[MS-SITESS]:
Sites Web Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

Revision Summary

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

Table of Contents

1 Introduction

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

1.1 Glossary

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

access control list (ACL)
anonymous authentication
Coordinated Universal Time (UTC)
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
language code identifier (LCID)
XML

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

available site template
collation order
content type
custom action
document library
farm
field
file
form
form digest validation
fully qualified URL
list
membership group
page
parent site
permission
presence
security group
server-relative URL
site
site collection
site collection identifier
site definition
site definition configuration
site solution
site template
site-collection relative URL
site-relative URL
SOAP (Simple Object Access Protocol)
SOAP action
SOAP body
SOAP fault
solution gallery
solution package
subsite
URL (Uniform Resource Locator)

user code
Web site
workflow
workflow template
WSDL message
WSDL operation
XML attribute
XML element
XML namespace
XML namespace prefix
XML Schema

The following terms are specific to this document:

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

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

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

1.2 References

1.2.1 Normative References

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

[MS-CSOM] Microsoft Corporation, "SharePoint Client Query Protocol Specification", July 2009.

[MS-PRIMEPF] Microsoft Corporation, "Deployment Package Format Specification", June 2008.

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

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

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

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

[SOAP1.2/2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part2-20030624

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

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

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

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

1.2.2 Informative References

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

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

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

1.3 Protocol Overview (Synopsis)

This protocol provides methods to create and delete **sites (2)** 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 **SOAP (Simple Object Access Protocol)** 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.

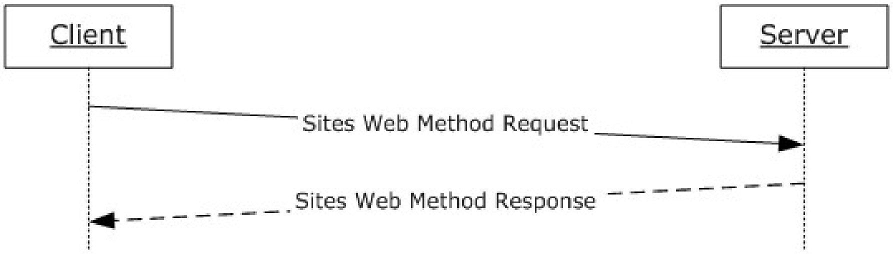

Figure 1: Sites Web method sequence diagram

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] and [SOAP1.2/2]. It transmits those messages by using **HTTP**, as described in [RFC2616], or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818].

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

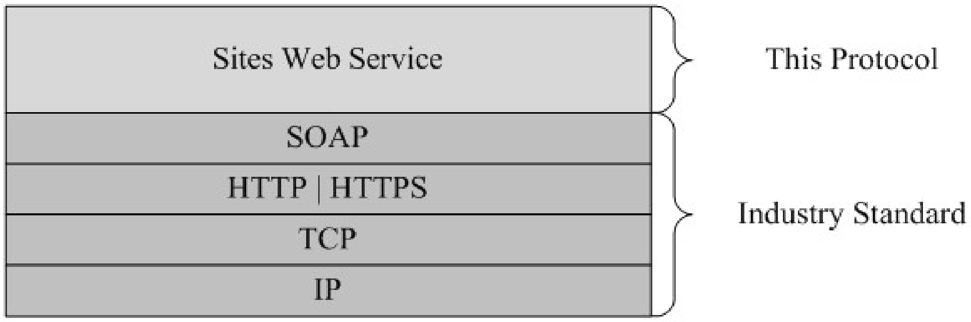

Figure 2: This protocol in relation to other protocols

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 (2).
- Creating and deleting a site (2).
- 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 specified in [XMLSCHEMA1] and [XMLSCHEMA2], and Web Services Description Language, as specified 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] [SOAP1.2/2]
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.

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.

2.2.5 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] 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.

Operation	Description
GetSiteTemplates	Getting available site template information.
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 action value of the message is defined as:

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

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 action value of the message is defined as:

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

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. This first file MUST include the list of the content migration package files created. 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 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 that invokes 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).

