string" />
        </s:sequence>
      </s:complexType>
       <s:element name="IsScriptSafeUrl">
         <s:complexType>
           <s:sequence>
             <s:element minOccurs="0" maxOccurs="1" name="urls" type="tns:ArrayOfString" />
           </s:sequence>
         </s:complexType>
       </s:element>
       <s:element name="IsScriptSafeUrlResponse">
         <s:complexType>
           <s:sequence>
             <s:element minOccurs="1" maxOccurs="1" name="IsScriptSafeUrlResult"
  type="tns:ArrayOfBoolean" />
           </s:sequence>
         </s:complexType>
       </s:element>
       <s:element name="IsScriptSafeUrlUsingCustomizedDomain">
         <s:complexType>
           <s:sequence>
             <s:element minOccurs="0" maxOccurs="1" name="urls" type="tns:ArrayOfString" />
           </s:sequence>
         </s:complexType>
       </s:element>
       <s:element name="IsScriptSafeUrlUsingCustomizedDomainResponse">
         <s:complexType>
           <s:sequence>
             <s:element minOccurs="1" maxOccurs="1"
  name="IsScriptSafeUrlUsingCustomizedDomainResult" type="tns:ArrayOfBoolean" />
           </s:sequence>
         </s:complexType>
       </s:element>
       <s:element name="ExportWeb">
         <s:complexType>
           <s:sequence>
             <s:element minOccurs="1" maxOccurs="1" name="jobName"
                 type="s:string" />
             <s:element minOccurs="1" maxOccurs="1" name="webUrl" type="s:string" />
             <s:element minOccurs="1" maxOccurs="1" name="dataPath"
                 type="s:string" />
             <s:element minOccurs="1" maxOccurs="1" name="includeSubwebs"
                 type="s:boolean" />
             <s:element minOccurs="1" maxOccurs="1" name="includeUserSecurity"
                 type="s:boolean" />
             <s:element minOccurs="1" maxOccurs="1" name="overWrite"
                 type="s:boolean" />
             <s:element minOccurs="1" maxOccurs="1" name="cabSize" type="s:int" />
           </s:sequence>
         </s:complexType>
       </s:element>
       <s:element name="ExportWebResponse">
```

```
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="ExportWebResult"
                  type="s:int" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ImportWeb">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="jobName"
                  type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="webUrl" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="dataFiles"
                  type="tns:ArrayOfString" />
            <s:element minOccurs="0" maxOccurs="1" name="logPath"
                  type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="includeUserSecurity"
                  type="s:boolean" />
            <s:element minOccurs="1" maxOccurs="1" name="overWrite"
                  type="s:boolean" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfString">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="unbounded" name="string"
                nillable="true" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfBoolean">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="boolean" type="s:boolean" />
        </s:sequence>
      </s:complexType>
      <s:element name="ImportWebResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="ImportWebResult"
                  type="s:int" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ExportSolution">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="solutionFileName" 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="fullReuseExportMode"
type="s:boolean" />
            <s:element minOccurs="1" maxOccurs="1" name="includeWebContent" type="s:boolean"
/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ExportSolutionResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="ExportSolutionResult"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ExportWorkflowTemplate">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="solutionFileName" 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="workflowTemplateName"
type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="destinationListUrl" type="s:string"
/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ExportWorkflowTemplateResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="ExportWorkflowTemplateResult" minOccurs="1" maxOccurs="1"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="detail" type="s:SOAPFaultDetails" />
      <s:element name="Detail" type="s:SOAPFaultDetails" />
      <s:complexType name="SOAPFaultDetails">
        <s:sequence>
         <s:element name="errorstring" type="s:string"/>
         <s:element name="errorcode" type="s:string" minOccurs="0"/>
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="CreateWebSoapIn">
    <wsdl:part name="parameters" element="tns:CreateWeb" />
  </wsdl:message>
  <wsdl:message name="CreateWebSoapOut">
    <wsdl:part name="parameters" element="tns:CreateWebResponse" />
  </wsdl:message>
  <wsdl:message name="DeleteWebSoapIn">
    <wsdl:part name="parameters" element="tns:DeleteWeb" />
  </wsdl:message>
  <wsdl:message name="DeleteWebSoapOut">
    <wsdl:part name="parameters" element="tns:DeleteWebResponse" />
  </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="GetSiteTemplatesSoapIn">