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 the ((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 and AddListItems permissions in the solution gallery.
7	ExportError: An error other than the errors listed in this table occurred during 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 can occur if the specified path was invalid, read-only, the caller is not authorized to access it, or the file is already open.

3.1.4.3 ExportWorkflowTemplate

This operation is used to export a declarative workflow template as a **site solution** to the specified document library.<4>

```
<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 action value of the message is defined as:

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

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 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 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 an existing workflow template with the specified name does not exist, the server MUST return a **SOAP fault** with no 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.<5>

```
<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 action value of the message is defined as:

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

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 URL (Uniform Resource Locator) 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="1" 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 MUST be of the following **XML** form:

```
  <Site Url="<Url>" Id="<Id>" UserCodeEnabled="<UserCodeEnabled>" />
```

Where <*Url*> is the absolute URL (Uniform Resource Locator) of the site collection, `<Id>` is the **site collection identifier** of the site collection and <*UserCodeEnabled*> specifies whether **user code** is enabled for the site collection. The value for <*UserCodeEnabled*> 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 action value of the message is defined as:

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

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="1" 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.<6>

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.<7> 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<8> 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.<9> 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.

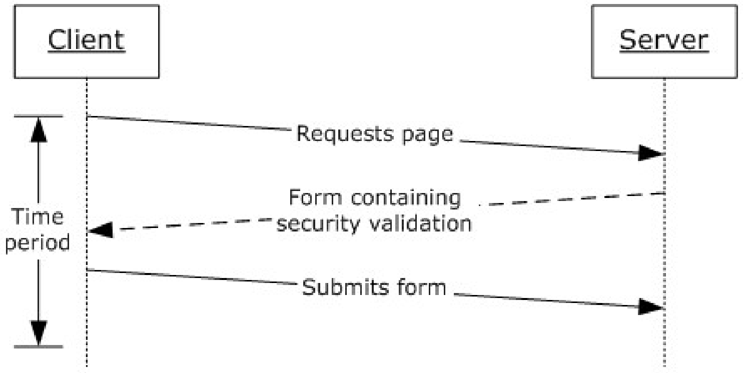

Figure 3: Form submission

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.

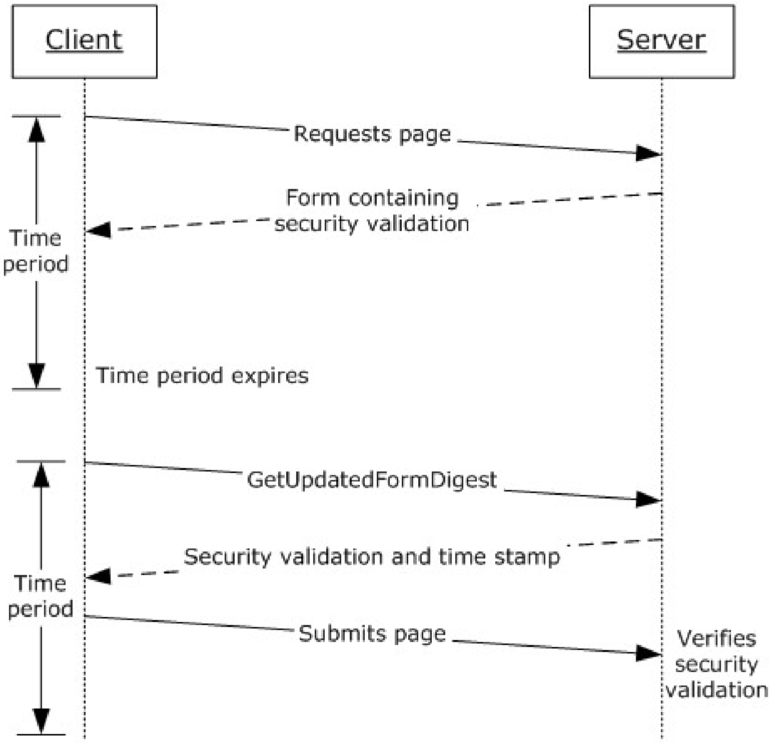

Figure 4: Renewal of an expired form digest validation

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 action value of the message is defined as:

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

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.<10>

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.<11>

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 action value of the message is defined as:

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

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 (2). Otherwise, the protocol server MUST return the **FormDigestInformation** of the site (2) 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="1" maxOccurs="1" name="DigestValue" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="TimeoutSeconds" type="s:int" />
      <s:element minOccurs="1" maxOccurs="1" name="WebFullUrl" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="LibraryVersion" type="s:string" />
      <s:element minOccurs="1" 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 (2).