    <wsdl:part name="parameters" element="tns:GetSiteTemplates" />
  </wsdl:message>
  <wsdl:message name="GetSiteTemplatesSoapOut">
    <wsdl:part name="parameters" element="tns:GetSiteTemplatesResponse" />
  </wsdl:message>
  <wsdl:message name="GetUpdatedFormDigestSoapIn">
    <wsdl:part name="parameters" element="tns:GetUpdatedFormDigest" />
  </wsdl:message>
  <wsdl:message name="GetUpdatedFormDigestSoapOut">
    <wsdl:part name="parameters" element="tns:GetUpdatedFormDigestResponse" />
  </wsdl:message>
  <wsdl:message name="GetUpdatedFormDigestInformationSoapIn">
    <wsdl:part name="parameters" element="tns:GetUpdatedFormDigestInformation" />
  </wsdl:message>
  <wsdl:message name="GetUpdatedFormDigestInformationSoapOut">
    <wsdl:part name="parameters" element="tns:GetUpdatedFormDigestInformationResponse" />
  </wsdl:message>
  <wsdl:message name="IsScriptSafeUrlSoapIn">
      <wsdl:part name="parameters" element="tns:IsScriptSafeUrl" />
  </wsdl:message>
  <wsdl:message name="IsScriptSafeUrlSoapOut">
      <wsdl:part name="parameters" element="tns:IsScriptSafeUrlResponse" />
  </wsdl:message>
  <wsdl:message name="IsScriptSafeUrlUsingCustomizedDomainSoapIn">
```

```
          <wsdl:part name="parameters" element="tns:IsScriptSafeUrlUsingCustomizedDomain" />
      </wsdl:message>
      <wsdl:message name="IsScriptSafeUrlUsingCustomizedDomainSoapOut">
          <wsdl:part name="parameters" element="tns:IsScriptSafeUrlUsingCustomizedDomainResponse"
/>
      </wsdl:message><wsdl:message name="ExportWebSoapIn">
          <wsdl:part name="parameters" element="tns:ExportWeb" />
      </wsdl:message>
      <wsdl:message name="ExportWebSoapOut">
          <wsdl:part name="parameters" element="tns:ExportWebResponse" />
      </wsdl:message>
      <wsdl:message name="ImportWebSoapIn">
          <wsdl:part name="parameters" element="tns:ImportWeb" />
      </wsdl:message>
      <wsdl:message name="ImportWebSoapOut">
          <wsdl:part name="parameters" element="tns:ImportWebResponse" />
      </wsdl:message>
      <wsdl:message name="ExportSolutionSoapIn">
          <wsdl:part name="parameters" element="tns:ExportSolution" />
      </wsdl:message>
      <wsdl:message name="ExportSolutionSoapOut">
          <wsdl:part name="parameters" element="tns:ExportSolutionResponse" />
      </wsdl:message>
      <wsdl:message name="ExportWorkflowTemplateSoapIn">
          <wsdl:part name="parameters" element="tns:ExportWorkflowTemplate" />
      </wsdl:message>
      <wsdl:message name="ExportWorkflowTemplateSoapOut">
          <wsdl:part name="parameters" element="tns:ExportWorkflowTemplateResponse" />
      </wsdl:message>
      <wsdl:portType name="SitesSoap">
        <wsdl:operation name="CreateWeb">
          <wsdl:input message="tns:CreateWebSoapIn" />
          <wsdl:output message="tns:CreateWebSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="DeleteWeb">
          <wsdl:input message="tns:DeleteWebSoapIn" />
          <wsdl:output message="tns:DeleteWebSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetSite">
          <wsdl:input message="tns:GetSiteSoapIn" />
          <wsdl:output message="tns:GetSiteSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetSiteTemplates">
          <wsdl:input message="tns:GetSiteTemplatesSoapIn" />
          <wsdl:output message="tns:GetSiteTemplatesSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetUpdatedFormDigest">
          <wsdl:input message="tns:GetUpdatedFormDigestSoapIn" />
          <wsdl:output message="tns:GetUpdatedFormDigestSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetUpdatedFormDigestInformation">
          <wsdl:input message="tns:GetUpdatedFormDigestInformationSoapIn" />
          <wsdl:output message="tns:GetUpdatedFormDigestInformationSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="IsScriptSafeUrl">
          <wsdl:input message="tns:IsScriptSafeUrlSoapIn" />
          <wsdl:output message="tns:IsScriptSafeUrlSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="IsScriptSafeUrlUsingCustomizedDomain">
          <wsdl:input message="tns:IsScriptSafeUrlUsingCustomizedDomainSoapIn" />
          <wsdl:output message="tns:IsScriptSafeUrlUsingCustomizedDomainSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="ExportWeb">
          <wsdl:input message="tns:ExportWebSoapIn" />
          <wsdl:output message="tns:ExportWebSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="ImportWeb">
          <wsdl:input message="tns:ImportWebSoapIn" />
          <wsdl:output message="tns:ImportWebSoapOut" />
```

```
      </wsdl:operation>
      <wsdl:operation name="ExportSolution">
        <wsdl:input message="tns:ExportSolutionSoapIn" />
        <wsdl:output message="tns:ExportSolutionSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="ExportWorkflowTemplate">
        <wsdl:input message="tns:ExportWorkflowTemplateSoapIn" />
        <wsdl:output message="tns:ExportWorkflowTemplateSoapOut" />
      </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="SitesSoap" type="tns:SitesSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="CreateWeb">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/CreateWeb"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteWeb">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteWeb"