LibraryVersion: The library version, as specified in [MS-CSOM] section 3.1.4.1.4.11, of the protocol server.

SupportedSchemaVersions: The comma-separated list of the SharePoint Client Query Protocol versions, as specified in [MS-CSOM] section 3.1.4.1.3.34.2, that are supported by the protocol server.

3.1.4.8 ImportWeb

This operation<12> 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.<13>
- 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 action value of the message is defined as:

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

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 3.1.4.8.3.1.

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:<14> 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.
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**.
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.

3.1.4.8.3 Complex Types

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

3.1.4.8.3.1 ArrayOfString

This complex type contains an array of string elements. It is used to submit the location of all necessary content migration package files for the **ImportWeb** operation.

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

string: The full path of a content migration package file.

3.1.4.9 CreateWeb

This operation is used to create a new subsite of the current site (2).<15>

```
<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 (2).

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 (2).

The SOAP action value of the message is defined as:

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

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.

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 (2).<16>

```
<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 (2).

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 (2).

The SOAP action value of the message is defined as:

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

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>
```

3.1.5 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 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="1" maxOccurs="1" name="WebFullUrl" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="LibraryVersion" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="SupportedSchemaVersions"
type="s:string" />
       </s:sequence>
     </s:complexType>
      <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: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: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="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="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="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="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 released service packs:

- Microsoft® Office SharePoint® Server 2007
- Microsoft® SharePoint® Foundation 2010
- Windows® SharePoint® Services 2.0
- Windows® SharePoint® Services 3.0

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

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

<1> Section 3.1.4.1: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and 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. The client retrieves the file when it is created and parses the file to discover the result of the operation.

<4> Section 3.1.4.3: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and Office SharePoint Server 2007 do not support this method.

<5> 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.

<6> Section 3.1.4.5.2.2: SharePoint Foundation 2010 does not return the SOAP fault. It returns a successful **GetSiteTemplatesResponse** with an empty **TemplateList** element.

<7> 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**.

<8> Section 3.1.4.6: Windows SharePoint Services 2.0 does not support this operation.

<9> Section 3.1.4.6: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 default to a 30-minute security validation time period.

<10> 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.

<11> 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.

<12> Section 3.1.4.8: Windows SharePoint Services 2.0 does not support this operation.

<13> 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.

<14> Section 3.1.4.8.2.1: This parameter SHOULD be optional. In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, ImportWeb returns an error if it is not specified.

<15> 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.

<16> 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.

8 Change Tracking

This section identifies changes that were made to the [MS-SITESS] protocol document between the November 2010 and December 2010 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- Changes made for template compliance.
- Removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type "Editorially updated."

Some important terms used in revision type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
1.1 Glossary	Added term "site template".	N	New content added.
3.1.4.2.2.1 ExportWeb	Appended description of the webUrl element specifying formatting of the host name portion of URL.	Y	New content added.
3.1.4.2.2.1 ExportWeb	Added product behavior note for the cabSize element.	Y	New product behavior note added.
3.1.4.2.2.1 ExportWeb	Removed product behavior note for cabSize.	Y	Product behavior note removed.
3.1.4.2.2.1 ExportWeb	Updated description of cabSize element.	Y	Content updated.
3.1.4.5.3.2 Template	Updated definition of attribute "FilterCategories" to refer to the term "site template".	N	Content updated.
3.1.4.5.3.2 Template	Added product behavior note for "FilterCategories" attribute.	Y	New product behavior note added.

9 Index