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="GetSiteTemplates">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSiteTemplates" style="document"
/>
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetUpdatedFormDigest">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetUpdatedFormDigest"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetUpdatedFormDigestInformation">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetUpdatedFormDigestInformation"
style="document" />
      <wsdl:input>
```

```
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="IsScriptSafeUrl">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/IsScriptSafeUrl" style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="ExportWeb">
        <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/ExportWeb"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="ImportWeb">
        <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/ImportWeb"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="ExportSolution">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/ExportSolution" style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="ExportWorkflowTemplate">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/ExportWorkflowTemplate"
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="SitesSoap12" type="tns:SitesSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="CreateWeb">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/CreateWeb"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
```

```
      </wsdl:operation>
      <wsdl:operation name="DeleteWeb">
        <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteWeb"
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="GetSiteTemplates">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSiteTemplates" style="document"
/>
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetUpdatedFormDigest">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetUpdatedFormDigest"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetUpdatedFormDigestInformation">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetUpdatedFormDigestInformation"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="IsScriptSafeUrl">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/IsScriptSafeUrl" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="ExportWeb">
        <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/ExportWeb"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
```

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

7   Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Windows SharePoint Services 2.0

- Windows SharePoint Services 3.0

- Microsoft SharePoint Foundation 2010

- Microsoft SharePoint Foundation 2013

- Microsoft SharePoint Server 2016

- Microsoft SharePoint Server 2019

- Microsoft SharePoint Server Subscription Edition Preview

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

<1> Section 3.1.4.1: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and Microsoft Office SharePoint Server 2007 do not support this method.

<2> Section 3.1.4.2: Windows SharePoint Services 2.0 does not support this operation.

<3> Section 3.1.4.2: Windows SharePoint Services creates a file that contains the result of the export operation in the server location specified in the request message. The file includes a status code indicating the success of the operation or an error code. If the export operation succeeds, this file also includes the list of the **content migration package** files created. The client retrieves the file when it is created and parses the file to discover the result of the operation.

<4> Section 3.1.4.2.2.1: If the value of **cabSize** is less than zero, Windows SharePoint Services 3.0, SharePoint Foundation 2010, and SharePoint Foundation 2013 return a value of 1 or 7 but the server does not successfully complete the operation. The return code is not deterministic.

<5> Section 3.1.4.3: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and Office SharePoint Server 2007 do not support this method.

<6> Section 3.1.4.4: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and Office SharePoint Server 2007 do not support this method.

<7> Section 3.1.4.5.2.2: SharePoint Foundation 2010, SharePoint Foundation 2013, SharePoint Server 2016 and SharePoint Server 2019 do not return the **SOAP fault**. A successful **GetSiteTemplatesResponse** with an empty **TemplateList** element will be returned.

<8> Section 3.1.4.5.3.2: Windows SharePoint Services 3.0 and Office SharePoint Server 2007 do not use the values defined for **FilterCategories**.

<9> Section 3.1.4.6: Windows SharePoint Services 2.0 does not support this operation.

<10> Section 3.1.4.6: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 default to a 30-minute security validation time period.

<11> Section 3.1.4.6.2.2: The Windows SharePoint Services implementation of the security validation consists of two values separated by a comma. The first value is a security validation token also known as message digest, and the second value is a time stamp in **Coordinated Universal Time (UTC)** format.

The Windows SharePoint Services implementation of the digest includes validation specific to the user, site, and time period. When the client sends a page to the server, Windows SharePoint Services verifies that the security validation has not changed. This security validation expires after a configurable amount of time, causing Windows SharePoint Services validation to fail. Clients can use the **Sites.GetUpdatedFormDigest** operation to refresh the security validation so that validation will succeed.

<12> Section 3.1.4.7: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and Office SharePoint Server 2007 do not support this method.

<13> Section 3.1.4.8: Windows SharePoint Services 2.0 does not support this operation.

<14> Section 3.1.4.8: Windows SharePoint Services creates a file that contains the result of the import operation in the server location specified in the request message. The file includes a status code indicating the success of the operation or an error code. The client retrieves the file when it's created and parses the file to discover the result of the operation.

<15> Section 3.1.4.8.2.1: This parameter is optional. In SharePoint Foundation 2010 and SharePoint Foundation 2013, **ImportWeb** returns an error if **logPath** is set to an empty string.

<16> Section 3.1.4.9: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and Office SharePoint Server 2007 do not support this method.

<17> Section 3.1.4.9.2.1: If **anonymous authentication** is not allowed for the **subsite**, the online **presence** information will be enabled on SharePoint Foundation 2010 and Microsoft SharePoint Foundation 2013 Service Pack 1 (SP1).

<18> Section 3.1.4.10: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and Office SharePoint Server 2007 do not support this method.

<19> Section 3.1.4.11: Only SharePoint Foundation 2013 supports this method.

<20> Section 3.1.4.12: Only SharePoint Server 2019 supports this method.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

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.
- A document revision that captures changes to protocol functionality.

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 **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

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

Section	Description	Revision class
3.1.4.12 IsScriptSafeUrlUsingCustomizedDomain	Changed the property name CustomScriptSafeDomains as general desciption customized script safe domains.	Minor
3.1.4.12.1.1 IsScriptSafeUrlUsingCustomizedDomainSoapIn	Changed the property name CustomScriptSafeDomains as general desciption customized script safe domains.	Minor
3.1.4.12.1.2 IsScriptSafeUrlUsingCustomizedDomainSoapOut	Changed the property name CustomScriptSafeDomains as general desciption customized script safe domains.	Minor
3.1.4.12.3.1 ArrayOfBoolean	Changed the property name CustomScriptSafeDomains as general desciption customized script safe domains.	Minor
6 Appendix A: Full WSDL	Added SOAPFaultDetails and IsScriptSafeUrlUsingCustomizedDomain related content in Full WSDL.	Minor
7 Appendix B: Product Behavior	Updated list of supported products.	Major

9 Index

A

Abstract data model
server 16
Applicability 13
ArrayOfString complex type 15
Attribute groups 15
Attributes 15

C

Capability negotiation 13
Change tracking 57
Client
overview 16
Complex types 14
ArrayOfString 15
SOAPFaultDetails 15

D

Data model - abstract
server 16

E

Events
local - server 41
timer - server 41
Examples
migrating a site 42

F

Fields - vendor-extensible 13
Full WSDL 45

G

Glossary 8
Groups 15

I

Implementer - security considerations 44
Index of security parameters 44
Informative references 12
Initialization
server 16
Introduction 8

L

Local events
server 41

M

Message processing
server 16

Messages
ArrayOfString complex type 15
attribute groups 15
attributes 15
complex types 14
elements 14
enumerated 14
groups 15
namespaces 14
simple types 15
SOAPFaultDetails complex type 15
syntax 14
transport 14
Migrating a site example 42

N

Namespaces 14
Normative references 11

O

Operations
CreateWeb 34
DeleteWeb 36
ExportSolution 17
ExportWeb 18
ExportWorkflowTemplate 21
GetSite 23
GetSiteTemplates 24
GetUpdatedFormDigest 27
GetUpdatedFormDigestInformation 29
ImportWeb 31
IsScriptSafeUrl 37
IsScriptSafeUrlUsingCustomizedDomain 39
Overview (synopsis) 12

P

Parameters - security index 44
Preconditions 13
Prerequisites 13
Product behavior 55
Protocol Details
overview 16

R

References 11
informative 12
normative 11
Relationship to other protocols 13

S

Security
implementer considerations 44
parameter index 44
Sequencing rules
server 16
Server